Lecture Notes in Networks and Systems 531

Pablo García Bringas · Hilde Pérez García ·
Francisco Javier Martinez-de-Pison ·
José Ramón Villar Flecha · Alicia Troncoso Lora ·
Enrique A. de la Cal · Álvaro Herrero ·
Francisco Martínez Álvarez · Giuseppe Psaila ·
Héctor Quintián · Emilio S. Corchado Rodriguez *Editors*

17th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2022)

Salamanca, Spain, September 5–7, 2022, Proceedings

Springer

Lecture Notes in Networks and Systems

Volume 531

The series "Lecture Notes in Networks and Systems" publishes the latest developments in Networks and Systems—quickly, informally and with high quality. Original research reported in proceedings and post-proceedings represents the core of LNNS.

Volumes published in LNNS embrace all aspects and subfields of, as well as new challenges in, Networks and Systems.

The series contains proceedings and edited volumes in systems and networks, spanning the areas of Cyber-Physical Systems, Autonomous Systems, Sensor Networks, Control Systems, Energy Systems, Automotive Systems, Biological Systems, Vehicular Networking and Connected Vehicles, Aerospace Systems, Automation, Manufacturing, Smart Grids, Nonlinear Systems, Power Systems, Robotics, Social Systems, Economic Systems and other. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution and exposure which enable both a wide and rapid dissemination of research output.

The series covers the theory, applications, and perspectives on the state of the art and future developments relevant to systems and networks, decision making, control, complex processes and related areas, as embedded in the fields of interdisciplinary and applied sciences, engineering, computer science, physics, economics, social, and life sciences, as well as the paradigms and methodologies behind them.

Indexed by SCOPUS, INSPEC, WTI Frankfurt eG, zbMATH, SCImago.

All books published in the series are submitted for consideration in Web of Science.

For proposals from Asia please contact Aninda Bose (aninda.bose@springer.com).

More information about this series at https://link.springer.com/bookseries/15179

Pablo García Bringas · Hilde Pérez García ·
Francisco Javier Martinez-de-Pison ·
José Ramón Villar Flecha ·
Alicia Troncoso Lora · Enrique A. de la Cal ·
Álvaro Herrero · Francisco Martínez Álvarez ·
Giuseppe Psaila · Héctor Quintián ·
Emilio S. Corchado Rodriguez
Editors

17th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2022)

Salamanca, Spain, September 5–7, 2022
Proceedings

Springer

International Conference on
SOCO
Soft Computing Models in Industrial
and Environmental Applications

Editors

Pablo García Bringas
Faculty of Engineering
University of Deusto
Bilbao, Spain

Francisco Javier Martinez-de-Pison
Mechanical Engineering Department
University of La Rioja
Logroño, La Rioja, Spain

Alicia Troncoso Lora
Data Science and Big Data Lab
Pablo de Olavide University
Sevilla, Spain

Álvaro Herrero
Department of Civil Engineering
University of Burgos
Burgos, Spain

Giuseppe Psaila
DIGIP
University of Bergamo
Dalmine, Bergamo, Italy

Emilio S. Corchado Rodriguez
Department of Computing Science
University of Salamanca
Salamanca, Spain

Hilde Pérez García
University of León
León, Spain

José Ramón Villar Flecha
Inteligencia Artificial
University of Oviedo
A Coruña, La Coruña, Spain

Enrique A. de la Cal
University of Oviedo
Oviedo, Spain

Francisco Martínez Álvarez
School of Engineering
Pablo Olavide University
Seville, Spain

Héctor Quintián
Department of Industrial Engineering
University of A Coruña
Ferrol, Spain

ISSN 2367-3370 ISSN 2367-3389 (electronic)
Lecture Notes in Networks and Systems
ISBN 978-3-031-18049-1 ISBN 978-3-031-18050-7 (eBook)
https://doi.org/10.1007/978-3-031-18050-7

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This volume of Lecture Notes in Networks and Systems contains accepted papers presented at the *17th International Conference on Soft Computing Models in Industrial and Environmental Applications* (SOCO 2022). This conference was held in the beautiful city of Salamanca, Spain, in September 2022.

Soft computing represents a collection or set of computational techniques in machine learning, computer science, and some engineering disciplines, which investigate, simulate, and analyze very complex issues and phenomena.

After a peer-review process, the SOCO 2022 International Program Committee selected 64 papers published in these conference proceedings, representing an acceptance rate of 60%. In this relevant edition, a particular emphasis was put on the organization of special sessions. Seven special sessions were organized related to relevant topics such as Machine Learning and Computer Vision in Industry 4.0; Time Series Forecasting in Industrial and Environmental Applications; Optimization, Modeling, and Control by Soft Computing Techniques; Soft Computing Applied to Renewable Energy Systems; Preprocessing Big Data in Machine Learning; and Tackling Real-World Problems with Artificial Intelligence.

The selection of papers was extremely rigorous to maintain the high quality of the conference. We want to thank the members of the Program Committees for their hard work during the reviewing process. This is a crucial process for creating a high-standard conference; the SOCO conference would not exist without their help.

SOCO 2022 enjoyed outstanding keynote speeches by distinguished guest speakers: Prof. Ajith Abraham, Director of Machine Intelligence Research Labs (MIR Labs), and Prof. Guy De Tré head of the research group on Database, Document, and Content Management (DDCM) at Ghent University (Belgium), and Felix Barrio General Director at INCIBE (Spain).

SOCO 2022 has teamed up with "Neurocomputing" (Elsevier), "Logic Journal of the IGPL" (Oxford University Press), and Cybernetics & Systems (Taylor & Francis) for a suite of special issues, including selected papers from SOCO 2022.

Particular thanks go as well to the conference's main sponsors, Startup Olé, the CYL-HUB project financed with next-generation funds from the European Union; the Ministry of Labor and Social Economy; the Recovery, Transformation, and

Resilience Plan; and the State Public Employment Service, channeled through the Junta de Castilla y León, BISITE research group at the University of Salamanca, CTC research group at the University of A Coruña, and the University of Salamanca. They jointly contributed in an active and constructive manner to the success of this initiative.

We would like to thank all the special session organizers, contributing authors, as well as the members of the Program Committees and the Local Organizing Committee for their hard and highly valuable work. Their work has helped to contribute to the success of the SOCO 2022 event.

September 2022

Pablo García Bringas
Hilde Pérez García
Francisco Javier Martinez-de-Pison
José Ramón Villar Flecha
Alicia Troncoso Lora
Enrique A. de la Cal
Álvaro Herrero
Francisco Martínez Álvarez
Giuseppe Psaila
Héctor Quintián
Emilio S. Corchado Rodriguez

Organization

General Chair

Emilio Corchado	University of Salamanca, Spain

International Advisory Committee

Ashraf Saad	Armstrong Atlantic State University, USA
Amy Neustein	Linguistic Technology Systems, USA
Ajith Abraham	Machine Intelligence Research Labs—MIR Labs, Europe
Jon G. Hall	The Open University, UK
Paulo Novais	Universidade do Minho, Portugal
Amparo Alonso Betanzos	President Spanish Association for Artificial Intelligence (AEPIA), Spain
Michael Gabbay	King's College London, UK
Aditya Ghose	University of Wollongong, Australia
Saeid Nahavandi	Deakin University, Australia
Henri Pierreval	LIMOS UMR CNRS 6158 IFMA, France

Program Committee Chairs

Pablo García Bringas	University of Deusto, Spain
Hilde Pérez García	University of León, Spain
Francisco Javier Martínez de Pisón	University of La Rioja, Spain
José Ramón Villar Flecha	University of Oviedo, Spain
Alicia Troncoso Lora	Pablo Olavide University, Spain
Enrique A. de la Cal	University of Oviedo, Spain
Álvaro Herrero	University of Burgos, Spain
Francisco Martínez Álvarez	Pablo Olavide University, Spain
Giuseppe Psaila	University of Bergamo, Italy

Héctor Quintián	University of A Coruña, Spain
Emilio Corchado	University of Salamanca, Spain

Program Committee

Agustina Bouchet	University of Oviedo, Spain
Akemi Galvez-Tomida	University of Cantabria, Spain
Alfredo Jimenez	KEDGE Business School, Spain
Álvaro Herrero Cosio	University of Burgos, Spain
Álvaro Michelena Grandío	University of A Coruña, Spain
Andreea Vescan	Babes-Bolyai University, Cluj-Napoca, Romania
Andres Fuster-Guillo	University of Alicante, Spain
Andres Iglesias Prieto	University of Cantabria, Spain
Angel Arroyo	University of Burgos, Spain
Anna Bartkowiak	University of Wroclaw, Poland
Anna Kamińska-Chuchmała	Wrocław University of Technology, Poland
Anton Koval	Luleå University of Technology, Sweden
Antonio Bahamonde	University of Oviedo at Gijón, Spain
Antonio Sala	Polytechnique University of Valencia, Spain
Bartosz Krawczyk	Virginia Commonwealth University, USA
Beatriz De La Iglesia	University of East Anglia, UK
Bogdan Okreša Đurić	University of Zagreb, Croatia
Borja Sanz	University of Deusto, Spain
Carlos Cambra	University of Burgos, Spain
Carlos Pereira	ISEC, Portugal
Carmen Benavides	University of León, Spain
Damian Krenczyk	Silesian University of Technology, Poland
Daniel Honc	University of Pardubice, Czechia
Daniel Urda	University of Burgos, Spain
Daniela Perdukova	Technical University of Kosice, Slovakia
David Griol	University of Granada, Spain
Dragan Simic	University of Novi Sad, Serbia
Eduardo Solteiro Pires	UTAD University, Portugal
Eleni Mangina	UCD, Ireland
Eloy Irigoyen	University of the Basque Country, Spain
Enrique De La Cal Marín	University of Oviedo, Spain
Enrique Onieva	University of Deusto, Spain
Esteban Jove	University of A Coruña, Spain
Fernando Ribeiro	EST, Portugal
Fernando Sanchez Lasheras	University of Oviedo, Spain
Florentino Fdez-Riverola	University of Vigo, Spain
Francisco Martínez-Álvarez	Pablo de Olavide University, Spain
Francisco Zayas-Gato	University of A Coruña, Spain
Franjo Jovic	University of Osijek, Croatia

Giuseppe Psaila	University of Bergamo, Italy
Grzegorz Ćwikła	Silesian University of Technology, Poland
Hector Cogollos Adrián	University of Burgos, Spain
Héctor Quintián	University of A Coruña, Spain
Henri Pierreval	LIMOS-IFMA, France
Humberto Bustince	UPNA, Spain
Ioana Zelina	Technical University of Cluj-Napoca, North Center in Baia Mare, Romania
Isabel Sofia Sousa Brito	Polytechnic Institute of Beja, Portugal
Isaias Garcia	University of León, Spain
Jaroslav Marek	University of Pardubice, Czechia
Jaume Jordán	Polytechnique University of Valencia, Spain
Javier Díez-González	University of León, Spain
Javier Sanchis Saez	Polytechnique University of Valencia, Spain
Jesús D. Santos	University of Oviedo, Spain
Jiri Pospichal	University of Ss. Cyril and Methodius, Slovakia
Jorge Barbosa	ISEC, Portugal
Jorge García-Gutiérrez	University of Seville, Spain
Jose Carlos Metrolho	IPCB, Portugal
Jose Dorronsoro	Autonomous University of Madrid, Spain
José Francisco Torres Maldonado	Pablo de Olavide University, Spain
Jose Luis Calvo-Rolle	University of A Coruña, Spain
José Luis Casteleiro-Roca	University of A Coruña, Spain
Jose M. Molina	University Carlos III of Madrid, Spain
Jose Manuel Lopez-Guede	Basque Country University, Spain
José Ramón Villar	University of Oviedo, Spain
José Valente de Oliveira	University of Algarve, Portugal
Juan Albino Mendez	University of La Laguna, Spain
Juan Gomez Romero	University of Granada, Spain
Juan M. Alberola	Polytechnique University of Valencia, Spain
Julio César Puche Regaliza	University of Burgos, Spain
Laura Melgar-García	Pablo de Olavide University, Spain
Lidia Sánchez-González	Universidad de León, Spain
Luis Alfonso Fernández Serantes	FH-Joanneum University of Applied Sciences, Spain
Luis Paulo Reis	University of Porto, Portugal
Manuel Castejón-Limas	University of León, Spain
Manuel Graña	University of the Basque Country, Spain
Marcin Paprzycki	IBS PAN and WSM, Poland
Maria Fuente	University of Valladolid, Spain
Maria Teresa Godinho	Polytechnic Institute of Beja, Portugal
Matilde Santos	Complutense University of Madrid, Spain
Mehmet Emin Aydin	University of the West of England, UK
Michal Wozniak	Wroclaw University of Technology, Poland

Míriam Timiraos Díaz	University of A Coruña, Spain
Nashwa El-Bendary	Arab Academy for Science, Technology, and Maritime Transport, Egypt
Noelia Rico	University of Oviedo, Spain
Nuño Basurto	University of Burgos, Spain
Oscar Castillo	Tijuana Institute of Technology, Mexico
Ovidiu Cosma	Technical University of Cluj-Napoca, Romania
Pablo García Bringas	University of Deusto, Spain
Panagiotis Kyratsis	University of Western Macedonia, Greece
Paulo Moura Oliveira	UTAD University, Portugal
Pavel Skrabanek	Brno University of Technology, Czechia
Petr Dolezel	University of Pardubice, Czechia
Petrica Pop	Technical University of Cluj-Napoca, North University Center at Baia Mare, Romania
Qing Tan	Athabasca University, Canada, Canada
Reggie Davidrajuh	University of Stavanger, Norway
Robert Burduk	University Wroclaw, Poland
Rogério Dionísio	Polytechnic Institute of Castelo Branco, Portugal
Santiago Porras Alfonso	University of Burgos, Spain
Sebastian Saniuk	University of Zielona Gora, Poland
Stefano Pizzuti	Energy New Technologies and Sustainable Economic Development Agency (ENEA), Italy
Valeriu Manuel Ionescu	University of Pitesti, Romania
Vladimir Ilin	Fakultet Tehničkih Nauka, Serbia
Wei-Chiang Hong	Asia Eastern University of Science and Technology, Taiwan
Wilfried Elmenreich	Alpen-Adria-Universität Klagenfurt, Austria
Wojciech Bozejko	Wroclaw University of Technology, Poland
Zita Vale	GECAD—ISEP/IPP, Portugal
Andre D. L. Batako	Liverpool John Moores University, UK
Anna Burduk	Wroclaw University of Science and Technology, Poland
Antoon Bronselaer	University of Ghent, Belgium
Bożena Skołud	Silesian University of Technology, Poland
Gloria Bordogna	CNR IREA, Italy
Guy De Tré	University of Ghent, Belgium
Iker Pastor-López	University of Deusto, Spain
Javier Del Ser	University of the Basque Country, Spain
Javiwer Diaz	Aingura IoT, Spain
Katarzyna Antosz	Rzeszow University of Technology, Poland
Krzysztof Kalinowski	Silesian University of Technology, Poland
Marek Placzek	Silesian University of Technology, Poland

Pablo Garcia Bringas	University of Deusto, Spain
Paolo Fosci	University of Bergamo, Italy
Wojciech Bożejko	Wroclaw University of Science and Technology, Poland

Special Sessions

Machine Learning and Computer Vision in Industry 4.0

Program Committee

Enrique Dominguez (Organizer)	University of Málaga, Spain
Jose Garcia Rodriguez (Organizer)	University of Alicante, Spain
Ramón Moreno Jiménez (Organizer)	Grupo Antolin, Spain
Andres Fuster-Guillo	University of Alicante, Spain
Esteban José Palomo	University of Malaga, Spain
Ezequiel López-Rubio	University of Málaga, Spain
Jorge Azorín-López	University of Alicante, Spain
Jorge García-González	University of Málaga, Spain
Jose Luis Calvo-Rolle	University of A Coruña, Spain
Karl Thurnhofer-Hemsi	University of Málaga, Spain
Marcelo Saval-Calvo	University of Alicante, Spain
Miguel A. Molina-Cabello	University of Málaga, Spain
Miguel Cazorla	University of Alicante, Spain
Rafael M. Luque-Baena	University of Extremadura, Spain

Time Series Forecasting in Industrial and Environmental Applications

Program Committee

Federico Divina (Organizer)	Pablo de Olavide University, Spain
José F. Torres (Organizer)	Pablo de Olavide University, Spain
José Luis Vázquez Noguera (Organizer)	Universidad Nacional de Asunción, Paraguay
Mario Giacobini (Organizer)	University of Torino, Italy
Miguel García Torres (Organizer)	Pablo de Olavide University, Spain
Antonio Morales-Esteban	University of Seville, Spain

David Gutiérrez-Avilés	University of Seville, Spain
Diego Pedro Pinto Roa	Universidad Nacional de Asunción, Paraguay
Elvira Di Nardo	University of Torino, Italy
Jorge Reyes	NT2 Labs, Chile
José-Lázaro Amaro-Mellado	University of Seville, Spain
Laura Melgar-García	Pablo de Olavide University, Spain
Laura Sacerdote	University of Torino, Italy
Luís Filipe Domingues	Polytechnic Institute of Beja, Portugal
Manuel Jesús Jiménez Navarro	Pablo de Olavide University, Spain
Zeyar Aung	Khalifa University of Science and Technology, United Arab Emirates

Optimization, Modeling, and Control by Soft Computing Techniques

Program Committee

Ahmed Al-Jumaily (Organizer)	Auckland University of Technology, New Zealand
Eloy Irigoyen (Organizer)	University of the Basque Country, Spain
Jose Luis Calvo-Rolle (Organizer)	University of A Coruña, Spain
Maria Tomas-Rodriguez (Organizer)	University of London, UK
Matilde Santos Peñas (Organizer)	Complutense University of Madrid, Spain
Mikel Larrea Sukia (Organizer)	University of the Basque Country, Spain
Anna Burduk	Wrocław University of Technology, Poland
Antonio Javier Barragán	University of Huelva, Spain
Antonio Robles Alvarez	University of Oviedo, Spain
Antonio Sala	Polytechnique University of Valencia, Spain
Camelia-M. Pintea	Technical University of Cluj-Napoca, North University Center at Baia Mare, Romania
Davide Carneiro	Polytechnic Institute of Porto, Portugal
Eukene Imatz-Ojanguren	Tecnalia Research and Innovation, Spain
Fábio Silva	University of Minho, Portugal
Fernando Matia	Polytechnic University of Madrid, Spain
Ignacio Trojaola Bolinaga	IKERLAN, Spain
Javier Sanchis Saez	Polytechnique University of Valencia, Spain
Jesus Lozano	University of Extremadura, Spain
Jose Manuel Lopez-Guede	University of the Basque Country, Spain
Jose Luis Diez	Polytechnic University of Valencia, Spain

Juan Albino Mendez	University of La Laguna, Spain
Luciano Alonso	University of Cantabria, Spain
Luis Magdalena	Polytechnic University of Madrid, Spain
María José Pérez-Ilzarbe	UPNA, Spain
Pavel Brandstetter	VSB-Technical University of Ostrava, Czechia
Petr Dolezel	Polytechnique University of Valencia, Czechia
Ramon Vilanova	Autonomous University of Barcelona, Spain
Vicente Gomez-Garay	University of the Basque Country, Spain
Xabier Basogain Olabe	University of the Basque Country, Spain

Soft Computing Applied to Renewable Energy Systems

Program Committee

Fares M'Zoughi (Organizer)	University of the Basque Country, Spain
Jesus Enrique Sierra Garcia (Organizer)	University of Burgos, Spain
Matilde Santos Peñas (Organizer)	Complutense University of Madrid, Spain
Paweł Martynowicz (Organizer)	AGH University of Science and Technology, Poland
Payam Aboutalebi (Organizer)	University of the Basque Country, Spain
Asier Ibeas	Autonomous University of Barcelona, Spain
Hugo Diaz-Martínez	University of Lisbon, Portugal
Izaskun Garrido	University of the Basque Country, Spain
Jesus Fernandez-Lozano	University of Málaga, Spain
Ravi Pandit	University of Exeter, UK

Preprocessing Big Data in Machine Learning

Program Committee

Antonio J. Tallón-Ballesteros (Organizer)	University of Huelva, Spain
Luís Cavique (Organizer)	University of Aberta, Portugal
Simon Fong (Organizer)	University of Macau, Macao
Akash Punhani	SRM Institute of Science and Technology, India
David Glass	University of Ulster, UK
Elsa Rodrigues	Polytechnique Institute of Beja, Portugal
Hamidah Jantan	Universiti Teknologi MARA, Malaysia
Marcin Szpyrka	AGH University of Science and Technology, Poland
María José Ginzo Villamayor	University of Santiago de Compostela, Spain
Mohamed Ali Hadj Taieb	University of Sfax, Tunisia

Tatsuo Nakajima	Waseda University, Japan
Antonio J. Tallón-Ballesteros	University of Huelva, Spain
Luís Cavique	University of Aberta, Portugal

Tackling Real-World Problems with Artificial Intelligence

Program Committee

Alberto Gallucci Suárez	University of Oviedo, Spain
Beatriz de la Iglesia	University of East Anglia, UK
Enol García González	University of Oviedo, Spain
Enrique De La Cal Marín	University of Oviedo, Spain
Fernando Moncada	University of Oviedo, Spain
José R. Villar	University of Oviedo, Spain
Mirko Fáñez	University of Oviedo, Spain
Noelia Rico	University of Oviedo, Spain
Paloma Valverde	Technological Institute of Castilla y León, Spain
Petrica Pop	Technical University of Cluj-Napoca, Romania
Samad Barri Khojasteh	University of Oviedo, Spain
Víctor Gonzalez	University of Oviedo, Spain
Víctor M. Álvarez	University of Oviedo, Spain

SOCO 2022 Organizing Committee Chairs

Emilio Corchado	University of Salamanca, Spain
Héctor Quintián	University of A Coruña, Spain

SOCO 2022 Organizing Committee

Álvaro Herrero Cosio	University of Burgos, Spain
Jose Luis Calvo-Rolle	University of A Coruña, Spain
Ángel Arroyo	University of Burgos, Spain
Daniel Urda	University of Burgos, Spain
Nuño Basurto	University of Burgos, Spain
Carlos Cambra	University of Burgos, Spain
Esteban Jove	University of A Coruña, Spain
José Luis Casteleiro-Roca	University of A Coruña, Spain
Francisco Zayas-Gato	University of A Coruña, Spain
Álvaro Michelena	University of A Coruña, Spain
Míriam Timiraos Díaz	University of A Coruña, Spain

Contents

Decision Support and Deep Learning

Evolutionary Computing

Neural Networks and Data Mining

Soft Computing Applications

Special Session on Machine Learning and Computer Vision in Industry 4.0

Special Session on Time Series Forecasting in Industrial and Environmental Applications

Special Session on Optimization, Modeling and Control by Soft Computing Techniques

Special Session on Soft Computing Applied to Renewable Energy Systems

Special Session on Pre-processing Big Data in Machine Learning

Special Session on Tackling Real World Problems with Artificial Intelligence

Decision Support and Deep Learning

Anomaly Detection of Security Threats to Cyber-Physical Systems: A Study

Nicholas Jeffrey[1] (✉), Qing Tan[2], and José R. Villar[1]

[1] University of Oviedo, Oviedo, Spain
{uo292630,villarjose}@uniovi.es
[2] Athabasca University, Athabasca, Canada
qingt@athabasca.edu

Abstract. As the presence of Cyber-Physical Systems (CPS) becomes ubiquitous throughout all facets of modern society, malicious attacks by hostile actors have increased exponentially in recent years. Attacks on critical national infrastructure (CNI) such as oil pipelines or electrical power grids have become commonplace, as increased connectivity to the public internet increases the attack surface of CPS. This paper presents a study of the current academic literature describing the state of the art for anomaly detection of security threats to Cyber-Physical Systems, with a focus on life safety issues for industrial control networks (ICS), with the goal of improving the accuracy of anomaly detection. As a new contribution, this paper also identifies outstanding challenges in the field, and maps selected challenges to potential solutions and/or opportunities for further research.

Keywords: Cyber-physical systems security · IoT security · SCADA security · AI/ML in CPS · Human-in-the-loop cyber-physical systems (HitL-CPS) · Anomaly detection in CPS

1 Introduction

Cyber-Physical Systems (CPS) are integrated systems that combine software and physical components [1]. CPS have experienced exponential growth over the past decade, from fields as disparate as telemedicine, smart manufacturing, autonomous vehicles, Internet of Things, industrial control systems, smart power grids, remote laboratory environments, and many more. Academia tends to use the term Cyber-Physical System, while industry tends to use IoT for consumer-grade devices, and IIoT (Industrial Internet of Things) [2] for industrial control systems (manufacturing, process control, etc.).

The rapid growth [3] of CPS has outpaced advancements in cybersecurity, with new threat models and security challenges that lack a unified framework for secure design, malware resistance, and risk mitigations. Much of the attention from academia and industry is focused on consumer-grade IoT devices (smart home automation, etc.). Industrial-grade IoT seems to have less attention from academia and industry, which is unfortunate, as the consequences of IIoT failure are much higher (power grid failure, oil pipeline shutdowns, train switching, etc.) [4].

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 3–12, 2023.
https://doi.org/10.1007/978-3-031-18050-7_1

Threat detection and prevention is a mature industry in enterprise networks, with large and entrenched vendors (Checkpoint, Cisco, F-Secure, Kapersky, Microsoft, Sophos, Trend Micro, etc.) providing host-based and network-based Intrusion Detection Systems/Intrusion Prevention Systems (IDS/IPS). Cyber-Physical Systems do not yet have similar IDS/IPS capabilities [5].

Traditional Industrial Control Systems (ICS), also known as Supervisory Control and Data Acquisition (SCADA) have not adjusted to the ubiquitous connectivity of Industry 4.0 [7], and still largely consider security to be an afterthought [7]. Much of this is due to the (no longer accurate) assumption that the ICS/SCADA environment is on an isolated, air-gapped, and trusted network [8, 9]. Historically, the primary design goal of SCADA/ICS systems was extreme reliability and predictability. Basic cybersecurity practices such as complex passwords or onerous authentication requirements were seen as barriers to system accessibility and were therefore avoided by the designers and operators of these systems [8]. Anti-malware programs such as signature-based antivirus tools were similarly avoided, to eliminate the possibility of a false positive inadvertently quarantining critical system files. These historical systems typically ran on fully isolated and trusted networks, without connectivity to corporate networks, and definitely without any connectivity to the public Internet.

Additionally, the lack of standardization [10, 11] of historical SCADA/ICS systems resulted in widespread usage of proprietary communication protocols, leading to "security by obscurity" [12], due to lack of a robust method of peer review. System vendors typically lacked any method of providing updates or bug fixes, so newly discovered vulnerable systems would typically remain in place for the entire lifespan of the system, relying on network isolation for protection from threats. As modern CPS grew out of legacy SCADA/ICS systems, those historical design considerations became untenable, as connectivity to wireless networks became ubiquitous, as well as a rapid abandonment of isolated air-gapped network environments.

Legacy protocols used in SCADA/ICS (Modbus, DNP, Fieldbus, HART, etc.) [13] are increasingly giving way to TCP/IP used in CPS, largely driven by commercial motivations for connectivity to corporate computer networks and the Internet. The modern reality of CPS is a hyper-connected world where threat actors are omnipresent, and a hostile network environment must be assumed. As modern CPS become increasingly interconnected with other networks, the attack surface has increased exponentially, leading to increasingly frequent breaches of critical national infrastructure (CNI) such as oil pipelines [14], power grids [4], etc.

Due to historical design goals of SCADA/ICS, observability of system state [4] has typically been limited to the current real-time status of a particular sensor or actuator, with relatively simple threshold-based alerts for the system operator. The historical assumption of a SCADA/ICS running on an isolated and fully trusted network meant that intrusion detection and intrusion prevention (IDS/IPS) were not design priorities, leading to a lack of observability in the increasingly hostile network layer of the CPS, making it difficult to detect threats and malicious activity in an increasingly connected world. Anomaly detection of security threats to CPS has become more urgent and critical to industry and life safety, as CNI becomes increasingly interconnected to public networks. Therefore, further study is needed to advance the state of academic research on the issue, and to develop and apply preventative solutions for industry to ensure safe and secure implementations of CPS.

This study aims to gather a full understanding of the research issue, and to identify existing gaps in the current state of the art that are opportunities for further research efforts. The remainder of this paper is organized as follows; Sect. 2 provides a statistical analysis of the areas of coverage in existing literature, which will allow identification of gaps in the current research. Section 3 provides a literature analysis for key identified topics. Section 4 illustrates the currently outstanding challenges in the field, with potential solutions for advancing the state of art. Finally, Sect. 5 discusses the conclusions reached in this paper, as well as identifies opportunities for future research.

2 Statistical Analysis

The keywords described previously were used to search literature from the various described sources. A total of 310 papers and online articles were selected and reviewed for this study. As a study done by literature review, this section will provide statistical analysis to describe the existing research presented in the reviewed literature by publisher, publication type, publication year, and country of origin.

The top 5 publishers (IEEE 47%, ScienceDirect 15%, Springer 12%, ACM 9%, MDPI 4%, all others 13%) comprise the bulk of available research in this field and are all well-established academic publishers with robust levels of peer review and quality assurance.

Most of the research in this area is published in academic journals (59%), with academic conferences a close second (39%). The field of CPS security is also heavily influenced by industry, but those efforts are typically for short-term tactical responses to current market threats and opportunities. For competitive advantage and trade secret reasons, industry efforts are rarely shared with the broader community, with “security by obscurity” still a common tactic in industry. There is a noticeable lack of industry and academic collaboration in this field, which is an opportunity for improvement.

To maintain relevance in a rapidly changing field, the reviewed literature in this paper is within the last decade, with most articles from the past 3 years. The term “Cyber-Physical Systems” was coined in 2006 by the US-based National Science Foundation (NSF) [4], so little research exists before that date. Earlier research related to CPS existed in fields of cybernetics, industrial process control, and control logic and engineering.

The USA is the largest single source of research in the area, with the top 5 countries generating more research than all other countries combined. Of the top 5 countries, there are 3 countries (USA, UK, India) with English as an official language, making the overwhelming majority of the published research available in English, often to the exclusion of other languages. The remaining 2 countries in the top 5 (China and Germany) typically publish research in English as well, due to greater availability of reference literature and collaboration opportunities. China and Russia appear to be the only two countries with significant publications in local languages, perhaps due to the large sizes of their domestic industry and academic communities (Fig. 1, Fig. 2 and Fig. 3).

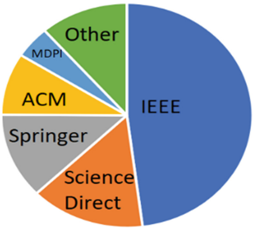


Fig. 1. Publishers

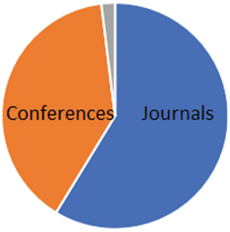


Fig. 2. Publication types

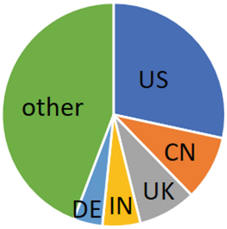


Fig. 3. Countries of Publication

3 Literature Analysis

Two of the commonly recurring themes in the available literature are CPS Security Design, and Anomaly Detection/Threat Detection in CPS, each of which will be discussed further below.

3.1 CPS Security Design

CPS is a broad field, and there is an interesting schism between the traditional SCADA systems used for industrial process control (now commonly referred to as IIoT), and the more consumer-focused IoT industry.

Due to product lifecycles measured in years or decades [15], and the historical design assumptions of operating in a fully trusted and air-gapped isolated environment, the traditional Industrial Control Systems (ICS) are much slower to adopt new technologies than their more agile counterparts in consumer-focused IoT devices that have product lifecycles measured in months to a few years.

Unlike their IIoT-based counterparts, the consumer-focused IoT industry was born in an age when ubiquitous connectivity to an increasingly hostile Internet was assumed, which helped drive adoption of standardized communication protocols around TCP/IP, with integrated authentication and encryption [16] functionality designed for the lightweight messaging protocols of devices assumed to have constrained processing power, battery life, and unreliable network connectivity.

Security design efforts for ICS/IIoT tend to focus on a hardened perimeter firewall separating the CPS from other networks, with little in the way of protection once inside the trusted network, reminiscent of the “hard shell, soft center” security posture of enterprise networks in decades past [17]. Due to historical design assumptions of a fully trusted network environment, there is still considerable resistance to actively blocking Intrusion Prevention Systems (IPS) being deployed with CPS, due to the high cost of false positives. Passive Intrusion Detection Systems (IDS) are seeing increasing acceptance in CPS, but due to the extreme heterogeneity, false positives are still a significant issue, making it difficult for the CPS operators to determine what is truly hostile network activity.

The more modern consumer-focused IoT industry has been quicker to adopt a zero-trust model of information security, accepting the reality that they operate in a potentially hostile network environment, and embedding strong authentication and encryption protocols by default [16]. Unfortunately, the rapid advancement of IoT means that product lifecycles are very short, making devices become obsolete quickly, leaving many "orphaned" devices without ongoing vendor support or upgrades to counter new security threats. While some vendors have included functionality for receiving trusted over-the-air updates to counter newly discovered threats, there are many IoT devices that entirely lack any sort of update functionality, leaving them permanently vulnerable to emerging threats.

Human-in-the-Loop Cyber-Physical Systems (HitL-CPS) are a unique subset of CPS that partially or completely rely on human operator input to control the CPS. This introduces unique security challenges, due to unpredictability from human error, inattentiveness, slower reaction time of humans, susceptibility to social engineering, inconsistent decision-making, etc. The key research in this area is from Nunes [18], who describes the most significant outstanding challenges in this area as gathering a full understanding of the problem domain, improvements in modeling unpredictable human behaviour, autonomic mitigations against intentional and unintentional human-introduced risks, and development of a formal methodology of integrating human feedback in the control loop. Each of these challenges are still in rapid states of development, so the maturity of this area of research is still in its early stages.

3.2 Anomaly Detection/Threat Detection in CPS

Threat detection methodologies can be broadly categorized [19] as signature-based, threshold-based, or behaviour-based. Traditional antivirus programs are an example of a signature-based threat detection methodology, using a centralized and regularly updated database of signatures of malicious files or traffic to trip an alarm on an IDS and/or IPS. Signature-based detection works well on IT networks thanks to standardized communication protocols and low levels of heterogeneity but suffers from high levels of false negatives on OT networks due to their proprietary communication protocols and heterogeneous physical components.

Threshold-based methodologies rely on known ranges of acceptable operation, which are relatively easy to define on IT networks. Examples of threshold-based threat detections for IT networks include network link utilization, communication latency, processor utilization levels, etc. However, OT networks have proven more difficult to accurately define known ranges of acceptable operation, due to real-world environmental fluctuations [20]. For example, a wireless mesh network of air quality sensors in a smart city environment may have communication latency impacted by fog or rain, making the thresholds of acceptable operation differ based on unpredictable weather conditions.

Kabiri and Chavoshi [20] propose a CPS design model that includes a inline hardware device that interrogates all commands sent to actuators, passively relaying all commands until a malicious or anomalous pattern is detected that would run the actuator outside of specified operational tolerances. This provides active protection of physical devices in OT networks, similar to inline IPS that are commonly used for protection of cyber assets in IT networks. IPS products suffer from false positives in IT networks due to complexity and unpredictability of traffic patterns. False positives have a higher impact in OT networks, due to financial implications of shutting down a CPS like a factory line or water treatment plant, so the tolerance for false positives is much lower in OT networks, which is still an outstanding research problem.

Behaviour-based methodologies are the most difficult to accurately define on IT networks and are even more challenging for OT networks [21]. Defining an accurate baseline of normal behaviour on an IT network requires a deep understanding of what normal system activity looks like, and it is rare that IT networks are completely unchanged over their entire lifecycle, making any definition of normal behaviour a moving target at best. These challenges are exacerbated on OT networks, which tend to be even more dynamic due to environmental factors such as weather-related variations in temperature, humidity, ambient light, etc. Additionally, the negative impact of a false positive or false negative detection on an OT network has more significant consequences, including physical equipment damage and life safety concerns.

There is considerable interest in the use of machine learning (ML) algorithms for automated threat detection in CPS, but few of the proposed frameworks from academia have seen significant adoption in industry. Due to the extreme diversity in CPS, it has proven difficult to generate a useful training model for AI/ML algorithms, which has resulted in unacceptably high levels of false positives and false negatives for automated anomaly detection. This appears to be a significant discontinuity between the efforts of academia and industry, and is an opportunity to improve collaboration.

Manufacturing processes in the so-called "Industry 4.0" have been particularly quick to adopt AI in CPS, both for protection from cybersecurity attacks, and operational efficiency. Alhaidari and AL-Dahasi [22] propose an improved framework for using AI to detect DDoS attacks using multiple machine learning algorithms. Different datasets were analyzed with ML algorithms for rapid detection of DDoS attacks, with preprocessing of raw data to minimize the size of the training model particularly helpful in increasing the mean time to detection of an attack. A significant conclusion drawn was the need for greater collaboration between the operators of individual CPS and broader industry participants for sharing of vulnerability information.

The major strategies, as well as their relative advantages and disadvantages are shown in the table below (Table 1).

Table 1. Comparison of anomaly detection strategies

Detection strategy	Advantages	Disadvantages
Signature	High accuracy for known threats	Wide variation in CPS makes it difficult to develop and maintain signature databases
Threshold	Simple design	External factors such as weather or operational changes can cause thresholds to vary over time, which can cause false positives
Behaviour	Best accuracy for unknown threats	Most difficult to accurately define

4 Outstanding Challenges

A modern CPS can be considered as a combination of corporate computer networks and industrial control networks, sometimes referred to as Information Technology (IT) and Operational Technology (OT), each of which have differing priorities.

Traditional IT networks have used the so-called Confidentiality, Integrity, Availability (CIA) triad to define the organizational security posture, with each facet listed in order of importance. OT networks reverse that order [23], with availability being the most important factor, followed by integrity, with confidentiality the least important facet of overall system security. This difference is largely due to CPS growing out of earlier SCADA/ICS networks used for industrial control processes, where availability was of the utmost importance, with integrity and confidentiality rarely considered due to usage of trusted and air-gapped isolated network environments.

As OT networks merged with IT networks to form modern CPS, those differing priorities have resulted in ongoing challenges that have yet to be fully resolved. IT networks heavily prioritize authentication (who you are) and authorization (what you are allowed to do), which roughly map to the confidentiality and integrity facets of the CIA triad of information security. However, OT networks have traditionally focused so heavily on the availability facet of the CIA triad, that authentication and authorization were assumed to be true [8] by virtue of physical access to the trusted and isolated OT network.

This historical assumption of a fully trusted and isolated environment is no longer true after the interconnection of IT and OT networks, resulting in vulnerability to common network-based attacks such as DDoS, MitM, replay attacks, impersonation, spoofing, false data injection, etc. Compounding the problem, OT networks typically lack integration with antimalware programs, as well as detailed logging capabilities, making it difficult to observe potentially hostile activity on OT networks [24].

There are ongoing efforts [12] to extend the IDS/IPS capabilities of IT networks into OT networks, but the lack of standardized protocols and interfaces to the physical components of CPS makes threat detection very challenging. Those IDS/IPS systems that have been extended into CPS environments struggle with high levels of false positives and false negatives, due to the complexity of CPS.

The single largest challenge facing the secure design and operation of CPS is their lack of standardized communication protocols and proprietary nature [25]. Due to the lack of even rough industry consensus for the system development life cycle of CPS, each system designer essentially builds each new CPS from scratch, without much consideration for multivendor interoperability, secure and robust patching mechanisms, or exposing system telemetry details in a consistent manner for health and security monitoring. This is slowly changing with industry consortiums forming standards bodies such as O-PAS (Open Process Automation Standard) [26], but broad industry consensus has proved elusive.

The highly proprietary nature of CPS products is due to their historical evolution from ICS, which were designed to operate on closed networks without interoperability or communication requirements with external networks. As OT and IT networks merged to become CPS, the open standards and communication protocols used by IT networks have been rapidly adopted by OT networks [27], but there is still significant opportunity for improvement, particularly for the OT networks that have unexpectedly found themselves connected to public and untrusted networks, including the Internet.

5 Conclusions

As a relatively young (since 2006) field of study, the state of the art for CPS is still rapidly evolving. For historical reasons, CPS lacked a coherent or standardized architecture, so are notable by their extreme diversity, which has hampered the development of threat mitigations in increasingly hostile networked environments. As malicious attacks on critical infrastructure continue to increase, the need for secure and resilient CPS becomes more urgent every day.

Opportunities for further development include increased collaboration between academia and industry, towards the development of best practices for secure design and operation of CPS, with security and observability included much earlier in the system development life cycle.

As the proprietary communication protocols of legacy CPS environments give way to modern standards-based TCP/IP protocols, there are opportunities for cross-pollination between the OT networks of yesterday and the IT networks of today. The use of AI/ML for threat detection is already commonplace in IT networks, but OT networks have seen very limited adoption of this technology, due to the higher cost of false positives. Further research work in this area is recommended.

It is particularly notable that the goals of academia and industry do not appear to be entirely aligned, with industry extremely hesitant to adopt academic proposals for changes to safety-critical systems. This is a potential opportunity for improved collaboration, as well as research into the development of more realistic simulated CPS environments for low-impact testing. Collaboration efforts are further hampered by the extreme diversity of CPS, making consensus-building around standardized best-practice design and architectural strategies for CPS a significant opportunity for improvement.

Acknowledgement. This research has been funded by the SUDOE Interreg Program -grant INUNDATIO-, by the Spanish Ministry of Economics and Industry, grant PID2020-112726RB-I00, by the Spanish Research Agency (AEI, Spain) under grant agreement RED2018-102312-T (IA-Biomed), and by the Ministry of Science and Innovation under CERVERA Excellence Network project CER-20211003 (IBERUS) and Missions Science and Innovation project MIG-20211008 (INMERBOT). Also, by Principado de Asturias, grant SV-PA-21-AYUD/2021/50994.

References

1. Zanero, S.: Cyber-physical systems. Computer **50**(4), 14–16 (2017). https://doi.org/10.1109/MC.2017.105
2. Radanliev, P., De Roure, D., Van Kleek, M., Santos, O., Ani, U.: Artificial intelligence in cyber physical systems. AI & Soc. **36**(3), 783–796 (2020). https://doi.org/10.1007/s00146-020-01049-0

3. Rouzbahani, H.M., Karimipour, H., Rahimnejad, A., Dehghantanha, A., Srivastava, G.: Anomaly detection in cyber-physical systems using machine learning. In: Choo, K.-K.R., Dehghantanha, A. (eds.) Handbook of Big Data Privacy, pp. 219–235. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-38557-6_10
4. Wolf, M., Serpanos, D.: Safety and security in cyber-physical systems and internet-of-things systems. Proc. IEEE **106**(1), 9–20 (2018). https://doi.org/10.1109/JPROC.2017.2781198
5. Langner, R.: To kill a centrifuge: a technical analysis of what stuxnet's creators tried to achieve. The Langner Group https://www.langner.com/to-kill-a-centrifuge/ (2011)
6. Tsochev, G., Sharabov, M.: Artificial intelligence methods used in industry 4.0 in particular industrial control systems. In: AIP Conference Proceedings, vol. 2333, p. 070017 (2021). https://doi.org/10.1063/5.0041610
7. Craggs, B., Rashid, A.: Smart cyber-physical systems: beyond usable security to security ergonomics by design. In: 2017 IEEE/ACM 3rd International Workshop on Software Engineering for Smart Cyber-Physical Systems (SEsCPS), pp. 22–25 (2017). https://doi.org/10.1109/SEsCPS.2017.5
8. Stout, W.M.S.: Toward a multi-agent system architecture for insight & cybersecurity in cyber-physical networks. In: 2018 International Carnahan Conference on Security Technology (ICCST), pp. 1–5 (2018). https://doi.org/10.1109/CCST.2018.8585632
9. Das, R., Menon, V., Morris, T.H.: On the edge realtime intrusion prevention system for DoS attack. In: Proceedings of 5th International Symposium for ICS & SCADA Cyber Security Research 2018 (ICS-CSR 2018) (2018). https://doi.org/10.14236/ewic/ICS2018.10
10. Maloney, M., Reilly, E., Siegel, M., Falco, G.: Cyber physical iot device manage-ment using a lightweight agent. In: 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp. 1009–1014 (2019). https://doi.org/10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00176
11. Rehman, S., Gruhn, V.: An effective security requirements engineering framework for cyber-physical systems. Technologies **2018**(6), 65 (2018). https://doi.org/10.3390/technologies6030065
12. Qassim, Q.S., Jamil, N., Mahdi, M.N., Abdul Rahim, A.A.: Towards SCADA threat intelligence based on intrusion detection systems - a short review. In: 2020 8th International Conference on Information Technology and Multimedia (ICIMU), pp. 144–149 (2020). https://doi.org/10.1109/ICIMU49871.2020.9243337
13. Benbenishti, L.: SCADA MODBUS Protocol Vulnerabilities. Cyberbit (2017). https://www.cyberbit.com/blog/ot-security/scada-modbus-protocol-vulnerabilities/
14. Osborne, C.: Colonial pipeline attack: everything you need to know. Zdnet (2021). https://www.zdnet.com/article/colonial-pipeline-ransomware-attack-everything-you-need-to-know/
15. Rubio, J.E., Alcaraz, C., Roman, R., Lopez, J.: Current cyber-defense trends in industrial control systems. Comput. Secur. **87**, 101561 (2019). https://doi.org/10.1016/j.cose.2019.06.015
16. Toshihiko, O.: Lightweight cryptography applicable to various IoT devices. NEC Technical J. **12**(1), 67–71 (2017). https://www.nec.com/en/global/techrep/journal/g17/n01/170114.html
17. Adamsky, F., et al.: Integrated protection of industrial control systems from cyber-attacks: the ATENA approach. Int. J. Crit. Infrastruct. Prot. **21**, 72–82 (2018). https://doi.org/10.1016/j.ijcip.2018.04.004
18. Nunes, D., Sá Silva, J., Boavida, F.: A Practical Introduction to Human-in-the-Loop Cyber-Physical Systems. Wiley Publishing (2018). https://doi.org/10.1002/9781119377795
19. Wu, M., Song, Z., Moon, Y.B.: Detecting cyber-physical attacks in CyberManufacturing systems with machine learning methods. J. Intell. Manuf. **30**(3), 1111–1123 (2017). https://doi.org/10.1007/s10845-017-1315-5

20. Kabiri, P., Chavoshi, M.: Destructive attacks detection and response system for physical devices in cyber-physical systems. In: 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), pp. 1–6 (2019). https://doi.org/10.1109/CyberSecPODS.2019.8884999
21. Etalle, S.: Network monitoring of industrial control systems: the lessons of security matters. In: CPS-SPC'19: Proceedings of the ACM Workshop on Cyber-Physical Systems Security & Privacy (2019). https://doi.org/10.1145/3338499.3357354
22. Alhaidari, F.A., AL-Dahasi, E.M.: New approach to determine DDoS attack patterns on SCADA system using machine learning. In: 2019 International Conference on Computer and Information Sciences (ICCIS), pp. 1–6 (2019). https://doi.org/10.1109/ICCISci.2019.8716432
23. Ashibani, Y., Mahmoud, Q.H.: Cyber physical systems security: analysis, challenges and solutions. Comput. Secur. **68**(2017), 81–97 (2017). https://doi.org/10.1016/j.cose.2017.04.005
24. Bostjancic Rakas, S.V., Stojanovic, M.D., Markovic-Petrovic, J.D.: A review of research work on network-based SCADA intrusion detection systems. IEEE Access **8**, 93083–93108 (2020). https://doi.org/10.1109/ACCESS.2020.2994961
25. Sundararajan, A., Chavan, D., Saleem, A.I.S.: A survey of protocol-level challenges and solutions for distributed energy resource cyber-physical security. Energies **2018**(11), 2360 (2018). https://doi.org/10.3390/en11092360
26. Bartusiak, R.D., et al.: Open Process Automation: A standards-based, open, secure, interoperable process control archi-tecture. Control Eng. Pract. **121**, 105034 (2022). https://doi.org/10.1016/j.conengprac.2021.105034
27. Kabore, R., Kouassi, A., N'goran, R., Asseu, O., Kermarrec, Y., Lenca, P.: Review of anomaly detection systems in industrial control systems using deep feature learning approach. Engineering **13**(01), 30–44 (2021). https://doi.org/10.4236/eng.2021.131003

Predictive Maintenance for Maintenance-Effective Manufacturing Using Machine Learning Approaches

Bruno Mota[1,2], Pedro Faria[1,2](✉), and Carlos Ramos[1,2]

[1] GECAD - Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development, Porto, Portugal
{bamoa,pnf,csr}@isep.ipp.pt

[2] LASI - Intelligent Systems Associate Laboratory, Polytechnic of Porto, Porto, Portugal

Abstract. Recent advancements in technologies such as in Big data and Internet of Things have made Predictive Maintenance (PdM) a key strategy to reduce unnecessary maintenance costs and improve product quality in the manufacturing sector. The premise of this paper is to implement and explore some of the most promising machine learning models for PdM, a Gradient Boosting (GB) model, and a Support Vector Machine (SVM) model. An innovative methodology for model training is proposed that aims to improve model performance while also allowing for continuous training. Furthermore, it is proposed an automatic hyperparameter tunning approach for the GB and SVM models. A synthetic dataset that reflects industrial machine data was used to validate the proposed methodology. The implemented models can achieve up to 0.92 recall and 94.55% accuracy, highlighting the effectiveness of the proposed methodology.

Keywords: Data preprocessing · Gradient boosting · Hyperparameter optimization · Predictive maintenance · Support vector machine

1 Introduction

Nowadays, due to the emergence of new and improved technologies such as in Big data, Internet of Things, data analytics, augmented reality, and cloud computing, there has been a shift in the industrial maintenance strategy to systems capable of predicting machine longevity [1, 2]. Moreover, energy aspects are also very important to consider when optimizing production lines in manufacturing environments, since machines' health can play a big role in affecting a machine's energy efficiency capabilities [3, 4]. This has led mainly to the research of two new maintenance concepts to identify irregularities in the manufacturing environment, condition-based maintenance, and prognostic and health management [5]. Predictive Maintenance (PdM), which uses prior data to predict behavior patterns, is often employed with these two concepts in mind, either with condition-based maintenance or prognostic and health management, and in some cases, with both of them [6].

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 13–22, 2023.
https://doi.org/10.1007/978-3-031-18050-7_2

This paper aims to implement and explore the benefits and drawbacks of two machine learning techniques for PdM: GBs and SVMs, due to their robustness, performance, and the author's extensive experience in using these models. It focuses on proposing a novel training machine learning methodology for PdM that increases the models' performance when dealing with unbalanced and irrelevant/erroneous data, as well as explore how these models perform in this type of problem. In addition, an automatic hyperparameter optimization approach is taken into account in order to find the best hyperparameters for the GB and SVM, thus further improving the models' performance. Moreover, it proposes how an application of the proposed models could be implemented in real-time, both for user application and retraining of the models. Also, to validate the implemented models, these are compared to a bagged trees ensemble classifier proposed in [7], that uses the same dataset.

The main advantages of the proposed methodology and models are:

- Good to high performance when dealing with unbalanced and irrelevant/erroneous data;
- Continuous improvement of the models, due to their ability of retraining;
- Easy implementation of the models, as a result of the automatic hyperparameter tunning approach;
- High flexibility to apply to other contexts, since it requires a low number of common features from machine data.

On the other hand, its biggest disadvantages/limitations are:

- Long execution times for the automatic hyperparameter tunning, depending on the possible range of values for each hyperparameter;
- Inability to detect types of machine failures.

This paper structure is divided into six main sections. This first introductory section presents a contextualization to the objective of the paper, while Sect. 2 shows the state-of-the-art regarding PdM techniques. Section 3 describes in detail the dataset used for training and testing of the machine learning models. Section 4 presents the proposed methodology for PdM on GB and SVM models. Section 5 provides the results to validate the proposed solution while also discussing the obtained results. Finally, in Sect. 6, the conclusions are presented.

2 State-of-the-Art

In a manufacturing environment, the use of predictive systems to detect when maintenance activities are required is crucial not only to reduce unnecessary costs but also to improve product quality. Predictive maintenance allows for continuous monitoring of the machine's integrity, allowing maintenance to be conducted only when it's really necessary. Furthermore, prediction systems based on statistical inference methods, historical data, integrity variables, and engineering approaches enable the early detection of problems [8]. There are a variety of techniques for predicting systems' health and/or condition, for instance, the usage of Gradient Boosting (GB) [9, 10], Support Vector Machines (SVMs) [11, 12], artificial neural networks, random forests, deep learning, k-means, naive bayes, and decision trees [8, 13].

While there are many techniques for PdM, GB models are highly used due to their robustness, for example, in [9] it is proposed a monitoring and fault detection system for wind turbines' critical components. It uses historical data from components such as gearboxes and generators to predict when there is anomalous behavior, and arrange maintenance according to it. In the proposed methodology, an extreme gradient boosting and long short-term memory models are explored, with the extreme gradient boosting being better at modeling the gearbox. Also, in [10] it is proposed a novel data-driven approach for PdM applied to woodworking industrial machines. It implements and explores the performance of three different machine learning models, GB, random forest, and extreme gradient boosting, to predict the remaining useful lifetime of a machine. Of the three models, the GB had the best performance when taking into consideration the metrics recall, precision, and accuracy.

Nevertheless, GB is not the only robust model for PdM, SVM models can also provide great performance. A PdM system for industrial IoT of vehicle fleets is proposed in [11], which focuses on detecting vehicle faults. It achieves these results by employing a hierarchical modified fuzzy support vector machine model. The effectiveness of the proposed model is demonstrated when compared to other popular PdM approaches such as logistic regression and random forests, with the two latter ones having worse performance that the proposed model. Moreover, in [12] it is proposed a SVM for machine prognostic in industrial environments. In addition, it also proposes an innovative modified regression kernel to be applied in the SVM that aims to improve the system's performance.

3 Training/Testing Dataset

The PdM dataset used for training and testing of the proposed solution was obtained from the Machine Learning Repository from the University of California, Irvine [14]. The PdM dataset, titled "AI4I 2020 Predictive Maintenance Dataset Data Set" from 2020, is publicly available in [15]. It represents a synthetic dataset that aims to reflect industrial machine data for PdM. It is worth noting that the dataset consists of 10000 data points, 14 features, and is characterized as being multivariate and time-series based.

The most relevant features from the PdM dataset used in the training of the proposed models are:

- Air temperature – Represents the air temperature outside the machine, in Kelvin (K);
- Process temperature – Describes the temperature generated inside the machine, in Kelvin (K);
- Rotational speed – Depicts the rotational speed of the tools inside the machine, in Revolutions per minute (rpm);
- Torque – Describes the force produced to rotate the tools inside the machine, in Newton-meters (Nm);
- Tool wear – Represents the level of degradation of the tools inside the machine, in minutes (min);

- Machine failure – Defines whether a machine has failed or not. It assumes the value of 0 or 1 for non-failure and failure, respectively.

4 Proposed Methodology

For PdM, two machine learning models are implemented and explored, a GB, and a SVM model.

The training process can be done in batches or mini-batches. In an initial phase, the batch approach was taken since it allows to have a model ready to be applied in the real world. Nevertheless, after the initial model is constructed the training process is made through mini-batches, in real-time, as represented in Fig. 1. It is worth noting that the GB and SVM models do not possess the ability to continue the training of an already existing model, and thus, need to be trained from the beginning.

For real-time training, the proposed solution starts by obtaining the newest machine data (i.e., air temperature, machines' process temperature, rotational speed, torque, tool wear, and machine failure information) from the machine data database employed in the facility. The database can describe either the culmination of all machine data from all the machines in the facility, or the data of a single machine. Then, before training starts, a data preprocessing phase begins in which: (1) aggregate all the data collected into a single data file, if the data is separated into different files (i.e., data aggregator); (2) normalize data scales and types, primarily between data from two different machines, using a Min-Max strategy (i.e., data normalization); (3) fill missing values on the gathered data, by using a k-Nearest Neighbors imputation technique (i.e., data imputation); (4) remove and correct possible irrelevant and erroneous data, by detecting outliers using the Z-score technique (i.e., data filtering); (5) transform raw data into features that better represent the underlying problem (i.e., data engineering); and finally, (6) balance machine data failure and non-failure points, by using the imbalanced-learn [16] library (i.e., data balancing).

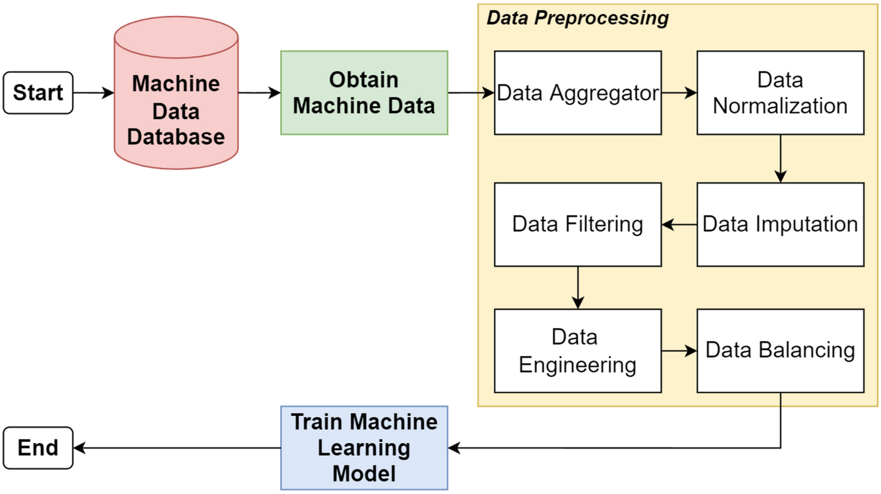


Fig. 1. Flowchart of the proposed machine learning model training process

Afterward, the preprocessed data is fed to the machine learning model (i.e., GB or SVM) for training, where the model has to be reconstructed from the beginning. This training process can start every time there is new data in the machine data database.

The initial training process, done on both models (GB, SVM) is characterized by:

- The Holdout method, with 80% of the data set apart for training, and the rest for testing (8000 data points for training and 2000 for testing);
- Data normalization, done using a Min-Max strategy;
- Feature engineering, by creating a new feature, temperature difference, which represents the machine process temperature minus the air temperature;
- Data balancing, done with a 5% oversampling strategy on the failure data points and a majority undersampling strategy on the non-failure data points, by using the imbalanced-learn [16] library. It is noteworthy that there is a big gap of around 1:28 ratio (339 failures and 9661 non-failures) of instances where a machine failed to when it did not fail, respectively;
- Cross-validation splitting strategy when searching for the best hyperparameters, done with a 5-fold cross validation.

Regarding the application in real-time of the proposed machine learning models, which is represented in Fig. 2, it can be divided into four crucial phases: data acquisition, data preprocessing, and the usage of the respective machine learning model.

The application in real-time of the proposed models for PdM starts with the acquisition of machine data from the machine intended to be evaluated. Next, a data preprocessing phase begins which focuses on data normalization, imputation, filtering, and engineering, similar to the training process. Finally, the preprocessed data is fed to the machine learning model, which predicts the machine failure status, representing 1 for failure and 0 for non-failure.

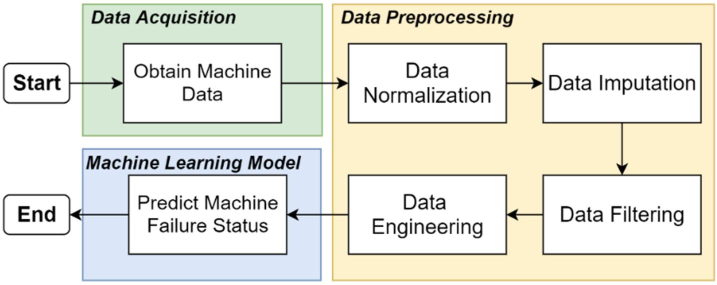


Fig. 2. Flowchart of the proposed machine learning model application in real-time

4.1 Gradient Boosting Training

The training of the GB was done using an automatic hyperparameter tuner, which aims to find the optimal hyperparameter values. This is accomplished through the RandomizedSearchCV [17] technique available in the Scikit-Learn library [18]. The proposed technique explores randomly the possible values for each hyperparameter in order to find the best GB model which contains the optimal values for each hyperparameter. Table 1 presents the possible and optimal hyperparameter values for the GB model using this method.

Table 1. Gradient Boosting hyperparameters possible and optimal values using the automatic hyperparameter tuner method

Hype-parameter	Possible values	Optimal value
Criterion	Friedman MSE, Squared error, MSE, or MAE	Friedman MSE
Learning rate	5%, 7.5%, 1%, 25%, 50%, 75%, or 100%	100%
Maximum depth	10 to 32	12
Maximum features	Auto, Sqrt, or Log2	Log2
Minimum samples in a leaf	1, 2, 4, 6, 8, or 10	4
Minimum samples to split	2, 5, 10, 20, or 30	2
Number of estimators	200 to 3000	2000

The used GB classifier is the GradientBoostingClassifier [19] from the Scikit-Learn library. During the training phase, the classifier devises rules to obtain the lowest possible accuracy error in comparison to the training classes. When the model has been appropriately fitted with training data, it is ready to make predictions.

4.2 Support Vector Machine Training

For the tuning of the hyperparameters of the SVM model, an automatic hyperparameter optimization approach was taken into account. To achieve this, it is used the RandomizedSearchCV [17] technique. This technique focuses on finding the best estimator to be used in the model, by randomly choosing one of the possible values for each hyperparameter and then scoring each estimator according to their accuracy scores. The possible values and corresponding optimal value for each hyperparameter, using the proposed method, are presented in Table 2.

Table 2. Support vector machine hyperparameters possible and optimal values using the automatic hyperparameter tuner method

Hype-parameter	Possible values	Optimal value
C (Regularization parameter)	0.1, 1, 10, 100, or 1000	100
Gamma	Scale, Auto, 1, 0.1, 0.01, 0.001, or 0.0001	Scale
Kernel	Linear, Poly, RBF, or Sigmoid	RBF
Probability estimates	True or False	False
Shrinking heuristic	True or False	True

The SVC [20] from the Scikit-Learn library was used as the SVM classifier. The same process of training occurs as in the GB model.

5 Results and Discussion

To validate the proposed machine learning models' performance four metrics were taken into account: recall, precision, f1-score, and accuracy.

In PdM, the recall, precision, and f1-score metrics are essential to validate the proposed models, since in this type of problem classes are almost always not balanced. As a result, recall allows to have an idea of how many correct predictions (i.e., true positives) to incorrect predictions (i.e., false negatives) are being made by the proposed models. False negatives can be can have major repercussions until the machine is not detected to have a failure, such as the manufacturing of defective items, resulting in a loss of money, resources, and time. Furthermore, false alarms (i.e., false positives) also need to be considered in order to minimize unnecessary maintenances activities, thus why the precision metric is also taken into account. To consolidate both recall and precision, the f1-score is also used to validate the proposed machine learning models. The last metric, accuracy, describes the overall ratio of the truly predicted test cases to all the test cases.

The performance metrics for the best GB and SVM models, using the optimal hyperparameters found in Table 1 and Table 2, respectively, are presented in Table 3.

Table 3. Performance metrics for the best machine learning models found

Model	Recall	Precision	F1-score	Accuracy
Gradient boosting	0.87	0.34	0.49	94.55%
Support vector machine	0.92	0.24	0.38	91.00%

From the results represented in Table 3, it is clear that each model has its own advantages with the SVM being better at predicting whenever there is a machine failure, while the GB excels at reducing the number of false alarms. Therefore, if maintenance activities use few resources to check whenever a machine has truly failed and undetected machine failures could lead to major repercussions, the SVM is the recommended option. However, in case there is a higher need to reduce the number of false alarms, then the GB is ideal.

Table 4 and Fig. 3 present the GB and SVM confusion matrixes and actual/predicted values, respectively. Figure 3 only presents 200 samples out of the 2000 of the testing data due to space limitations and to better demonstrate the obtained results. It is noteworthy that there is a trade-off between correctly predicting a machine failure and giving false alarms, with the GB having 3 more unsuccessful machine failure predictions than the SVM, but having 74 fewer false alarms.

It is worth noting that another work, available in [7] used the same dataset as in the present paper to validate the application of a bagged trees ensemble classifier. However, due to a lack of good practices, such as using the whole dataset for training and testing,

Table 4. Gradient boosting and support vector machine confusion matrix

Predicted		Actual	
		Failure	Non-failure
Gradiente boosting	Failure	53	101
	Non-failure	8	1838
Support vector machine	Failure	56	175
	Non-failure	5	1764

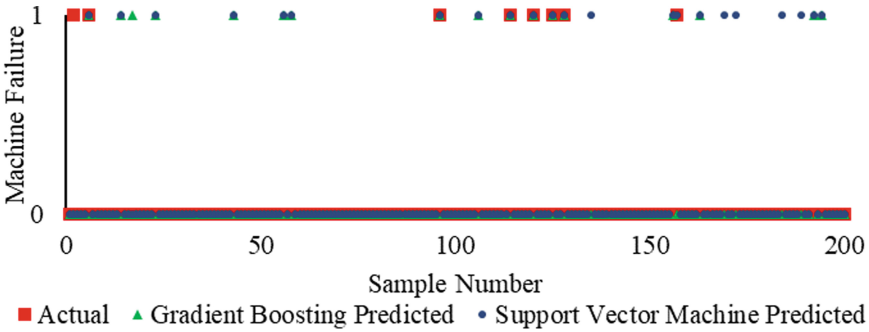


Fig. 3. Actual values and their respective predicted values from the gradient boosting and support vector machine models (200 samples out of 2000 of the testing data)

instead of dividing a part of the dataset for training and another for testing, said work was not used to compare to the models present in this paper. Nevertheless, it is worth noting that even though the other work inflated their obtained results due to overfitting, the present paper still achieved a better recall score of 0.92 when compared to theirs of 0.71.

6 Conclusions

Predictive maintenance strategies have become essential for the manufacturing sector, as it allows maintenances to be conducted only when it's really necessary. These strategies not only reduce unnecessary maintenance costs but also help maintain high product quality.

Accordingly, the methodology proposed in the present paper focuses on implementing and exploring a novel training methodology for machine learning models, more precisely a GB and a SVM models, for PdM problems. In addition, it proposes an automatic hyperparameter optimization approach for the implemented models.

The results obtained, using a synthetic dataset for PdM, show the robustness of the proposed methodology in dealing with unbalanced datasets. The best GB model created achieved a recall of 0.87, a precision of 0.34 leading to an f1-score of 0.49, and an accuracy of 94.55%. In turn, the best SVM model accomplished a recall of 0.92, a precision of 0.24, thus an f1-score of 0.38, and an accuracy of 91.00%, demonstrating that the GB is better at reducing false positives but has worse performance at detecting machine failures (i.e., true positives) than the SVM.

As future work, the authors will explore more machine learning models that could further improve performance, for instance, the usage of an artificial neural network and random forest models. Moreover, using the same dataset and methodology, a multi-class classification problem will be explored and implemented, allowing to not only predict when a machine will fail but also to identify the type of machine failure.

Acknowledgments. The present work has received funding from European Regional Development Fund through COMPETE 2020 - Operational Programme for Competitiveness and Internationalisation through the P2020 Project MUWO (ANI|P2020 POCI-01-0247-FEDER-069417), and has been developed under the EUREKA - ITEA3 Project MUWO (ITEA-19022). We also acknowledge the work facilities and equipment provided by GECAD research center (UIDB/00760/2020) to the project team.

References

1. Sharma, A., Yadava, G.S., Deshmukh, S.G.: A literature review and future perspectives on maintenance optimization. J. Qual. Maintenance Eng. **17**(1), 5–25 (2011). https://doi.org/10.1108/13552511111116222
2. Faccio, M., Persona, A., Sgarbossa, F., Zanin, G.: Industrial maintenance policy development: a quantitative framework. Int. J. Prod. Econ. **147**, 85–93 (2014). https://doi.org/10.1016/j.ijpe.2012.08.018
3. Mota, B., Gomes, L., Faria, P., Ramos, C., Vale, Z., Correia, R.: Production line optimization to minimize energy cost and participate in demand response events. Energies **14**(2), 462 (2021). https://doi.org/10.3390/en14020462
4. Ramos, C., Barreto, R., Mota, B., Gomes, L., Faria, P., Vale, Z.: Scheduling of a textile production line integrating PV generation using a genetic algorithm. Energy Rep. **6**, 148–154 (2020). https://doi.org/10.1016/j.egyr.2020.11.093
5. Garg, A., Deshmukh, S.G.: Maintenance management: literature review and directions. J. Qual. Maintenance Eng. **12**(3), 205–238 (2006). https://doi.org/10.1108/13552510610685075
6. Shin, J.H., Jun, H.B.: On condition based maintenance policy. J. Comput. Des. Eng. **2**(2), 119–127 (2015). https://doi.org/10.1016/j.jcde.2014.12.006
7. Matzka, S.: Explainable artificial intelligence for predictive maintenance applications. In: Proceedings - 2020 3rd International Conference on Artificial Intelligence for Industries, AI4I 2020, pp. 69–74 (2020). https://doi.org/10.1109/AI4I49448.2020.00023
8. Carvalho, T.P., Soares, F.A.A.M.N., Vita, R., da Francisco, R.P., Basto, J.P., Alcalá, S.G.S.: A systematic literature review of machine learning methods applied to predictive maintenance. Comput. Ind. Eng. **137**, 106024 (2019). https://doi.org/10.1016/j.cie.2019.106024
9. Udo, W., Muhammad, Y.: Data-driven predictive maintenance of wind turbine based on SCADA Data. IEEE Access **9**, 162370–162388 (2021). https://doi.org/10.1109/ACCESS.2021.3132684
10. Calabrese, M., et al.: SOPHIA: An event-based IoT and machine learning architecture for predictive maintenance in industry 4.0. Information **11**(4), 202 (2020). https://doi.org/10.3390/info11040202

11. Chaudhuri, A.: Predictive Maintenance for Industrial IoT of Vehicle Fleets using Hierarchical Modified Fuzzy Support Vector Machine. https://arxiv.org/abs/1806.09612v1 (2018). Accessed 28 Dec 2021
12. Mathew, J., Luo, M., Pang, C.K.: Regression kernel for prognostics with support vector machines. In: IEEE International Conference on Emerging Technologies and Factory Automation, pp. 1–5. ETFA (2017). https://doi.org/10.1109/ETFA.2017.8247740
13. Zonta, T., da Costa, C.A., da Rosa Righi, R., de Lima, M.J., da Trindade, E.S., Li, G.P.: Predictive maintenance in the Industry 4.0: a systematic literature review. Comput. Ind. Eng. **150**, 106889 (2020). https://doi.org/10.1016/j.cie.2020.106889
14. Frank, A., Asuncion, A.: {UCI} Machine Learning Repository. https://archive.ics.uci.edu/ml/index.php (2010)
15. Matzka, S.: UCI Machine Learning Repository: AI4I 2020 Predictive Maintenance Dataset Data Set. https://archive.ics.uci.edu/ml/datasets/AI4I+2020+Predictive+Maintenance+Dataset (2020)
16. imbalanced-learn documentation — Version 0.9.0 (2022). https://imbalanced-learn.org/stable/. Accessed 20 Apr 2022
17. sklearn.model_selection.RandomizedSearchCV — scikit-learn 1.0.2 documentation (2011). https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html. Accessed 20 Apr 2022
18. scikit-learn: machine learning in Python — scikit-learn 1.0.2 documentation (2011). https://scikit-learn.org/stable/index.html. Accessed 20 Apr 2022
19. sklearn.ensemble.GradientBoostingClassifier — scikit-learn 1.0.2 documentation (2011). https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html. Accessed 29 Apr 2022
20. sklearn.svm.SVC — scikit-learn 1.0.2 documentation (2011). https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html. Accessed 29 Apr 2022

Estimation of Lamb Weight Using Transfer Learning and Regression

Virginia Riego del Castillo[1], Lidia Sánchez-González[1](✉),
Laura Fernández-Robles[1], Manuel Castejón-Limas[1], and Rubén Rebollar[2]

[1] Departamento de Ingenierías Mecánica, Informática y Aeroespacial,
Universidad de León, 24071 León, Spain
{vriec,lidia.sanchez,l.fernandez,manuel.castejon}@unileon.es

[2] Group in Product and Project Management, Escuela de Ingeniería y Arquitectura,
Universidad de Zaragoza, c/María de Luna 3, C.P. 50018 Zaragoza, Spain
rebollar@unizar.es

Abstract. Meat production needs of accurate measurement of livestock weight. In lambs, traditional scales are still used to weigh live animals, which is a tedious process for the operators and stressful for the animal. In this paper, we propose a method to estimate the weight of live lambs automatically, fast, non-invasive and affordably. The system only requires a camera like those that can be found in mobile phones. Our approach is based on the use of a known Convolutional Neural Network architecture (Xception) pre-trained on the ImageNet dataset. The acquired knowledge during training is used to estimate the weight, which is known as transfer learning. The best results are achieved with a model that receives the image, the sex of the lamb and the height from where the image is taken. A mean absolute error (MAE) of 0.58 kg and an R^2 of 0.96 were obtained, improving on current techniques. Only one image and two values specified by the user (sex and height) allow to estimate with a minimum error the optimal weight of a lamb, maximising the economic profit.

1 Introduction

Intelligent systems have been applied to many fields to assist daily routines and make them easier and faster [14]. This is also the case of animal production, where traditional farms are optimising their production, improving animal welfare and increasing their benefit [9].

There are many works in the literature that propose approaches based on the use of intelligent techniques to assist farmers in their daily routines. So, there are studies to analyse the behaviour of the animals to detect diseases or to achieve a better understanding about the most influence aspects to improve the animal welfare [2,4]. For cows, weight has been predicted with regression techniques provided by SciKit-Learn library by using different body size measurements and age [17]. However, most of them are focused on species like pigs or cows as it is a more profitable industry. On the contrary, there are not so many applications

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 23–30, 2023.
https://doi.org/10.1007/978-3-031-18050-7_3

for lambs, perhaps because they present more difficulties due to their fast movements and a lower economic profit. As this economic profit is directly related to determining the proper time for slaughter, it is crucial to identify when a lamb has reached its optimum weight. In order to weigh animals automatically, computer vision techniques have been employed to different species like pigs [1], cows [15] or sheep [7].

Regarding the techniques used, most of them apply traditional computer vision techniques like morphological operations [1,10,12,18]. More recently, Convolutional Neural Networks (CNN) have been also employed to estimate body weight of pigs [16]. For sheep, image segmentation using an auto-encoder is proposed in [20], achieving a R^2 of 0.80 for weight estimation.

Although some methods are based on the use of certain cameras like laser [21] or thermal ones [8], most of them employ traditional cameras or depth-cameras like Kinetic [1] or Intel® Real Sense$^{\text{TM}}$ [18]. It is important to notice that most of these methods require special infrastructures to acquire the images named walk-over weighing (WoW) systems; that is the case of the method proposed in [19], where only 32 images of sheep are considered and they extracted different features to predict the sheep weight by using regression methods, obtaining a MAE of 3.099 (1.52) kg and an Adjusted R^2 of 0.687. For that reason, they are not so widely adopted by farmers as they require an important investment, specially in lamb industry [11].

This paper proposes a method for weighing lambs automatically using computer vision techniques. This proposal only requires a system to acquire images from a zenithal view, which can be a mobile phone or other electronic device. So, the implementation cost is really reduced and can be easily adopted by farmers. By using the proposed approach, lamb weight is estimated using a CNN that extracts the body contour of the lamb and assesses its weight. As a result, farmers can use a friendly application to determine automatically the proper time for slaughter based on the weight of the lamb. In addition to this, as lambs do not suffer stress since the image acquisition employs an indirect process, animal welfare is increased. Farmers also avoid the use of traditional scales which is a tedious and physical demanding process.

This paper is structured as follows. Section 2 explains the dataset and how it has been acquired. Section 3 gives details of the proposed architecture and Sect. 4 shows the results obtained that validate the proposed architecture. Finally, Sect. 5 gathers the achieved conclusions.

2 Image Acquisition and Data Preparation

Zenithal images of the Rasa Aragonesa breed lambs were taken during the experiments of [18]. This breed is usually distributed in the northeast of Spain and these farms require to maximise the number of births so as to sell more lambs and increase the benefit. A decisive factor that affects the number of births is the weight.

For each image, we know the sex (Male or Female), the live weight and an identifier of the lamb. The acquired system is equipped with a 3D Intel Camera, which allows the acquisition of color images (RGB channels) and depth images. Figure 1 shows the process to estimate the distance from them lamb at which the image was taken (we name it *height*). Firstly, the depth image is binarized with a threshold (900, which is the mean distance to the lamb). Secondly, a mask is used to detect the region of the image where the lamb is (the bounding box that comprises the body lamb). Then, the bounding box is expanded by 50 pixels and this new area is used to crop the image and get the region of interest. Finally, intrinsic camera parameters are used to get the relationship between a stream's 2D and 3D coordinate systems. This process is known as deprojection, which consists on converting pixels of two dimensions (x, y) to points of the real world with three dimensions (x, y, z), being z the distance between the camera and the ground. This value is converted to meters and called Height. In a simplified and affordable acquisition system as a mobile phone, the farmer must introduce manually the height at which the image was taken.

For this experiment, 54 lambs and a total of 2030 images were taken into account. Figure 2 shows the distribution of the 2030 images according to its sex (Female or Male) according to the weight of the lamb (between 13.5 kg and 27.7 kg) and the height of the camera (between 1.16 m and 1.55 m). Some sample images of the dataset with the weight and calculated height of the camera can be seen in Fig. 3.

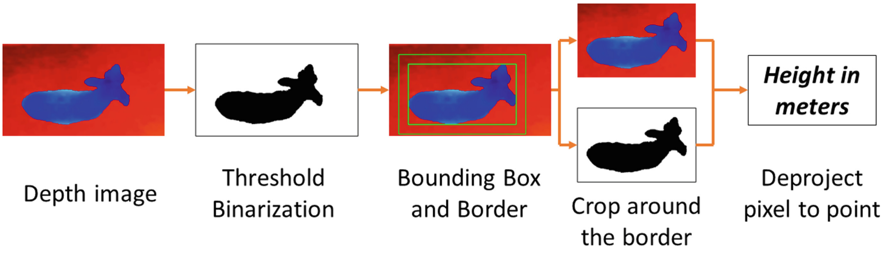


Fig. 1. Scheme of the lamb's height computation. Depth image is binarized to detect the lamb (Bounding Box). Image is cropped with a border of 50 pixels around the bounding box. Deprojection of pixels is used to calculate the height of the camera.

3 Proposed Architecture

In this paper, we propose the use of CNN to estimate the weight lamb from images. Our approach consists in the use of transfer learning, which is a technique that considers the weights of a pre-trained model as well as the extracted features and knowledge and apply all of them to a new task. We have used the Xception [5] architecture which has 22.9 millions of parameters and a depth of 81 layers. This architecture has been trained on the ImageNet dataset [6] which has 3.2 million of images and counts with a subtree category of mammals (862K of images). For this dataset, Xception achieves an accuracy of the 79%, and a Top-5 accuracy (which checks the top 5 predictions with the target label) of 94.5%.

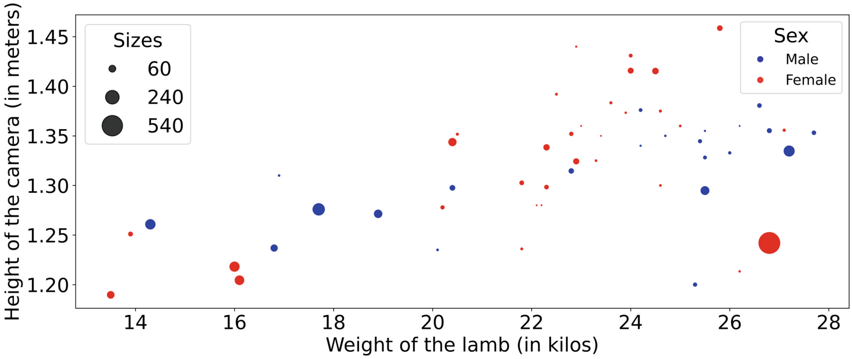
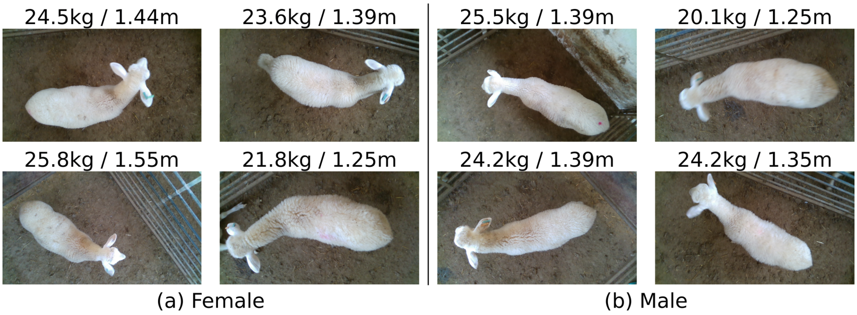


Fig. 2. Distribution of the images divided by sex (Male or Female) according to the weight of the lamb and the height of the camera.



(a) Female

(b) Male

Fig. 3. Image samples of the dataset for both female (left) and male (right) lambs with their weight and calculated height.

Figure 4 shows the proposed architecture, which uses the knowledge acquired during the training of Xception to classify ImageNet images. Fully-Connected layers are removed and then a fully connected layer of 128 neurons and ReLu activation is added. In addition to this, a last layer with one neuron which estimates the weight is also included. In our approach, inputs are formed by color images with a resolution of 240×424 pixels. Using CNN, features from the images are extracted. These features are the inputs of the fully connected layer. Besides that, two more inputs are considered: the sex of the lamb and the height that is defined as the distance from the camera to the ground. By applying this model a regression output is obtained that yields the lamb weight.

Let N be the number of samples, y the real weight and $\hat{y}$ the predicted weight, there are multiple regression metrics used to evaluate results:

- R Squared (R2), also known as coefficient of determination, measures the variability of a dependent variable. Values can go from 0 to 1, although can be interpreted as a percentage.

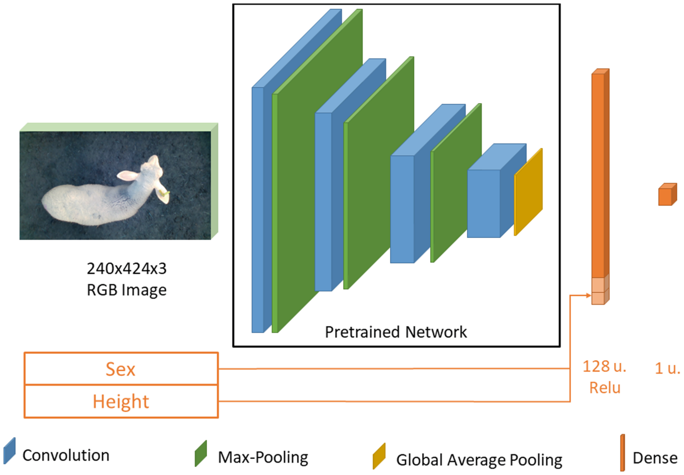


Fig. 4. Network architecture defined for weight estimation where the Xception model trained on ImageNet is employed to extract the features from the image that are combined with sex and height features to estimate the lamb weight

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{1}$$

- Mean Absolute Error (MAE) is the mean of the absolute error between the predicted and real values.

$$MAE = \frac{1}{N}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2}$$

- Mean Squared Error (MSE) is the mean of the square error between the predicted and real values.

$$MSE = \frac{1}{N}\sum_{i=1}^{n}|y_i - \hat{y}_i|^2 \tag{3}$$

- Mean Absolute Percentage Error (MAPE) is the ratio of the mean error to real values.

$$MAPE = \frac{1}{N}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i} \tag{4}$$

4 Experimental Results

The goal of the following experiment is to estimate the weight of a lamb automatically using an affordable system for farmers, such as a mobile phone.

The proposed architecture, explained in the previous Section, has been trained during 50 epochs using a batch size of 16 images. The pre-trained model has been frozen to avoid losing the previously acquired knowledge. Adam [13] is used as an optimizer, the learning rate is 10^{-4}, and the loss function is MAE. The dataset explained in Sect. 2 was split into training set (70% of the data) and test set (30% of data).

Multiple experiments were carried out: a model with just the image as input, a model with one additional input besides the image (sex or height) and a model with both additional inputs (sex and height). As the height values range between 1.16 m and 1.55 m, a normalisation has been done by subtracting one unit (0.16–0.55).

Table 1 summarises all the experiments and their results. Best results were achieved with the model that includes as inputs the image, the lamb sex and the normalised height, getting a MAE of 0.48 kg, a MAPE of 2.22% and an R^2 of 0.97 in the training set. The results of the test set provide evidence of the generalisation of the proposed system with a MAE of 0.59 kg, a MAPE of 2.84% and an R^2 of 0.96.

Table 1. Results of the experiments taken with our proposed architecture.

		Basic model	Model with sex	Model with height	Model with normalised height	Model with sex and heightZ	Model with sex and normalised height
Train	MAE	0.5877	0.4965	0.5207	0.5293	0.4817	**0.4760**
	MSE	1.0430	0.7708	0.8257	0.8509	0.6966	**0.6664**
	MAPE	2.6963	2.2979	2.4020	2.4405	2.2388	**2.2174**
	R2	0.9534	0.9656	0.9631	0.9620	0.9689	**0.9703**
Test	MAE	0.7077	0.6135	0.6319	0.6681	0.6010	**0.5896**
	MSE	1.3692	1.0468	1.1064	1.1721	0.9610	**0.9223**
	MAPE	3.3696	2.9437	3.0224	3.2224	2.9108	**2.8471**
	R2	0.9412	0.9551	0.9525	0.9497	0.9587	**0.9604**

These results outperform the ones obtained in [18], where the estimation weight was applied using a depth camera, as their method yielded a MAE of 1.37 kg and a R^2 of 0.86. Therefore, our proposal achieves better results with a more affordable system that can be supported by a regular mobile phone. All experiments are available in [3].

5 Conclusions

This paper proposes a deep learning model to help farmers to estimate the weight of live lambs. The presented system can be used in devices with camera, such as mobile phones, to take zenithal pictures of a lamb. A CNN architecture named Xception trained on ImageNet dataset has been considered to use transfer learning. This allows to take advantage of the previous knowledge to apply it to a regression problem, like the estimation of weight. Some external information can be included by the mobile application, such as the sex of the lamb and the distance from the mobile phone to the ground (named height), what has improved the obtained results. Model evaluation achieves a MAE of 0.59 kg (which corresponds with a MAPE of 2.84 %) and an R^2 of 0.96 on the test set. If additional data is not included and the estimation is made just with the acquired image, MAE is 0.71 kg and R^2 is 0.94. Experts consider these results adequate for the need of livestock farms in terms of accuracy, as well as the benefits of the easy implementation as it just requires a mobile device. Another remarkable advantage of the proposed method is that it reduces the human-animal interaction which increases the animal welfare. Future work will be focused on the design of a data-driven system to analyse in real time the weight progression of individual lambs that can be used to detect disease or predict the most appropriate time for slaughter.

Acknowledgements. This research was funded by the Aragonese government and the "Asociación Nacional de Criadores de Ganado Ovino de Raza Rasa Aragonesa (ANGRA)" through the "Programa de Desarrollo Rural de Aragón para el periodo 2014-2020 (GCP2017 – 2700)". Virginia Riego would like to thank Universidad de León for its funding support for her doctoral studies.

References

1. Alsahaf, A., Azzopardi, G., Ducro, B., Hanenberg, E., Veerkamp, R.F., Petkov, N.: Estimation of muscle scores of live pigs using a kinect camera. IEEE Access **7**, 52238–52245 (2019). https://doi.org/10.1109/ACCESS.2019.2910986
2. Byrne, D.T., Esmonde, H., Berry, D.P., McGovern, F., Creighton, P., McHugh, N.: Sheep lameness detection from individual hoof load. Comput. Electron. Agricult. **158**, 241–248 (2019). https://doi.org/10.1016/j.compag.2019.01.048
3. Riego del Castillo, V.: Transfer Learning to estimate lamb weight. https://github.com/uleroboticsgroup/transfer_learning_lamb_weight
4. Chen, C., Zhu, W., Guo, Y., Ma, C., Huang, W., Ruan, C.: A kinetic energy model based on machine vision for recognition of aggressive behaviours among group-housed pigs. Livestock Sci. **218**(2017), 70–78 (2018). https://doi.org/10.1016/j.livsci.2018.10.013
5. Chollet, F.: Xception: deep learning with depthwise separable convolutions (2016). https://doi.org/10.48550/ARXIV.1610.02357
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848
7. González-García, E., et a.: A mobile and automated walk-over-weighing system for a close and remote monitoring of liveweight in sheep. Comput. Electron. Agricult. **153**, 226–238 (2018). https://doi.org/10.1016/j.compag.2018.08.022
8. Halachmi, I., Klopčič, M., Polak, P., Roberts, D., Bewley, J.: Automatic assessment of dairy cattle body condition score using thermal imaging. Comput. Electron. Agricult. **99**, 35–40 (2013). https://doi.org/10.1016/j.compag.2013.08.012
9. Hostiou, N., et al.: Impact of precision livestock farming on work and human-animal interactions on dairy farms. A review. Biotechnol. Agron. Soc. Environ. **21**(4), 268–275 (2017). https://doi.org/10.25518/1780-4507.13706

10. Jun, K., Kim, S.J., Ji, H.W.: Estimating pig weights from images without constraint on posture and illumination. Comput. Electron. Agricult. **153**(August), 169–176 (2018). https://doi.org/10.1016/j.compag.2018.08.006
11. Kaler, J., Ruston, A.: Technology adoption on farms: using normalisation process theory to understand sheep farmers' attitudes and behaviours in relation to using precision technology in flock management. Prevent. Vet. Med. **170**, 104715 (2019). https://doi.org/10.1016/j.prevetmed.2019.104715
12. Kashiha, M.,et al.: Automatic weight estimation of individual pigs using image analysis. Comput. Electron. Agricult. **107**, 38–44 (2014). https://doi.org/10.1016/j.compag.2014.06.003
13. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, 7–9 May 2015, Conference Track Proceedings (2015). arXiv:1412.6980
14. Niloofar, P., et al.: Data-driven decision support in livestock farming for improved animal health, welfare and greenhouse gas emissions: overview and challenges. Comput. Electron. Agricult. **190**, 106406 (2021). https://doi.org/10.1016/j.compag.2021.106406
15. Nir, O., Parmet, Y., Werner, D., Adin, G., Halachmi, I.: 3D Computer-vision system for automatically estimating heifer height and body mass. Biosyst. Eng. **173**, 4–10 (2018). https://doi.org/10.1016/j.biosystemseng.2017.11.014
16. Rodríguez Alvarez, J., et al.: Body condition estimation on cows from depth images using convolutional neural networks. Comput. Electron. Agricult. **155**, 12–22 (2018). https://doi.org/10.1016/j.compag.2018.09.039
17. Ruchay, A.N., Kolpakov, V.I., Kalschikov, V.V., Dzhulamanov, K.M., Dorofeev, K.A.: Predicting the body weight of hereford cows using machine learning. IOP Conf. Ser. Earth Environ. Sci. **624**(1), 012056 (2021). https://doi.org/10.1088/1755-1315/624/1/012056
18. Samperio, E., Lidón, I., Rebollar, R., Castejón-Limas, M., Álvarez-Aparicio, C.: Lambs' live weight estimation using 3D images. Animal **15**(5), 100212 (2021). https://doi.org/10.1016/j.animal.2021.100212
19. Sant'Ana, D.A., et al.: Weighing live sheep using computer vision techniques and regression machine learning. Mach. Learn. Appl. **5**, 100076 (2021). https://doi.org/10.1016/j.mlwa.2021.100076
20. Shah, N.A., Thik, J., Bhatt, C., Hassanien, A.-E.: A deep convolutional encoder-decoder architecture approach for sheep weight estimation. In: Chiplunkar, N.N., Fukao, T. (eds.) Advances in Artificial Intelligence and Data Engineering. AISC, vol. 1133, pp. 43–53. Springer, Singapore (2021). https://doi.org/10.1007/978-981-15-3514-7_4
21. Yoshida, K., Kawasue, K.: Robust 3D pig measurement in pig farm. In: Leal-Taixé, L., Roth, S. (eds.) ECCV 2018. LNCS, vol. 11129, pp. 387–400. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11009-3_23

UAV Simulation for Object Detection and 3D Reconstruction Fusing 2D LiDAR and Camera

Daniel Amigo(✉), Jesús García, José M. Molina, and Jorge Lizcano

GIAA, University Carlos III of Madrid, Madrid, Spain
{damigo,jgherrer}@inf.uc3m.es, molina@ia.uc3m.es

Abstract. Currently it is hard to develop UAV in civil environments, being simulation the best option to develop complex UAV missions with AI. To carry out useful AI training in simulation for real-world use, it is best to do it over a similar environment as the one a real UAV will work, with realistic objects in the scene of interest (buildings, vehicles, structures, etc.). This work aims to detect, reconstruct, and extract metadata from those objects. A UAV mission was developed, which automatically detects all objects in a given area using both simulated camera and 2D LiDAR, and then performs a detailed scan of each object. Later, a reconstruct process will create a 3D model for each one of those objects, along with a georeferenced information layer that contains the object information. If applied on reality, this mission ease bringing real content to a digital twin, thus improving, and extending the simulation capabilities. Results show great potential even with the current budget specification sensors. Additional post-processing steps could reduce the resulting artefacts in the export of 3D objects. Code, dataset, and details are available on the project page: https://danielamigo.github.io/projects/soco22/.

Keywords: Object reconstruction · LiDAR-camera fusion · UAV simulation · Object detection · AirSim

1 Introduction

Unmanned Aerial Vehicles (UAV) are powerful tools capable of autonomously capturing the Earth at a given time. However, at present they are a technology that is neither smart nor robust enough to operate in cities along humans and are thus considered potential hazards. They are strongly legislated to restrict its use in habited areas [1, 2]. Eventually this risk will disappear as UAV get smarter. For this reason, work must be pursued in order to operate them safely around humans while performing complex and risky tasks [3].

Software in the loop (SITL) simulation is the easiest way for develop and test complex UAV missions with zero damage. AirSim [4] is one of the most popular UAV simulator. Based on the popular Unreal Engine (UE4), it provides high visual and customization capabilities, enabling to simulate sensors of the UAV such as lidar or cameras, or even to deploy multiple UAVs at the same time for swarm interactions. AirSim is compatible

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 31–40, 2023.
https://doi.org/10.1007/978-3-031-18050-7_4

with the highly used for Pixhawk 4 flight controller (PX4) [5]. All these features make it perfect for testing UAV missions with Artificial Intelligence (AI) before applying them on the real world.

A common problem in AI is introducing data into a model that is not close to the data used in training. Using simulation, it is easy to solve this problem for real-world deployment. If the simulated mission environment is like the real world, the data captured by the simulated UAV will be similar too and therefore good for AI training a model that aims to perform in reality [6]. It is possible to fly over photorealistic environments on AirSim, but those won't be as real environments. Ideally, the environment should be a digital representation of your own real environment, a digital twin. Creating them is a very complex and hard task [7]. It should recreate all static objects and habitual patterns of dynamics actors, such as people, vehicles, or animals. To generate digital twins of any location in the world the only feasible option is automatization using huge amounts of geolocated data.

This work is a further step of [8] and [9], where we detected a specific object and geolocate it to enhance a digital twin. Now using simulation, we first create data with an autonomous flight mission but also automatically create 3D representations of any environment object using only UAVs. With it, a further process can automatically add those objects on the digital twin, improving it for future simulation applications. The proposal tests realistic sensing using low-cost 2D LiDAR and an HD FPV camera. LiDAR sensor is used first to gather information of all the objects in a specific area. The UAV will then use its on-board capabilities to identify them and design a customized mission for each object detected, scanning it in detail with both sensors. After the flight, the 3D object reconstruction process fuses both sensors by coloring each LiDAR detection using the camera data and clean the data to get the object. Finally, it transforms the final point cloud into a 3D object. The knowledge extracted of each object (position, height and the other metadata as the object type generated by image and point cloud classifiers) is also stored in a GIS format for other future uses.

The results obtained are promising. The mission gathering component works as expected, perfectly identifying all objects in the mission, and performing a close and custom scanning for each object with both sensors. The 3D object reconstruction component results are good but could be improved with further post-processes or fusion with other algorithms. In any case, the proposed mission automatically successfully solves an existing problem, easing the generation of digital twins. In conclusion, it has been exemplified how a UAVs work designed in this simulation framework can be successfully developed, reducing the friction when performing it in real drones.

2 Related Works

This section briefly introduces several studies of other researchers regarding the 3D object reconstruction and of its detection using both point clouds and imagery.

Although there are many sensors [10], not all of them are suitable for use in current UAVs due to their size or weight. Object reconstruction can be performed only using camera data with algorithms as Structure from Motion (SfM). Its visual result is good, but the geometry is not reliable. It can also be done by LiDAR, generating precise geometry but lacking color or texture. Many researchers use drones for large areas reconstruction, but not specifically for object reconstruction. For example, [11] uses LiDAR to precisely map an excavated surface. With this approach no objects are scanned in detail, only a global view, so it is not ideal for digital twins [12]. [13] proposal is the only one found attempting to reconstruct a specific object with a UAV mission. They use a camera and SfM to reconstruct.

Although camera and LiDAR can work separately, it is typical they are fused for these tasks as they have high synergy. The image contains a lot of information with high detail and color, while the LiDAR is composed of lots of individual measurements, highly detailed in shape, but colorless. For example, [14] uses the point cloud from a Terrestrial Laser Scanning to improve the geometry of the point cloud obtained from SfM with a drone flight, obtaining very good results.

Despite the approach or the sensor, the goal is to generate only a 3D representation of an object, so it requires segmentation algorithms for discarding the information that does not belong to the object, as the floor, walls, or other objects. This task can be performed before the main reconstruction, by removing junk information from each raw data, or after generating the 3D point cloud, by detecting specific point cloud points not belonging to the object.

On the other hand, our proposal aims to detect objects in real time and at the end classify them in order to generate additional metadata. The object detection problem is a common task when dealing with point clouds. Clustering and segmentation algorithms can easily discriminate and group them to achieve the desired solution. The object classification approaches use deep learning to train convolutional networks for detecting patterns in images or point clouds relating a specific class. There are few researches, such as [15], that try to combine both in the same procedure. Image-based solutions are widely studied whereas the 3D solutions are relatively unexplored.

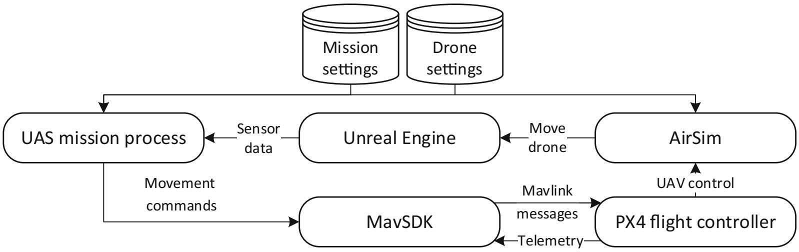


Fig. 1. Simulation framework interconnection diagram

3 Simulation Framework

Several components need to be combined in order to make this realistic simulation work correctly. This section explains the key aspects of each component and its logical connection with the others. A diagram illustrating the interactions between all components is provided on Fig. 1.

First, it is needed to define the mission to perform; what will the UAV do, with which sensors and where. A JSON file defining the drones' characteristics must be provided to AirSim, the central element of the framework. The UAV mission is designed as fully autonomous, simulating a UAV on-board processor that receives all sensors data and send to the flight controller the movements commands when needed, according to the mission stage. The mission process needs to connect both with AirSim to retrieve sensors data capturing the Unreal Engine environment at a specific time and place, and with MAVSDK library to communicate with PX4, to send those custom movement commands but also to receive telemetry updates to make onboard missions adjustments.

Furthermore, we have the UE4 environment, built around the Cesium plugin for Unreal. It adds a digital twin of the whole Earth, with its texture formed by satellite images and a global elevation model. It also adds a global coordinate system allowing to match the PX4 coordinates with the digital twin, making it possible to simulate UAV flights over real locations on the Cesium virtual Earth. Note that it is only a template, it does not bring the actual objects and dynamic behaviors into the simulator by default.

4 Proposed Process

This section introduces the process to automatically detect and reconstruct 3D objects using LiDAR and camera fusion onboard of a UAV. The process is composed by two main blocks. The first, detailed in Fig. 2, performs a completely autonomous UAV mission first detecting all objects in a specific area and then performing a custom mission to scan in detail each object. Then, detailed on Fig. 4, an offline process creates the 3D mesh by colorizing the LiDAR point cloud and discarding all points which are not part of the object. The output is the mentioned 3D mesh of each object but also a GIS layer with object metadata.

The mission parameters must be defined manually, both drone settings with the JSON file explained earlier and the parameters of the algorithms to adjust the mission operation. It also requires the area where to scan for objects.

An illustration of the data gathering component is provided in Fig. 3. The first step is to connect with AirSim and MAVSDK, and then to take-off. Once it is done, the mission starts, flying directly to the input area at constant altitude. When the UAV arrives at that position, it starts its sensors and starts the first part of the data gathering mission. It performs a sweep capturing all possible objects within the orbit with both LiDAR and camera. Then, the data is processed onboard to identify the objects placement and dimensions.

After this initial sweep, the process performs a LiDAR points analysis. First it merges all point clouds into one and applies the RANSAC algorithm to discard the floor points from the rest. Then, a DBSCAN algorithm is applied to cluster the point cloud in groups, identifying how many objects are and to know its positioning and dimensions. Using each group point cloud the process can design and perform a custom orbit around the object, from top to bottom, so both sensors capture it perfectly. Once the UAV performs the last orbit almost hitting the ground, it comes back to the origin point following the same initial path.

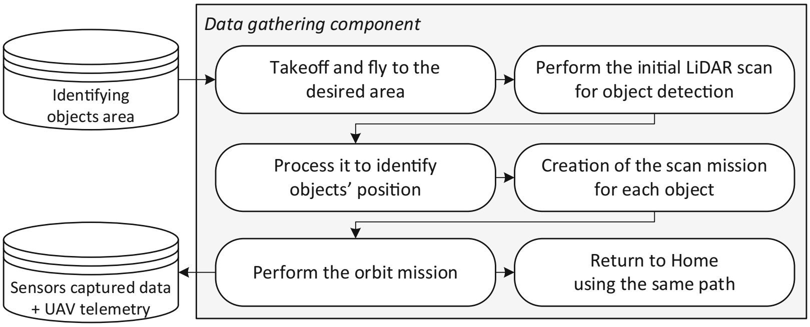

Data gathering component
Identifying objects area
Takeoff and fly to the desired area
Perform the initial LiDAR scan for object detection
Process it to identify objects' position
Creation of the scan mission for each object
Sensors captured data + UAV telemetry
Perform the orbit mission
Return to Home using the same path

Fig. 2. Data gathering component

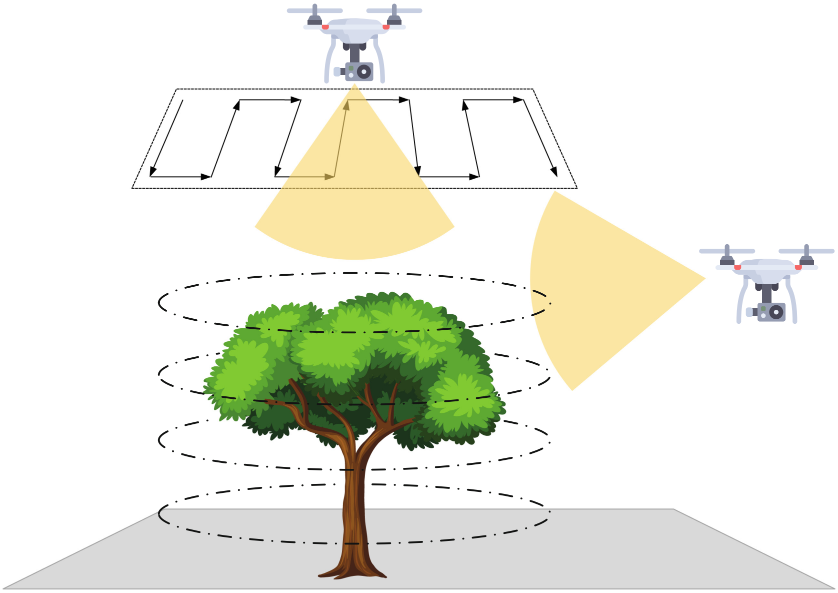


Fig. 3. UAV mission illustration

After all data is captured and the UAV is back, the 3D object reconstruction process is performed. The process gets each capture data at a specific time: a trio formed by the camera snapshot, LiDAR points and the UAV's position and rotation matrix. The aim is to project each one of those LiDAR points into the camera snapshot, obtaining a RGB color for the LiDAR point. To do that, the LiDAR tridimensional point is transformed to the camera coordinate system, and then inserted on the snapshot, using the intrinsic camera parameters.

With all LiDAR points colorized, the next step merges all of them into one point cloud. As done onboard, this detailed point cloud has noise. To discard those non-desired points, it performs first RANSAC algorithm to remove the floor and then DBSCAN to remove other points not from this object.

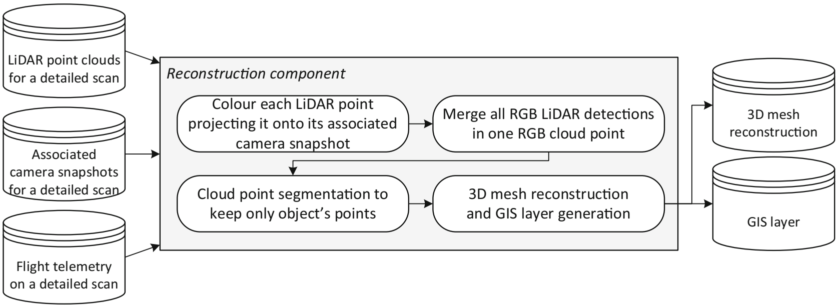

LiDAR point clouds for a detailed scan
Associated camera snapshots for a detailed scan
Flight telemetry on a detailed scan
Reconstruction component
Colour each LiDAR point projecting it onto its associated camera snapshot
Merge all RGB LiDAR detections in one RGB cloud point
Cloud point segmentation to keep only object's points
3D mesh reconstruction and GIS layer generation
3D mesh reconstruction
GIS layer

Fig. 4. Reconstruction process

Finally, the process applies the Poisson Surface Reconstruction algorithm to transform the final colorized point cloud it to a final 3D mesh.

The process at the end also generates a GIS layer with metadata for further uses. Specifically, it adds the object positioning, dimensions, and orientation. It also contains the object type, a useful attribute for all kinds of future processes. It is calculated by applying two classifiers, one image-based, applied to all the snapshots captured during the detailed scan, based on VGG16 and trained with MicroImageNet dataset. The other is a 3D mesh classifier to classify the final 3D object. The classifier is based on PointNet++ [16] and trained with CAD data from ModelNet40 [17]. If both classifiers conclude the object class is the same, that value is added.

5 Demonstration and Evaluation

To demonstrate the potential of the developed system, an example mission of the whole process using the presented simulation framework is performed. It contains three objects in the same area: a sculpture, a building, and a car. They are separated from each other for around 10 m, as illustrated on Fig. 6(a) and Fig. 7. Those will be detected using the sensors illustrated at Fig. 5. They have the following characteristics: The camera (represented in blue) has 1920 × 1080 resolution, 120° horizontal FOV and it is placed as a FPV view, and the 2D LiDAR (represented in green): 10 rotations per second, 30.000 points per second, 165° FOV.

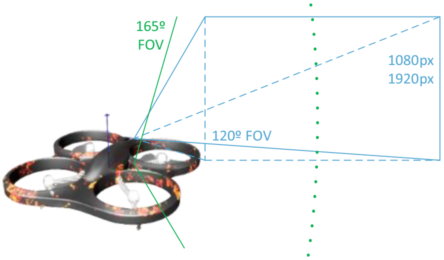
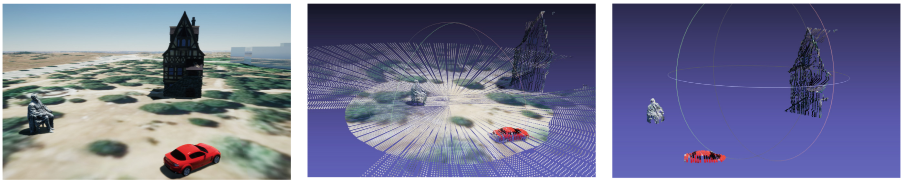


Fig. 5. LiDAR (green) and camera (blue) installed on AirSim's UAV

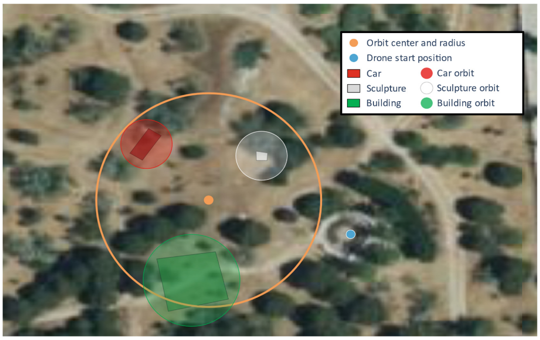
First, the drone performs the object detection, going to the specified position, and making an orbit with a radius of 30 m, obtaining Fig. 6(b) colorized LiDAR point cloud. Then, the point cloud is processed as explained, obtaining three clusters shown in Fig. 6(c). It is noteworthy that the building is not completely detected, which could lead to an inaccurate scanning mission design.

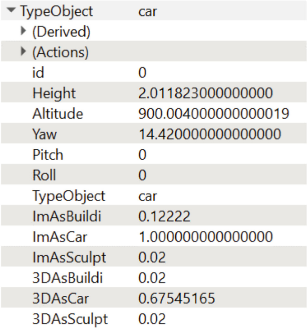
After it, the specific scans are performed with orbits at constant altitude, starting from its height and with one meter spacing between them.



(a) Environment image (b) LiDAR point cloud (c) Segmented point cloud

Fig. 6. Data gathering object detection example

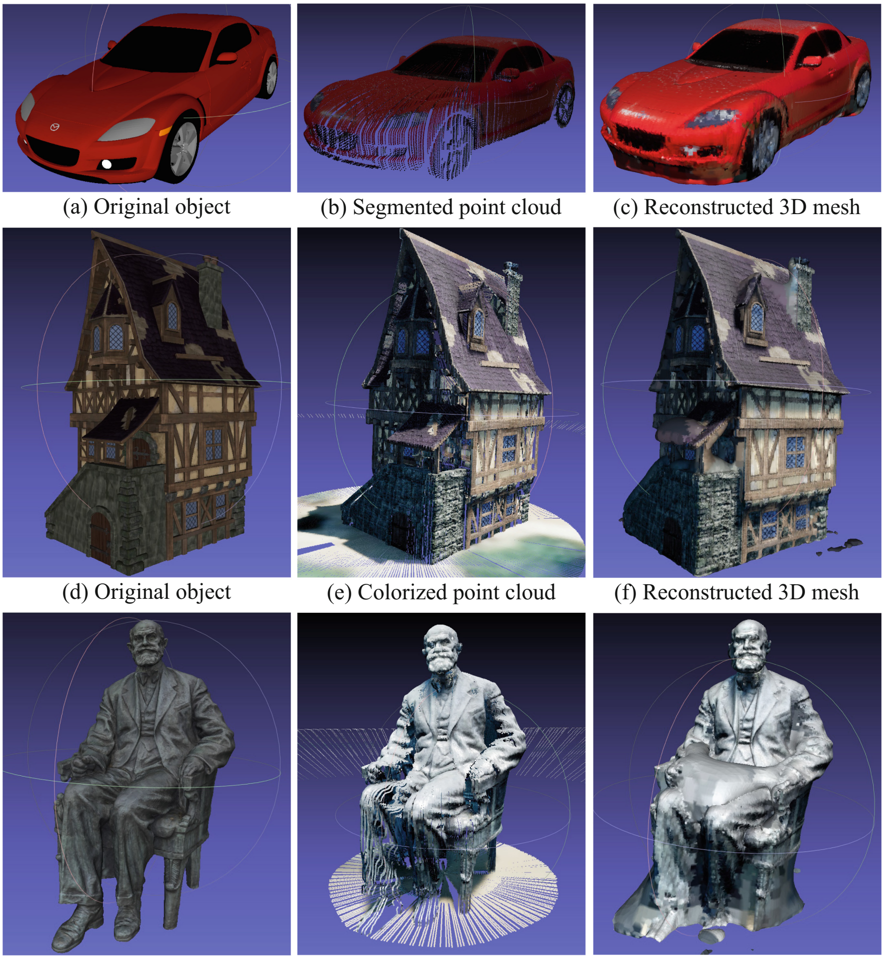




▾ TypeObject	car
▸ (Derived)	
▸ (Actions)	
id	0
Height	2.011823000000000
Altitude	900.004000000000019
Yaw	14.420000000000000
Pitch	0
Roll	0
TypeObject	car
ImAsBuildi	0.12222
ImAsCar	1.000000000000000
ImAsSculpt	0.02
3DAsBuildi	0.02
3DAsCar	0.67545165
3DAsSculpt	0.02

Fig. 7. Orbits and detected objects in GIS (left). Metadata of car GIS layer (right)

The results for each object are illustrated at Fig. 8. A visual analysis reveals gaps in the rows of 2D LiDAR points, which may cause trouble to the mesh generation algorithm. Also, in (c) it is noticeable how the meshes are illumination-dependent, as the car's front has sunlight reflected in the snapshots which is then transferred to the 3D model. In (e) it is observed that LiDAR data is missing in the chimney, although it is fairly well solved in (f). In contrast, in the roof behind the stairs, Poisson Surface Reconstruction generates a kind of imaginary bubble. The same happens in (i). Finally, the bottom of all 3D models is less detailed. This may be justified by the rough cropping of the ground, causing the scan mission not to perform lower orbits.



(a) Original object (b) Segmented point cloud (c) Reconstructed 3D mesh

(d) Original object (e) Colorized point cloud (f) Reconstructed 3D mesh

(g) Original object (h) Colorized point cloud (i) Reconstructed 3D mesh

Fig. 8. Demonstration 3D mesh generation

6 Conclusions and Perspectives

This work proves the potential of designing autonomous drone missions using a realistic simulation framework, which can facilitate the development of complex tasks with AI prior to their actual operation.

The data gathering components work well, detecting and performing a scanning mission using only onboard processing. However, the proposed solution is still at an early stage, and for actual deployment it should be more robust, for example avoiding obstacles during the mission or considering complex cases as whether the object is close to a wall, denying the drone the acquisition of data in such a perspective.

In addition, the proposed 3D reconstruction process works well but improvements are possible. Further work will be applied to quantify the results with different metrics for this colorizing LiDAR points approach. It should also be compared with alternatives such as SfM or deep learning approaches such as NERF, the best quality results may be obtained by combining several of them. Anyway, this way is a good alternative, as it requires low computational resources.

Other future developments must be seeking for a more robust object type classifier. Also, in using the object type in the 3D mesh generation, maybe as a post-processing. Lastly, to incorporate the 3d meshes automatically into the digital twin.

Acknowledgement. This research was partially funded by public research projects of Spanish Ministry of Science and Innovation, references PID2020-118249RB-C22 and PDC2021-121567-C22 - AEI/10.13039/501100011033, and by the Madrid Government (Comunidad de Madrid-Spain) under the Multiannual Agreement with UC3M in the line of Excellence of University Professors, reference EPUC3M17.

References

1. Grigoropoulos, N., Lalis, S.: Flexible deployment and enforcement of flight and privacy restrictions for drone applications. In: 2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W). pp. 110–117. IEEE, Valencia, Spain (2020)
2. Hildmann, H., Kovacs, E.: Review: using unmanned aerial vehicles (UAVs) as mobile sensing platforms (MSPs) for disaster response. Civil Secur. Public Saf. Drones **3**, 59 (2019). https://doi.org/10.3390/drones3030059
3. Kakaletsis, E., et al.: Computer vision for autonomous UAV flight safety: an overview and a vision-based safe landing pipeline example. ACM Comput. Surv. **54**, 1–37 (2022). https://doi.org/10.1145/3472288
4. Shah, S., Dey, D., Lovett, C., Kapoor, A.: AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. ArXiv170505065 Cs (2017)
5. Meier, L., Honegger, D., Pollefeys, M.: PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In: 2015 IEEE International Conference on Robotics and Automation (ICRA). pp. 6235–6240. IEEE, Seattle, WA, USA (2015)
6. Alvey, B., Anderson, D.T., Buck, A., Deardorff, M., Scott, G., Keller, J.M.: Simulated photo-realistic deep learning framework and workflows to accelerate computer vision and unmanned aerial vehicle research. In: 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW). pp. 3882–3891. IEEE, Montreal, BC, Canada (2021)
7. Cinar, Z.M., Nuhu, A.A., Zeeshan, Q., Korhan, O.: Digital twins for industry 4.0: a review. In: Calisir, F., Korhan, O. (eds.) GJCIE 2019. LNMIE, pp. 193–203. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-42416-9_18
8. Amigo, D., Pedroche, D.S., García, J., Molina, J.M.: Automatic individual tree detection from combination of aerial imagery, LiDAR and environment context. In: Sanjurjo González, H., Pastor López, I., García Bringas, P., Quintián, H., Corchado, E. (eds.) SOCO 2021. AISC, vol. 1401, pp. 294–303. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-87869-6_28
9. Amigo, D., Pedroche, D.S., García, J., Molina, J.M.: Automatic context learning based on 360 imageries triangulation and 3D LiDAR validation. In: 2021 24th International Conference on Information Fusion (FUSION), p. 8 (2021)

10. Ghamisi, P., et al.: Multisource and multitemporal data fusion in remote sensing: a comprehensive review of the state of the art. IEEE Geosci. Remote Sens. Mag. **7**, 6–39 (2019). https://doi.org/10.1109/MGRS.2018.2890023
11. Ribeiro, L.G.: 3D Reconstruction of Civil Infrastructures from UAV Lidar point clouds. 71
12. Siebert, S., Teizer, J.: Mobile 3D mapping for surveying earthwork projects using an unmanned aerial vehicle (UAV) system. Autom. Constr. **41**, 1–14 (2014). https://doi.org/10.1016/j.autcon.2014.01.004
13. Mentasti, S., Pedersini, F.: Controlling the flight of a drone and its camera for 3D reconstruction of large objects. Sensors **19**, 2333 (2019). https://doi.org/10.3390/s19102333
14. Luhmann, T., Chizhova, M., Gorkovchuk, D.: Fusion of UAV and terrestrial photogrammetry with laser scanning for 3D reconstruction of historic churches in Georgia. Drones. **4**, 53 (2020). https://doi.org/10.3390/drones4030053
15. Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J.: Frustum PointNets for 3D Object Detection from RGB-D Data. ArXiv171108488 Cs (2018)
16. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. ArXiv170602413 Cs (2017)
17. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3D ShapeNets: A Deep Representation for Volumetric Shapes. ArXiv14065670 Cs (2015)

A SO_2 Pollution Concentrations Prediction Approach Using Autoencoders

M. I. Rodríguez-García(✉), J. González-Enrique, J. J. Ruiz-Aguilar, and I. J. Turias

Higher Technical Engineering School, University of Cádiz, Avd. Ramón Puyol s/n,
11202 Algeciras, (Cádiz), Spain
{inma.rodriguezgarcia,javier.gonzalezenrique,juanjesus.ruiz,
ignacio.turias}@uca.es

Abstract. To achieve the prediction of $SO_2(t + 1)$ concentration values in the area of the Bay of Algeciras, Autoencoders (AE) and Sparse Autoencoders (SAE) have been applied to analyse air quality in this complex zone. A three-year hourly database of air pollutants, meteorological and vessel data were used to test different prediction scenarios. The data were divided into disjoint quartiles (Q1–Q4). AE models are better performed in the medium values (quartiles Q2 and Q3) and SAE models produce equivalent results in low and high values (Q1 and Q4). The results show that AE layers can be stacked to configure a more complex network with different levels of the sparsity of dimensions, together with a final supervised layer for the prediction of the index of the SO_2 level (quartiles Q1–Q4).

Keywords: Autoencoders · Air pollution forecasting · Sulphur dioxide · Unsupervised learning

1 Introduction

The last international MED ports conference celebrated on 28th April in The Algeciras Bay Port Authority, focused its attention on green ports. This new green development is the main motivation for this study which is carried out in the Bay of Algeciras (South of Spain), the most important port zone in Andalusia and the fourth in Europe. Algeciras port moves more than 1 million tons of goods every year since 2017 and is located in a peculiar zone. In this area, co-exist industries, roads, and the Gibraltar airport which, together with the port, contribute to a very complex air pollution scenario. This study is a continuation of other studies related to air quality in the Bay of Algeciras, together with the use of sensors and other exogenous information such as meteorological information [1–4]. The most relevant variables are obtained in [1] to estimate hourly concentrations of other pollutants in the Bay of Algeciras. This research can be useful to determine similar estimations of SO_2 pollutant in the same area of study. An in-depth analysis is presented in [2] comparing different filter ranking techniques which are applied to estimate air pollution in the Bay of Algeciras.

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 41–50, 2023.
https://doi.org/10.1007/978-3-031-18050-7_5

Using Machine Learning methods, in [3] authors propose an Air Quality Index (AQI) by means of a two-step forecasting approach to obtain eight hours ahead of future values.

Besides, in [4] SO_2 concentrations in two different monitoring stations are compared to get knowledge about the influence of the port of Algeciras.

In recent years, there are also several studies where Deep Learning (DL) techniques have been used in other locations and scenarios to analyse air quality [5–9].

The use of deep learning in the study [5], attempted to develop air pollution architectures to predict future results.

Spatiotemporal correlation is proposed in [6] using deep learning methods and the use of stacked autoencoder (SAE) obtained intrinsic features which could predict simultaneously the air quality of all stations in industrialised areas. The use of bidirectional long-term memory (Bi-LSTM) together with autoencoder layer can improve the prediction accuracy of pollutant concentrations [7]. In [8] a review about how deep learning can be applied to air quality forecast introduces all the architecture of deep networks (e.g., convolutional neural networks, recurrent neural networks, long short-term memory neural networks, and spatiotemporal deep networks). An interesting application of a autoencoder model is presented in [9] to forecast pollution.

There is no work where autoencoders have been used in this area of study. Thus, one of the objectives is to test the efficiency of autoencoders for use in air quality prediction and, particularly, in this complex scenario.

Deep Learning (DL) is a part of machine learning where there are a massive number of connections that allow other ways of learning the correspondences between inputs and outputs to be designed and which has received a great deal of interest in recent years among academics and industry [10, 11].

In this paper, we propose a deep learning-based method for SO_2 prediction. We use a stacked autoencoder (AE) to build a spatiotemporal prediction framework, which considers variable relations of the dataset in the modeling process. This model approach can predict the SO_2 values in Algeciras (Spain) and shows the accuracy of the process.

An autoencoder is a neural network that attempts to replicate its input at its output. Thus, the dimension of its input will be the same as the dimension of its output. When the number of neurons in the hidden layer is less than the size of the input, the autoencoder learns a compressed representation of the input. The scheme of a stacked autoencoder model is deeply explained in [12] and the authors used it to predict air quality. They compared the different prediction results with the Air Quality Index (AQI) in several locations, confirming that the predicted values of pollutants showed similar patterns.

This paper is organized as follows. The database is given in Sect. 2. The methodology is presented in Sect. 3. The results are conducted in Sect. 4. Finally, some conclusions are shown in Sect. 5.

2 Database

This study is located in The Bay of Algeciras. The database is formed from data kindly provided by the Andalusian Government, which has several sensors in this area (see

Fig. 1). Besides, a vessel database has been kindly provided by the Algeciras Bay Port Authority. The variables are meteorological, different air pollutants, and vessel data recorded hourly during the three-year period from 1st January 2017, to 31st December 2019 containing 26280 hourly records of 131 variables.

The first step was the preprocessing of the data as well as the separation into quartiles (Q1–Q4). The distribution of the quartiles from the cumulative probabilities that divided data in 0.25, 0.5, 0.75, resulting the intervals [0, 0.25), [0.25, 0.5), [0.5, 0.75), [0.75, 1].

Table 1 shows the used variables and their monitoring stations where they are measured. One of the most interesting variable could be the time series of vessels. Port Authority gives us a complete database of vessels that are located in the Bay of Algeciras. The Port of Algeciras records around 100.000 vessels per year. In this work, we have transformed this database into a time series of gross tons per hour (GT/h). This time series consists of considerable huge figures that had to be standardized along with the rest of the variables.

The SO_2 data from the Algeciras station (1) were predicted using all the meteorological variables and the rest of the air pollutants (see Table 1) (NO_x, CO, PM, etc.) in the rest of the stations (15) in the region of the Bay of Algeciras, located in the Strait of Gibraltar, which suffers very singular meteorological conditions and an area where numerous chemical industries are concentrated, the port of Algeciras with its important maritime traffic and the airport of Gibraltar as important sources of SO_2. The data are arranged in an autoregressive way to be able to predict the next $SO_2(t + 1)$ value based on the current values (at time t) of all variables.

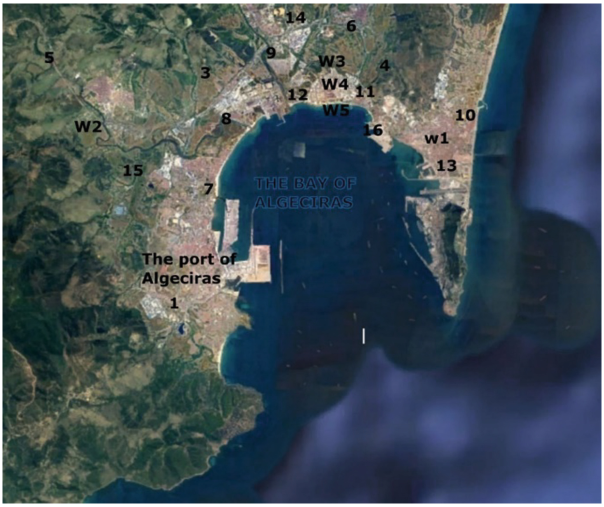


Fig. 1. Location of the monitoring stations (m.s.). [1–16 pollutant m.s.; W1–W5 weather m.s.]

Table 1. Variables of the study and their monitoring stations.

Variables	Units	Stations
Pollutants[1]	$\mu g/m^3$	1–16
Meteorological[2]	Various	W1–W5
Vessels	GT/h	Port Authority database

[1] Pollutants: SO_2, NO_2, NO_x, $PM_{2.5}$, PM_{10}, CO, O_3, Toluene, Benzene, Ethylbenzene.
[2] Meteorological variables: wind speed (km/h), wind direction (Degrees), solar radiation (w/m^2), relative humidity (%), atmospheric pressure (hPa), temperature (T), rainfall (l/m^2).

3 Methodology

Autoencoders (AEs) are neural networks whose goal is to replicate the system's input data to the output with as little distortion as possible. AEs play an important role in machine learning. They were first introduced in the 1980s by Hinton and the PDP group to address the problem of "unsupervised backpropagation" [13], using the input data as "supervision". AEs form one of the fundamental paradigms for unsupervised learning in which synaptic changes introduced by local events can be coordinated in a self-organising way to produce global learning and interesting behaviour [14]. In the hidden layer of AE, a mapping of input features takes place. The corresponding encoding of the data is obtained at the output of the hidden layer, distinguishing two possible situations: if the number of hidden neurons (nhiddens (NH)) is smaller than the dimension (D) of the input data $NH < D$, a compressed code of the data will result; while if $NH > D$, a sparse representation of the data is obtained [15, 16].

Autoencoders could be considered as simple multilayer networks trained in an unsupervised way, where the dimension of the input data matches with that of the output data. In other words, an autoencoder achieves a "mirror" representation from the input to the output of its network. One of the most important issues is that the autoencoder learns in its intermediate layer a representation of the input data on an of the input data in a different dimensional space.

This work consists of analysing the results of a Sparse Autoencoder (SAE) machine and comparing them with those obtained by AE machines. We will observe that the former are capable of extracting certain characteristics from the input data set, while the AE are not, although both can reproduce the input at the output of the network. For this purpose, we will train both networks (AE and SAE) independently until we reach a minimum value of the error function for the validation dataset.

In this work, a configuration based on 2 layers of autoencoders has been used, which are then stacked in series (stacked AE) together with a last supervised layer that allows predicting the quartile of the SO_2 level of pollution. The experimental process we have followed in this study consists first of a pre-processing phase of the data. For this purpose, missing values from the different time series of pollutant concentrations and meteorological variables have been imputed. The time series of ships in the Bay has been transformed to GT/h (Gross Tons per hour) because it was a recorded data of entries and exits of ships in the Bay. Subsequently, all the variables have been normalised

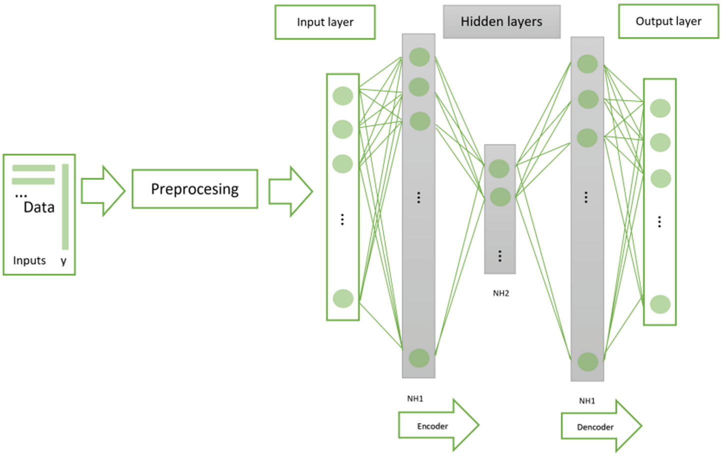


Fig. 2. Autoencoder/Sparse autoencoder scheme. Training stage.

from a statistical point of view. Once the information had been pre-processed, two autoencoders were trained, one of dimension NH1 and the other of dimension NH2. The two autoencoders were joined together in a stacked autoencoder as shown in Fig. 2, which illustrates the design or training phase. Once the autoencoders are trained (to reproduce the input at the output), they are used in the test phase as shown in Fig. 3, where an additional layer is included to learn the desired output, which in this case is the future value of the signal.

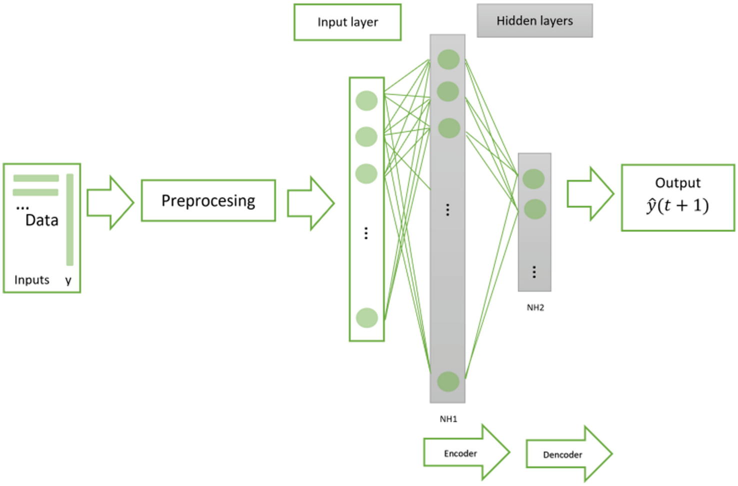


Fig. 3. Autoencoder/Sparse autoencoder scheme. Test stage.

Table 2. Multi-class confusion matrix (C(i,j)).

	Quartile/class	Real class			
Predicted class	1	C(1,1)	C(1,2)	C(1,3)	C(1,4)
	2	C(2,1)	C(2,2)	C(2,3)	C(2,4)
	3	C(3,1)	C(3,2)	C(3,3)	C(3,4)
	4	C(4,1)	C(4,2)	C(4,3)	C(4,4)
	Quartile/class	1	2	3	4

Table 3. Equivalent multi-class confusion matrix.

	Real class	
Predicted class	TP = C(i,i);	FP = sum(C(i,:)) − C(i,i);
	FN = sum(C(:,i)) − C(i,i);	TN = sum(sum(C)) − (TP + FP + FN);

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{3}$$

$$Specificity = \frac{TN}{TN + FP} \tag{4}$$

4 Results

An experimental procedure based on a grid search has been used in this work in order to calculate different measurements of the quality of the prediction approach. Confusion matrix was computed for each model (with different configuration of NH1 and NH2 parameters). The multi-class confusion matrix with dimension 4 × 4 (Table 2) was transformed into an equivalent confusion matrix 2 × 2 (Table 3) used to calculate the well-known classification measurements (i.e. sensitivity, specificity, accuracy, and precision, see Eqs. (1)–(4)). Also, the Euclidean distance to a perfect classifier ((1,1,1,1)-classifier with sensitivity, specificity, accuracy, and precision equal to 1) were computed (d_1). The procedure is repeated 20 times to assure the independence of the results, and to be able to test the existence of different groups of models using a multiple comparison test such as the Friedman test together with a post-hoc test such as Bonferroni. The mean values are shown in Table 4. The results collected in Table 4 comes from using the year 2019 as a test set and the years 2017–2018 as training set. A Multiple Linear Regression (MLR) has been computed as a benchmark method. The set of tested models has been compared using a multiple comparison test. The Bonferroni test allows us to select groups of models that are statistically different and choose the best model (i.e. the simplest within a group) in this case, for each quartile. The use of Bonferroni test is applied in many fields to discover groups of equivalent models [17].

On the other hand, hyperparameters of the Stacked AE were settled down through another grid search: L2 regulariser for the weights of the network (and not the biases), a Sparsity Regularization, and a Sparsity Proportionl. The values for the first autoencoder were (0.004, 4, 0.15), and for the second autoencoder were (0.002, 4,0.1).

The original vectors in the training data have 130 dimensions. After passing them through the first autoencoder, this was transformed to NH1 dimensions. After using the second autoencoder, this was transformed again to NH2 dimensions. After we have trained a final supervised layer to classify these NH2-dimensional vectors into the different 4 pattern classes (Q1–Q4) from lower SO_2 concentration values (Q1) to higher SO_2 concentration values (Q4).

After training the first autoencoder, we have trained the second autoencoder. The main difference is that we use the features that were generated from the first autoencoder as the training data in the second autoencoder. Also, we have decreased the size of the hidden representation to NH2, so that the autoencoder in the second autoencoder learns an even smaller representation of the input data.

After executing the planned experimental procedure, it has been found that in general the non-Sparse configurations produce better results in the prediction of the studied air pollution index than the Sparse configurations of the AE stacker.

In Table 4, the best configurations (best models) for each of the classes analyzed in terms of sensitivity, specificity, accuracy, and precision are marked in bold. Actually, the configurations marked in bold are results from a Bonferroni test, using the index d_1. The lowest distance values of d_1 are the best models. Friedman test is the non-parametric version of the ANOVA-test method for data not following a normal statistical distribution. Therefore, these configurations are significantly different models from statistical multiple comparisons using a post-hoc Bonferroni-test. In case there is a group of models that are not significantly different, i.e. statistically equivalent, we can select the simplest model within that group (Ockham's razor criterion). In Table 4, in the case of the Q1 quartile, there is a group with at least 3 equivalent models and although the model with the best indices is the 500 neuron model, the simplest model of 75 neurons has been selected as the best model.

Additionally, it is observed that the best indexes are obtained in classes 1 and 4 (the fourth quartile (Q4) is the highest level concentrations). It is worth mentioning that in the case of quartiles Q1–Q3 the best configurations are non-sparse autoencoders (AE). On the contrary, in the case of quartile Q4, the best model is a sparse autoencoder (SAE).

Table 4. Classification results for each quartile (Q1–Q4). NH1 = {50, 75, 200, 500} and NH2 = {5, 10, 50, 150}.

Quartile/class	MLR	Neurons		Sensitivity	Specificity	Accuracy	Precision	d_1
		NH1	NH2					
Q1	MLR	–		0.870	0.656	0.709	0.457	0.716
		50	5	0.889	0.883	0.886	0.868	0.238
		75	**50**	**0.897**	**0.884**	**0.890**	**0.869**	**0.231**
		200	5	0.887	0.888	0.888	0.875	0.231
		500	150	0.886	0.896	0.891	0.885	0.221
Q2	MLR	–		0.161	0.754	0.581	0.176	1.271
		50	5	0.498	0.900	0.799	0.622	0.667
		75	**5**	**0.534**	**0.904**	**0.816**	**0.630**	**0.630**
		200	5	0.495	0.894	0.798	0.597	0.686
		500	50	0.423	0.898	0.753	0.645	0.728
Q3	MLR	–		0.136	0.755	0.575	0.146	1.309
		50	10	0.543	0.897	0.825	0.575	0.657
		75	**5**	**0.577**	**0.896**	**0.836**	**0.564**	**0.638**
		200	5	0.544	0.889	0.824	0.534	0.684
		500	50	0.492	0.883	0.804	0.518	0.736
Q4	MLR	–		0.132	0.834	0.633	0.161	1.272
		50	10	0.711	0.929	0.908	0.516	0.575
		75	5	0.667	0.940	0.908	0.603	0.529
		200	**150**	**0.671**	**0.945**	**0.911**	**0.637**	**0.501**
		500	150	0.710	0.936	0.912	0.566	0.534

5 Conclusions

In this work, a prediction approach based on autoencoders (AE and SAE) has been presented. The model's results have been compared in terms of sensitivity, specificity, accuracy, and precision, and the distance to a perfect classifier (1,1,1,1) using Bonferroni and Friedman comparison tests. We applied the models to hourly air quality data (SO_2) from Algeciras station (Spain) in the function of other pollutants and meteorological variables recorded in the area of the Bay of Algeciras. SO_2 future concentrations were predicted using different values of hyperparameters and different architectures of stacked autoencoders. The proposed models effectively predicted SO_2 concentrations, with non-sparse AE models showing slightly better performance in the lower values (quartiles Q1–Q3) and SAE models producing equivalent results in quartile Q4 (high values).

With our forecasting approach, it is possible to give reliable SO_2 prediction information for the Q1 and Q4 classes. Future works will address in using LSTM networks combined with AS/SAE to improve forecasting results.

Acknowledgements. This work is part of the research project RTI2018-098160-B-I00 supported by ‘MICINN’ Programa Estatal de I+D+i Orientada a ‘Los Retos de la Sociedad’. Data used in this work have been kindly provided by the Algeciras Bay Port Authority and the Andalusian Regional Government.

References

1. González-Enrique, J., Turias, I.J., Ruiz-Aguilar, J.J., Moscoso-López, J.A., Franco, L.: Spatial and meteorological relevance in NO_2 estimations: a case study in the Bay of Algeciras (Spain). Stoch. Environ. Res. Risk Assess. **33**, 801–815 (2019)
2. González-Enrique, J., Ruiz-Aguilar, J.J., Moscoso-López, J.A., Urda, D., Turias, I.J.: A comparison of ranking filter methods applied to the estimation of NO_2 concentrations in the Bay of Algeciras (Spain). Stoch. Env. Res. Risk Assess. **35**, 1999–2019 (2021)
3. Moscoso-López, J.A., Urda, D., González-Enrique, J., Ruiz-Aguilar, J.J., Turias, I.J.: Hourly air quality index (AQI) forecasting using machine learning methods. In: Herrero, Á., Cambra, C., Urda, D., Sedano, J., Quintián, H., Corchado, E. (eds.) SOCO 2020. AISC, vol. 1268, pp. 123–132. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-57802-2_12
4. Rodríguez-García, M.I., González-Enrique, J., Moscoso-López, J.A., Ruiz-Aguilar, J.J., Rodríguez-López, J., Turias, I.J.: Comparison of maritime transport influence of SO_2 levels in Algeciras and Alcornocales Park (Spain). In: XIV Conference on Transport Engineering, CIT2021, vol. 58, pp. 2352–1465 (2021)
5. Akin, Y., Cansu, Z., Oktay, H.: Air pollution modelling with deep learning: a review. Int. J. Environ. Pollut. Environ. Model. **1**(3), 58–62 (2018)
6. Li, X., Peng, L., Hu, Y., Shao, J., Chi, T.: Deep learning architecture for air quality predictions. Environ. Sci. Pollut. Res. **23**, 22048–22417 (2016)
7. Bo Zhang, B., Zhang, H., Zhao, G., Lian, J.: Constructing a PM2.5 concentration prediction model by combining autoencoder with Bi-LSTM neural networks. Environ. Model. Softw. **124** (2020)
8. Liao, Q., Zhu, M., Wu, L., Pan, X., Tang, X., Wang, Z.: Deep learning for air quality forecasts: a review. Curr. Pollut. Rep. **6**, 399–499 (2020)
9. Mengara, A.G., Park, E., Jang, J., Yoo, Y.: Attention-based distributed deep learning model for air quality forecasting. Sustainability **14**(6), 3269 (2022)
10. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
11. Baldorj, B., Tsagaan, M., Sereeter, L., Bulkhbai, A.: Embedded generative air pollution model with variational autoencoder and environmental factor effect in Ulaanbaatar city. Atmosphere **13**(1), 71 (2022)
12. Xayasouk, T., Lee, H.: Air pollution prediction system using deep learning. WIT Trans. Ecol. Environ. **230** (2018)
13. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**, 533–536 (1986)
14. Baldi, P., Lu, Z.: Complex-valued autoencoders. Neural Netw. **33**, 136–147 (2012)
15. Makhzani, A., Frey, B.: k-Sparse autoencoders. In: 2nd International Conference on Learning Representations, ICLR 2014 – Conference Track Proceedings (2014)

16. Tino, P., Benuskova, L., Sperduti, A.: Artificial Neural Network Models, pp. 455–471. Springer Handbook of Computational Intelligence (2015)
17. Bland, J.M., Altman, D.G.: Multiple significance tests: the Bonferroni method. BMJ **310**(6973), 170 (1995)

CPU Computation Influence on Energy Consumption Forecasting Activities of a Building

Daniel Ramos[1,2], Pedro Faria[1,2](✉), Luis Gomes[1,2], and Zita Vale[1,2]

[1] GECAD - Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development, Polytechnic of Porto, Porto, Portugal
{dados,pnf,log,zav}@isep.ipp.pt
[2] LASI - Intelligent Systems Associate Laboratory, Porto, Portugal

Abstract. Scheduling forecasting activities and improving the forecasting accuracy is important to deliver energy efficiency to the customers. However, it is also important to reduce the computational effort dedicated to these forecasting activities to ensure more effective environment sustainability. This paper proposes two forecasting algorithms known as artificial neural networks and k-nearest neighbors to anticipate energy patterns of a building monitoring data from five-to-five minutes. Using a case study with an annual historic and one week test, different scenarios are defined to test the forecasting activities with both higher and lower computational effort. It is achieved to ensure energy predictions with above reasonable accuracies evaluations while decreasing the computational effort, and the respective energy consumption, dedicated to forecasting activities.

Keywords: Computational effort · Electricity consumption · Environment sustainability · Forecasting algorithms

1 Introduction

Forecasting activities are essential in the energy sector to predict consumption patterns with uncertainty behaviors [1]. These activities should be integrated in demand response programs to convince consumers to reduce the power consumption during peaks events [2]. Therefore, improving the accuracy of forecasting models is essential to avoid demand response issues. However, this improvement has the disadvantage of requiring more energy spent to compute these forecasting activities. With this in mind, it is important to define a balance between the forecasting accuracy improvement and the energy effort required to compute the forecasting activities. The forecasting accuracy of some classic models have been compared to a hybrid model that uses feature selection techniques and a dataset decrease strategy. These two abilities decrease the forecasting speed of the hybrid model, therefore displaying a better performance than the other classic models [3]. An optimization model determines optimal prices according to the stakeholders' objectives using a decision-making criterion with economic, social

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 51–61, 2023.
https://doi.org/10.1007/978-3-031-18050-7_6

and environmental levels [4]. Smart grid is a concept related with the advancements in the electrical grid accommodating automated and intelligent decisions for optimized operation. Smart grids technology may be integrated in demand response programs to reduce consumption power during peak events. Forecasting activities are also useful on this demand response issue, for example artificial neural network are very effective on anticipating uncertainty prices variations. This supports decision makers to plan better which schedules should consumers be convinced to reduce their power consumption to avoid peak events [5]. Computing devices may be integrated on the green computing area relying on the idea that these should decrease the environment impact and contribute for the formation of smart environments [6]. Green computing is crucial due to IT technologies causing a lot of environment impact while spending unnecessary amounts of energy power. Therefore, solutions to minimize the energy costs and to improve the energy efficiency are required in the green computing area [7]. The network devices play a crucial role in the environment protection, thus the cloud computing field is a promising way to achieve the green computing goals [8]. Smart grids technologies integration with green computing is very important to assure the energy optimization guaranteeing more effective and efficient management operations. This integration has also the advantage of decreasing the greenhouse emissions, thus ensuring environment protection [9]. One example is the estimation of future energy events in residential households equipped with green energy sources namely solar and electricity energy [10]. The energy management of datacenters equipped with green computing devices is another example with two primary objectives: to improve the energy efficiency and to decrease the power consumption during demand response peak events [11]. To accomplish these goals, it is crucial to maximize the usage of green energy sources and to decrease the energy costs [12]. The energy efficiency of buildings may be improved with the support of suitable algorithms used for forecasting activities. The authors of this paper used on previous publications two forecasting algorithms known as artificial neural networks and k-nearest neighbors to reach effective energy power predictions [13–15]. However, these publications do not incorporate the benefit of green computing on decreasing the computational effort during forecasting activities. Actually, this is a more recent topic that the authors of this paper researched recently, exploring why decreasing the computational effort spent in forecasting tasks is important in the green computing domain. The authors of this paper also added planned scenarios intended on decreasing the size of the historic and target data for forecasting tasks [16]. Similarly to what has been done in recent publications previously indicated, this paper proposes two forecasting algorithms known as artificial neural networks and k-nearest neighbors. This paper also incorporates one strategy planned recently on the green computing research to decrease the computational effort in forecasting activities while predicting a week on a single run with five minutes periods or hour schedules.

After presenting all the aspects of this introduction, a methodology is proposed in Sect. 2, followed by the case study and results presented in Sect. 3, finally the main conclusions are explained in Sect. 4.

2 Methodology

This section describes the different phases involved in the proposed methodology including the data transformations, CPU training and forecasting and a performance evaluation that studies the forecasting error and the total energy cost spent. The methodology is illustrated in Fig. 1.

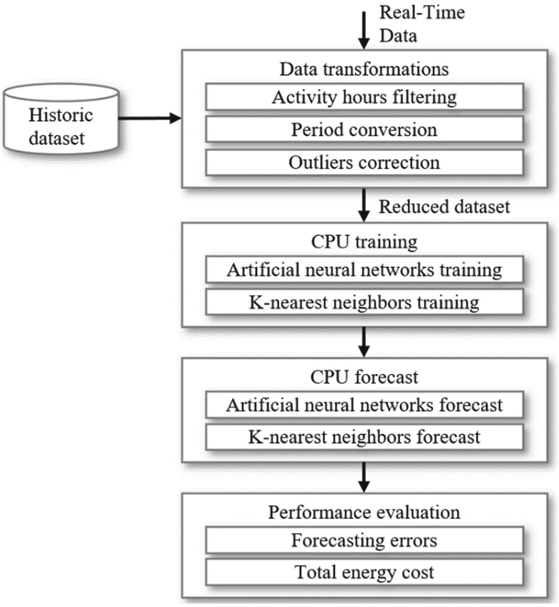


Fig. 1. Proposed methodology for data simplification and forecasting activities.

This historic dataset contains consumption and sensors data in different five minutes periods. This data is monitored from installed IoT devices in an electrical building. The selected sensors are CO_2 and light presence presenting the latter logic values between 0 and 1 evidencing respectively whether there is no light activity in the building or if there is instead light activity in at least a part of the building. Data transformations are applied to the historic dataset to reduce the computational effort during later forecasting activities. These transformations start by filtering the consumptions during activity hours between 11 a.m. and 6 p.m from Monday to Friday. The five minutes consumptions and sensors data may be reformulated to hour schedules during a period conversion thus decreasing the computational effort during later forecasting activities even further. The transaction of five minutes to hour schedules involves average consumption and sensor calculations of twelve five minutes observations in each hour.

The outliers correction reformulates the consumption and sensors behaviors corresponding to detected outliers for the new context. Outliers are detected for the new context due to inconsistence values compared to other values in other periods. These mistakes may occur for example due to wrong devices reading or inconsistence scale conversions to hour schedules. The outliers detection use the average and standard deviation calculation.

An outlier algorithm detects mistaken consumption values through a careful consumption analysis. This assumes that outlier consumptions correspond to values higher or equal to the average consumption plus an error factor times the standard deviation or to values lower or equal to the average consumption minus an error factor times the standard deviation. The detection of each outlier consumption is followed by a correction featuring an average between the previous and following consumption.

The rules of the filtering and simplification of data procedure apply for the historic dataset and for the real-time data as well. The obtained reduced version of data is sent to a training procedure that trains an annual historic of consumptions and sensors with the CPU processing unit effort and the artificial neural network and k-nearest neighbors algorithms. A following forecasting procedure uses the CPU processing unit to forecast all the consumptions in a single run for all periods of an entire week either in five minutes or hour schedules with the support of artificial neural network and k-nearest neighbors algorithms. The training and forecasting activities may either be performed sequentially or train the data for each forecasting algorithm and save it in the disk storage to load it later in RAM during forecasting activities. The artificial neural networks model is composed by an input layer with twelve neurons, two sequential hidden layers with thirty-two neurons, and an output layer with one neuron. The neurons from each layer are connected to the neurons of the following layer through weights. Moreover, the input layer is responsible to send to the hidden layers ten sequential consumption inputs and the values of the selected sensors preceding the forecasted period and separated in five minutes or hour schedules, processing the information with the rectified linear unit activation function. The information is also processed from the hidden layers to the output layer which takes only a neuron to calculate the forecast consumption. This algorithm uses a small learning rate assigned to 0.001 for a rigorous search intended to minimize the forecasting error. The artificial neural networks model is trained with two hundred epochs and added with an early stopping procedure that automatically stops the training model if no improvements are found. The procedure use is justified by the need to avoid the overfitting of data. The k-nearest neighbors model finds similarities in the data with the support of five neighbors and a Euclidean distance procedure. The artificial neural networks and k-nearest neighbors configuration selection features the cases resulting in lower forecasting errors as studied previously by the authors of this paper [14]. A performance evaluation procedure calculates the forecasting errors for all periods of the forecasted week and for both forecasting algorithms. The performance evaluation also measures the total energy cost spent previously with the CPU processing unit in the arduino device. The forecasting metrics used to calculate the error is Symmetric Mean Absolute Percentage Error (SMAPE). SMAPE calculates the sum of the consumption error calculation on all periods either in five minutes or hour schedules.

3 Case Study and Results

This section presents case study and results, respectively in Sub-Sects. 3.1 and 3.2.

3.1 Case Study

This case study intends to reduce the computational effort for forecasting activities and to obtain accurate predictions. The electrical building controls and monitors consumption data in five minutes periods from 22nd May 2017 to 17th November 2019. Data from 2020 and 2021 is excluded due to the energy behavior being less trustable during the pandemic. The activity hours of this annual historic are explored due to the productive consumption behaviors during this schedule. Therefore, periods from 11 AM to 6 PM during weekdays from Monday to Friday are filtered in this case study. Moreover, the adoption of five minutes or hour schedules has an impact on the computational effort and forecasting accuracy. Therefore, it is wise to consider both schedule alternatives during forecasting activities. Moreover, forecasting activities require the split of consumption data from 22nd May 2017 to 15th November 2019 in train and test datasets. Train data takes all consumptions from 22nd May 2017 to 8th November 2019 while the test data takes all consumptions from 11th to 15th November 2019. The training data presents consumption behavior ranges between 0 and 5000 kWh from 22nd May 2017 to 8th November 2019. The test data presents consumption behavior ranges between 500 and 2500 kWh. While it is clear in the training dataset that the consumption behavior differs from week to week, the test dataset makes clear that the usual consumption behavior presents consumptions between 500 and 1500 kWh as evidenced on Monday to Thursday patterns. Friday shows a lot more productivity in the week from 11st to 15th November 2019 presenting consumption ranges above 2000 kWh.

Different scenarios are declared with different parameterizations to perform forecasting activities on the activity hours of a test dataset from 11st to 15th November 2019 with the support of a historic with consumption data presenting activity behaviors from 22nd May 2017 to 8th November 2019. All the forecasted consumptions are calculated for the whole week in a single turn with the support of the indicated historic. All scenarios use the CPU processing unit to perform forecasting activities. The period of the training and test data may be formulated in five minutes or hour schedules. It is expected that hour schedules require less computational effort to process, however present less information than using five minutes schedules. The training may be categorized as included or previous meaning respectively that the consumptions are predicted right after the training of the whole historic, or that the historic training is saved in the disk storage and loaded later in RAM for forecasting tasks. Four scenarios with the period and training parameterizations are declared in Table 1.

Table 1. Forecasting scenarios calculated in a single run for whole week and targeted for activity hours while using the CPU processing unit.

Scenario	Period	Training
A	5 min	Included
B	1 h	Included
C	5 min	Previous
D	1 h	Previous

The scenarios presented in Table 1 evidence a total of four alternatives labeled from A to D. The scenarios A and C involving five minutes schedules use a total of 61920 records for train data and a total of 480 records for test data. The scenarios B and D involving hour schedules use a total of 5160 records for train data and a total of 40 records for test data.

3.2 Results

The results study the forecasting accuracies and the forecasting effort on the different scenarios previously mentioned in the case study. The artificial neural networks and k-nearest neighbors feature the forecasting algorithms indicated to measure the forecasting error, the dedicated time for training and cleaning operations and the total energy spent with the CPU processing unit. The forecasting errors use the SMAPE metrics to measure the forecasting deviation to the real consumptions on all activity hour on the week from 11 to 15 November 2019 for each one of the four scenarios as evidenced in Fig. 2. The train time of both algorithms and this added with the cleaning operation is analyzed for the different scenarios in Fig. 3. The total energy spent in the different scenarios with the CPU processing unit is studied as well in Fig. 4. Scenarios A and C result in lower SMAPE errors for both k-nearest neighbors and artificial neural networks algorithms, respectively of values corresponding to 6.37% and ranges between 6.46 and 8.42%. One the other hand scenarios B and D feature higher SMAPE errors for both k-nearest neighbors and artificial neural networks algorithms, respectively of values corresponding to 17.59% and ranges between 12.11 and 12.52%. It is noted that the SMAPE errors of scenarios A and B are equal respectively to the SMAPE errors of scenarios C and D for the k-nearest neighbors algorithm.

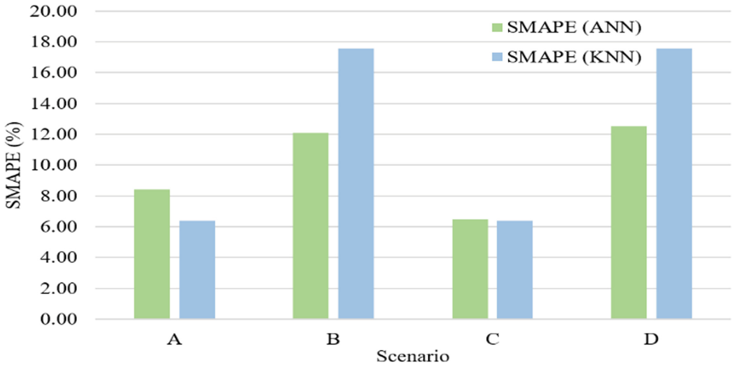


Fig. 2. SMAPE errors with artificial neural networks and k-nearest neighbors forecasting algorithms and a CPU processing unit.

This is understandable as the forecasting activities process the same training and target data for scenarios A and C featuring five minutes periods and for scenarios B and D featuring hour schedules.

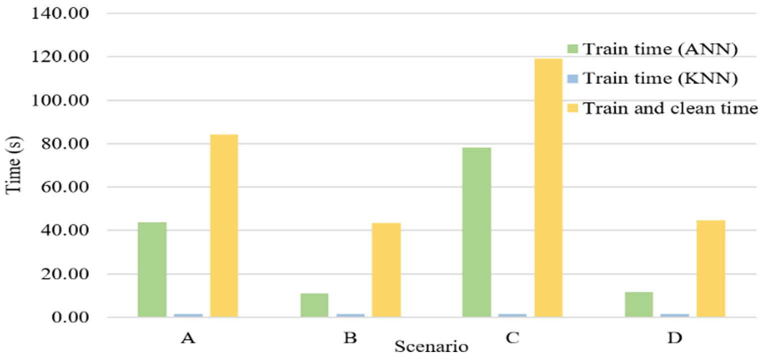


Fig. 3. Train time for artificial neural networks and k-nearest forecasting algorithms and the clean time added to the train time using a CPU processing unit.

The train time of artificial neural networks and k-nearest neighbors algorithms and this added with the cleaning operations time evidences non linear values for the different scenarios. The training of data with periods of five minutes requires much more time with artificial neural networks algorithm as featured in scenarios A and C. The required training time using the artificial neural networks algorithm requires a total of 43.98 and 78.06 s respectively for scenarios A and C. On the other hand, the training of data with hour schedules requires less time with artificial neural networks algorithm as featured in scenarios B and D. The required training time using the artificial neural networks algorithm requires a total of 11.05 and 11.80 s respectively for scenarios B and D. It is noted that saving the training in the storage disk and loading it later in RAM requires more training time using the artificial neural networks algorithm when comparing the higher value of scenario C with lower value of scenario A.

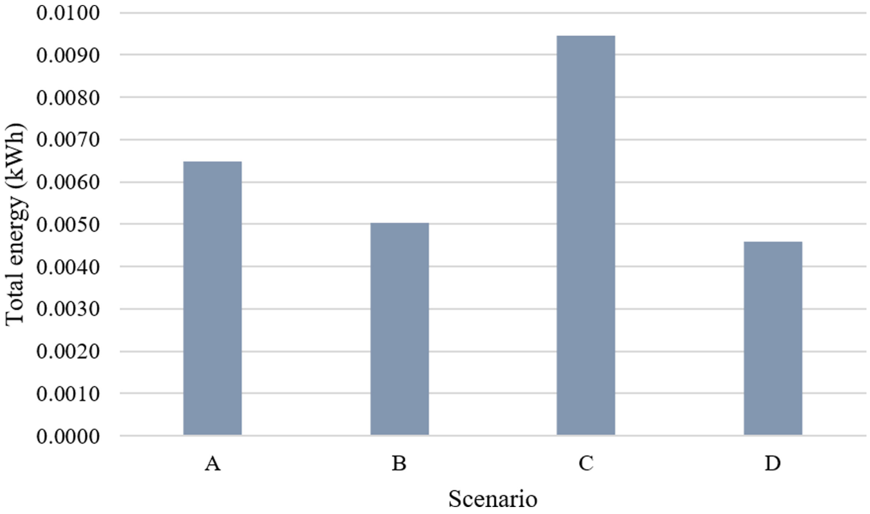


Fig. 4. Total energy spent with CPU processing unit effort.

The training time dedicated with k-nearest neighbor algorithm features a very small value for all four scenarios. Moreover, the train and clean time shows a high correlation with the artificial neural networks training time. Training the activity hours of the annual historic, performing cleaning operations and forecasting energy consumption values for all the activity hours on an entire week and in a single run has shown to do not take more than 120 and 45 s respectively using five minutes periods and hour schedules.

The total energy required for training and forecasting tasks with the CPU processing unit effort is different for the four scenarios. Using five minutes requires much more energy as evidenced by scenarios A and C with respectively 0.0065 and 0.0094 kWh. Moreover, it noted that for five minutes periods saving the training in the disk storage and loading it later in RAM as evidenced in scenario C requires more energy as shown while comparing to scenario A. Using hour schedules requires much less energy as evidenced by scenarios B and D with respectively 0.0050 and 0.0046 kWh. Moreover, it is noted that for hour schedules saving the training in the disk storage and loading it later in RAM as evidenced in scenario D requires less energy as shown while comparing to scenario B. The consumption forecasts of four scenarios are compared to the real consumptions for the activity hours of a week from 11 to 15 November 2019 in Fig. 5. The forecasts are focused on different small periods including five minutes frames as evidenced by scenarios A and C and hour schedules as evidenced by scenarios B and D. The forecasts are scheduled with artificial neural networks and k-nearest neighbors algorithms. Training the activity hours of the annual historic, performing cleaning operations and forecasting energy consumption values for all the activity hours of an entire week in a single run has shown to consume no more than 0.0094 and 0.0050 respectively using five minutes periods and hour schedules. The forecasting consumptions are very accurate for scenarios A and C featuring periods of five minutes. Scenarios B and D present above reasonable consumption forecasts with accuracies lower than the other scenarios due to hour schedules featuring less consumption patterns. Despite this, lower training and cleaning time and the less costly energy spent in scenarios B and D are advantages to use these two scenarios instead of scenarios A and C. It is noted that consumption patterns present usual behaviors between 500 and 1500 kWh for five minutes periods in scenarios A and C as evidenced during the activity hours from 11 to 14 November 2019 representing the Monday to Thursday usual behavior. In these scenarios, Friday shows a lot more productivity on 15 November 2019 corresponding to a Friday, reaching consumption behaviors between 500 and 2500 kWh. The hour schedules featured in scenarios B and D present consumption usual behaviors between 600 and 1200 kWh as evidenced from Monday to Thursday patterns, while Friday presents more productivity behaviors between 800 and 1800 kWh.

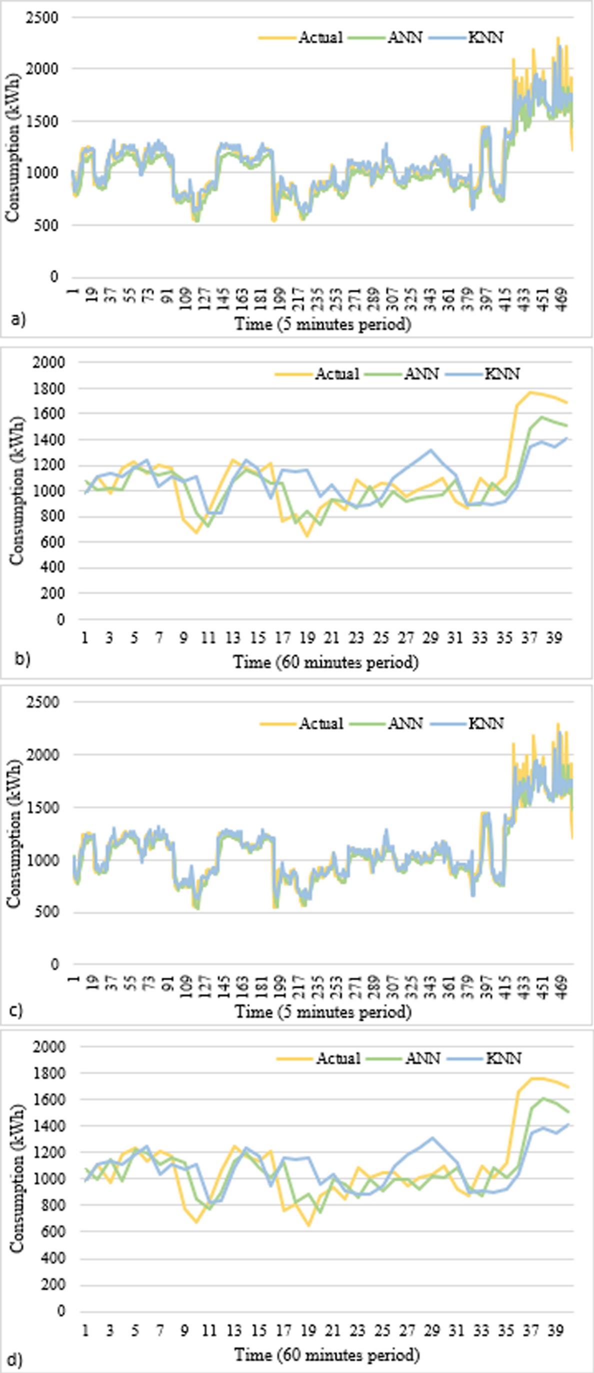


Fig. 5. Real and forecast consumption from 11 to 15 November 2019 with activity behaviors for scenarios A, B, C, and D.

4 Conclusions

This paper analyzes the forecasting accuracy and the computational effort for different scenarios using consumption data presenting activity behaviors and the artificial neural networks and k-nearest neighbors forecasting algorithms. Using data from five-to-five minutes results in much more accurate forecasts than using data from hour schedules as shown by the SMAPE forecasting errors and by the forecasts interval comparations to the real consumptions. Despite the forecasting accuracy improvements for five minutes data rather than hour schedules, the time dedicated for algorithms training and cleaning operations and the CPU effort show on the other hand that using five minutes requires much more time to perform the forecasting activities and it additionally requires much more CPU effort. Training the activity hours of an annual historic, performing cleaning operations and forecasting energy consumption values for all the activity hours of an entire week in a single run has shown to take low time and consume low CPU. With all this in mind, using the hour schedules require less computational effort and result in accurate forecasts despite not being as good as five minutes data. The obtained results will support decision makers on deciding the time anticipation to run the forecast according to the desired and needed accuracy, the available time, and the reasonable energy costs for such tasks.

Acknowledgements. This article is a result of the project RETINA (NORTE-01–0145-FEDER-000062), by (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, through the European Regional Development Fund (ERDF). Pedro Faria is supported by FCT with grant CEECIND/01423/2021. The authors acknowledge the work facilities and equipment provided by GECAD research center (UIDB/00760/2020) to the project team.

References

1. Lee, K., Vale, Z.: Applications of Modern Heuristic Optimization Methods in Power and Energy Systems. IEEE Press Series on Power and Energy Systems (2020)
2. Faria, P., Vale, Z.: Distributed energy resource scheduling with focus on demand response complex contracts. Journal of Modern Power Systems and Clean Energy **9**(5), 1172–1182 (2021)
3. Feijoo, F., Silva, W., Das, T.: A computationally efficient electricity price forecasting model for real time energy markets. Energy Convers. Manage. **113**, 27–35 (2016)
4. Dehghan, H., Amin-Naseri, M.: A simulation-based optimization model to determine optimal electricity prices under various scenarios considering stakeholders' objectives. Energy **238** (2022)
5. Collotta, M., Pau, G.: An innovative approach for forecasting of energy requirements to improve a smart home management system based on BLE. IEEE Trans. Green Commu. Netw. **1**(1), 112–120 (2017)
6. Pedrycz, W.: Welcome to the exciting world of green computing and smart environments. J Smart Environ Green Comput **1**, 1–2 (2021)
7. Patel, Y., Mehrotra, N., Soner, S.: Green cloud computing: a review on green IT areas for cloud computing environment. In: International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), pp. 327–332 (2015)

8. Radu, L.-D.: Green cloud computing: a literature survey. Symmetry **9**(12), 295 (2017)
9. Byun, J., Hong, I., Kang, B., Park, S.: A smart energy distribution and management system for renewable energy distribution and context-aware services based on user patterns and load forecasting. IEEE Trans. Consum. Electron. **57**(2), 436–444 (2011)
10. Liu, Z., Zhang, C., Dong, M., Gu, B., Ji, Y., Tanaka, Y.: Markov-decision-process-assisted consumer scheduling in a networked smart grid. IEEE Access **5**, 2448–2458 (2017)
11. Basmadjian, R.: Flexibility-based energy and demand management in data centers: a case study for cloud computing. Energies **12**(17), 3301 (2019)
12. Kiani, A., Ansari, N.: Toward low-cost workload distribution for integrated green data centers. IEEE Commun. Lett. **19**(1), 26–29 (2015)
13. Ramos, D., Teixeira, B., Faria, et al.: Using diverse sensors in load forecasting in an office building to support energy management. Energy Reports **6**(8), 182–187 (2020)
14. Ramos, D., Khorram, M., Faria, P., Vale, Z.: Load forecasting in an office building with different data structure and learning parameters. Forecasting **3**(1), 242–255 (2021)
15. Vale, Z., Faria, P., Abrishambaf, O., Gomes, L., Pinto, T.: MARTINE—A platform for real-time energy management in smart grids. Energies **14**(7), 1820 (2021)
16. Vale, Z., Gomes, L., Ramos, D., Faria, P.: Green computing: a realistic evaluation of energy consumption for building load forecasting computation. J Smart Environ Green Comput **2**, 34–45 (2022)

Python-Based Ecosystem for Agent Communities Simulation

Bruno Ribeiro(✉), Helder Pereira, Luis Gomes, and Zita Vale

GECAD—Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development, Polytechnic of Porto, 4200-072 Porto, Portugal
{brgri,hjvlp,lfg,zav}@isep.ipp.pt

Abstract. This paper presents an innovative multi-agent ecosystem framework designed to simulate various energy communities and smart grids while providing an easy and practical solution to manage and control each simulation. This framework allows the coexistence of various multi-agent systems and provides tools to enable the management of the ecosystem and its agents. The framework also provides a web application programming interface that allows the management to use third-party's software. The proposed framework was based on the Smart Python Agent Development Environment (SPADE) framework. Finally, this paper presents a case study that simulates an energy community with 50 members. The case study evaluates the community's energy bill by comparing a scenario without battery energy storage systems with a scenario where storage systems are available for some members of the community. The case study uses real storage units that are integrated into the proposed system and used for simulation.

Keywords: Multi-agent systems · Energy community · Simulation · Energy storage systems

1 Introduction

Smart grids are increasingly becoming more inevitable since these modern systems improve the power grid in such a way that this becomes more resilient, efficient, and cost-effective while promoting the use of renewable energy and distributed generation [1]. In addition, the smart grids allow the end-user to be more participative in the energy market as a result of the possibility that the end-users could produce and sell energy directly in the market, and among their peers [2]. This not only reduces market prices and tariffs but also promotes the use of renewable energy sources on the end-users-side, decreasing the CO2 emissions in the smart grid [3].

In the new context of smart grids, end-user can implement management models and actively participate in the management of the grid, contributing to the stability of the grid [4]. However, to enable this participation, end-users must be willing to share information about their energy profiles [5].

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 62–71, 2023.
https://doi.org/10.1007/978-3-031-18050-7_7

Multi-Agent Systems (MAS) can overcome many issues related to complex environments, distributed computing, distributed knowledge networks, parallel operations, asynchronous communications, and autonomy [6]. That is why this approach can be found in the literature to solve many complex problems [7–9].

More specifically, MAS are being used and explored in the context of smart grids [20]. This is because MAS meets the requirements for the simulation, development, management, and operation of smart grids. The use of MAS in smart grids improves reliability [10], responsiveness [7], fault tolerance [11], and stability of the smart grid [8]. The use of MAS enables the representation of multiple actors as well as their interactions, allowing the different actors to collaborate or compete.

The proposed Python-based Ecosystem for Agent Communities (PEAK) framework targets smart grids management and simulation with the ability to integrate simulated, emulated, and real energy loads and resources to enable the test and validation of new management and operation models in the context of smart grids, microgrids, and energy communities. This novel framework can integrate multiple MAS and energy communities in one single ecosystem while being able to manage them all at the same time. This unique way of managing such systems opens a range of possibilities within the multi-agent system paradigm applied to smart grids. The proposed framework is backed up by a case study, which was used a community with 50 consumers, of which 15 of them are prosumers, having photovoltaic generation. The community generation and consumption, for one day, is simulated considering two scenarios: where no battery energy storage systems (BESS) are installed; and where BESS are installed. This case study allows the evaluation of the influence of BESS in the energy community by measuring the financial balance of the community. Furthermore, the case study shows the potential and the capabilities that the proposed PEAK has towards the simulation of energy communities.

This paper is organized in the following manner: the following section has a concise description of related works; Sect. 3 describes the architecture of the proposed framework and its functionalities like the MAS management, the agent modularity, and the simulation environment tool; Sect. 4 describes the case study and discusses the results of the simulations that were made using the PEAK framework; and finally, in Sect. 5, it is presented the main conclusions of this paper.

2 Related Works

Agent-oriented programming (AOP) is the newest computer programming paradigm for large and distributed systems [12]. With higher-level abstraction, it has autonomic and proactive properties. In a dynamic environment, an agent is a computer program (process) capable of flexible and autonomous activity. Due to its intrinsic capacity to dramatically enhance operational efficiency in a distributive setting, the MAS has gotten a lot of attention [13]. MAS can enhance modularity, flexibility, adaptability, scalability, reconfigurability, and responsiveness.

In [14], authors purpose a MAS where the agents integrate optimization algorithms for players in an energy community to better manage their energy, reflecting the benefits in the community. In [15], the authors describe a MAS solution integrated with deep reinforcement learning to address the problem of energy sharing between zero energy buildings inside a zero-energy community. In [16], is proposed a MAS for citizen energy communities where members can trade energy in electricity markets, and manage their energy. Neither of these papers addresses an ecosystem of MAS that can module various energy communities in one single system.

To solve communication delay problems in networked MAS, in [17], is proposed a solution based on algebraic graphs and matrix theories. In [18], is shown a survey about the sample-data cooperative control of MAS and the research made since 2011. In [19], it is proposed a MAS solution to solve the constraints and communication time-delays related to direct graphs. Despite these papers trying to solve communication issues, there is still a lack of MAS frameworks that can provide reliable and fast communication mechanisms based on instant messaging technologies.

To build and deploy a MAS, there are some open-source tools available for example Java Agent DEvelopment Framework (JADE) [21], Smart Python Agent Development Environment [22], and Python Agent DEvelopment framework [23]. Nonetheless, these frameworks lack functionalities like managing tools and integration with data sources and real devices. JADE has another limitation that is related to Java language, it lacks artificial intelligence third-party software as powerful as Python library.

The current paper aims to describe a MAS framework architecture capable of creating and managing several MAS in a unique solution, or in other words, an ecosystem of MAS, while having mechanisms for fast and reliable communication to promote the integration of real loads and resources.

3 Proposed Solution

The proposed framework, PEAK, describes a new approach when it comes to handling smart grid simulations as it allows the coexistence of multiple MAS (ecosystem) at the same time, without interfering negatively with each other, while permitting the configuration of different types of simulations and agents. In addition, the PEAK framework allows the operation and monitorization of each MAS using a centralized web-based graphical interface.

The development of PEAK is based on SPADE, a Python framework focused on the development of peer-to-peer communication MAS. SPADE is XMPP-based, which means that it uses some Instant Messaging functionalities inherited from XMPP, as well as its architecture. To overcome the current limitation of SPADE, the PEAK framework was developed to enable concurrent agent executions, the existence of group communication, the existence of a directory facilitator, among other updates that would be described below.

In Fig. 1, it is possible to see the overall architecture of the PEAK framework. For PEAK execution, it is needed to configure at least one XMPP server for agents to be able to register, login, and communicate with the system and among other agents. One advantage that the XMPP gives to the framework is the ability to use any XMPP server, either public or private. In the XMPP context, the agents will act as regular users, making possible human-agent interactions.

As shown in Fig. 1, the PEAK's architecture is divided in two main modules: PEAK MAS Platform, which is responsible for everything related to the creation, implementation, synchronization, and communication among agents, and PEAK Management, which is the module responsible for the administration, and monitorization of the MAS ecosystem created by the framework.

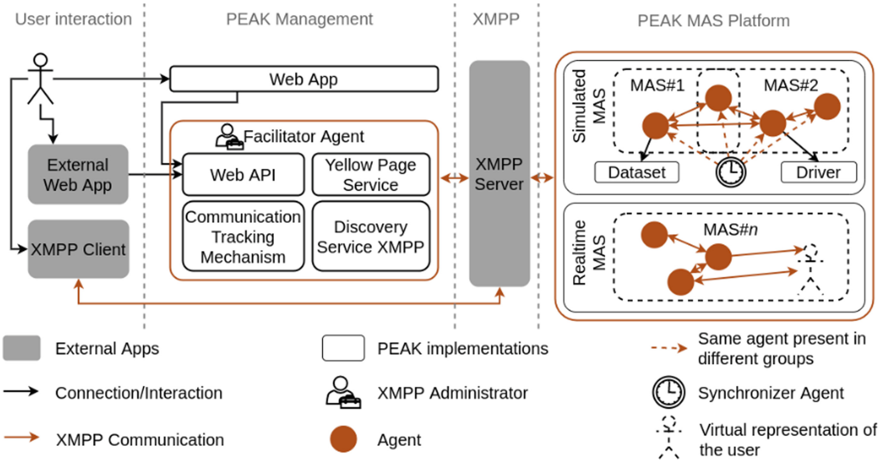


Fig. 1. Overall architecture of the PEAK framework.

3.1 PEAK Multi-agent System Platform

PEAK MAS Platform was designed to handle three main functionalities: the integration of datasets and drivers with the agents, the simulation of environments, and allowing the coexistence of multiple MAS in the same environment. In addition to that, this module allows the creation of the agents as different computer processes.

Normally, a MAS uses data to simulate an environment. The data can either be simulated based on approximations to real values or be real data extracted from physical devices. Because of it, the PEAK MAS Platform allows the integration of datasets from files and databases with the agent as well as integration with real devices through several communication protocols. With this PEAK module, it is possible to create four types of agents based on these types of data sources:

- Dataset-based Agent: an agent that extracts its data from datasets, such as CSV, Excel, SQL databases, etc.;
- Driver-based Agent: an agent that extracts its data from real devices or sources using communication protocols, such as ModBus/TCP, HTTP/S;
- Hybrid Agent: an agent that uses both datasets and drivers to extract its data;
- Event-driven Agent: an agent that does not use any type of data source, instead, its behavior is driven by events and interactions with the PEAK ecosystem.

MAS are mostly used for simulations, and without exception, PEAK allows the creation of simulation environments. This is possible since the XMPP has a useful functionality called PubSub, which stands for Publisher/Subscriber paradigm. PubSub allows agents to create nodes in the XMPP server, named publisher nodes where other agents can subscribe to it and receive any publication made to that node. In the simulation environment, it is needed an agent named Synchronizer, responsible to synchronize the MAS clock. The Synchronizer will create a publisher node in the XMPP server that will synchronize all the agents subscribed to it. In simulation mode, all the agents must subscribe to that node to receive the clock synchronization messages.

The Synchronizer has an internal clock that is used to mark the rhythm of the simulation. This clock has three properties: the time between tics (i.e., periods), the velocity of the simulation, which can be changed in runtime, and the number of tics that the simulation lasts.

With the PEAK MAS Platform, it is possible to create a MAS ecosystem. To be able to create these ecosystems it was used the XMPP functionality of Multi-User Chat. This XMPP functionality allows the creation of chat groups that can have many different utilities inside a MAS. The way the PEAK module uses this functionality is to divide each MAS into different groups. This allows the messages to be exchanged only between the same MAS/group. Despite that, this mechanism does not restrict the communication between different MAS, which can be achieved by using the agents' peer-to-peer communication, provided by the XMPP server.

3.2 Management

PEAK Management is a module focused on managing and monitoring the ecosystems created by the PEAK framework. To make that possible, this module uses the agent named Directory Facilitator (DF) which is responsible for exchanging the information through the user interface and the XMPP system.

For an administrator to interact with the system, they could use the web graphical interface. This web interface can either be the one implemented internally in the PEAK Management module or another one developed by a third party.

This interface will then communicate with DF agent. This agent has two main objectives: be an administrative tool for the administrator to get information about the MAS that are running on the ecosystem and serves as a centralized service provider for the agents in the ecosystem.

For DF to work as an administrative tool it needs to have administrative privileges in the XMPP server. This must be done when configuring the XMPP server, by adding the DF to the administrators' list, enabling the following functionalities:

- See the list and state of each user/agent in the server;
- See the list of the groups' chats, including the list of users/agents;
- Create and delete users/agents as well as groups.

The administrator privileges give access also to the messages exchanged between the agents. Is with this access that the DF can use the Communication Tracking mechanism to monitor and debug the agents' messages and interactions in PEAK.

Furthermore, the DF agent offers two important services that are available to every agent in the MAS: the Tree Hierarchy and the Yellow Pages Service.

The Tree Hierarchy is based on the XMPP Multi-User Chat functionality, but because it does not have any hierarchy mechanism for the groups, was created this functionality in the DF agent. The agents can send a message to the DF containing the group they want to register and the parent group to which is associated. The DF then searches for the group, if it does not exist it creates the room, otherwise, the agent joins the existing group. While this happens the DF constructs the hierarchy tree of the groups that can be accessed using the web interface. The Yellow Page Service is a service that can be used by the other agents to register, search, and use services from other agents, including from other MAS.

4 Case Study

To test the PEAK framework, it was created a case study of an energy community with 50 consumer agents, of which 15 of them are producers, with renewable energy generation. One prosumer represents a public library, while the others are regular households. One agent is the community manager that will gather the energy information about every consumer in the community and will publish the market prices for them. In addition, six consumers have one BESS each. Three of those BESS have 2.4 kWh capacity and the other three 3.6 kWh. The simulation lasts 24 h and the energy calculation and agents' interaction will be done in 15 min periods. In Fig. 2 it is possible to see the energy generation and consumption of the day that will be simulated. As is natural to see, the consumption is higher at mid-day and dinner time. The energy generation is bigger at mid-day when there is more solar energy. In Fig. 3, it is shown the data regarding solar radiation and the outside humidity.

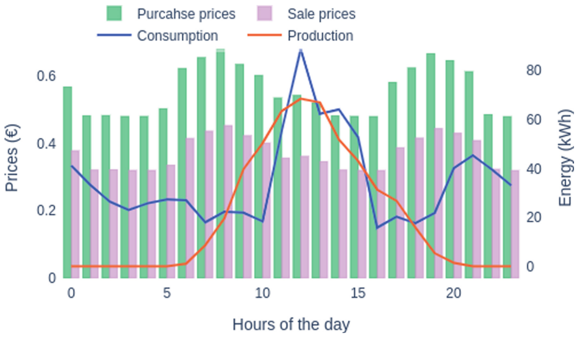


Fig. 2. Energy profile of the energy community and market prices.

Figure 2 shows the hour-ahead market prices used in the simulations. For the sale prices, it was used the MIBEL market of 10th March of 2022, and for the purchasing prices, it was multiplied the sale prices by the value of 1.5 to account for the distribution company profit. The market is higher either for selling and buying at mid-day and night.

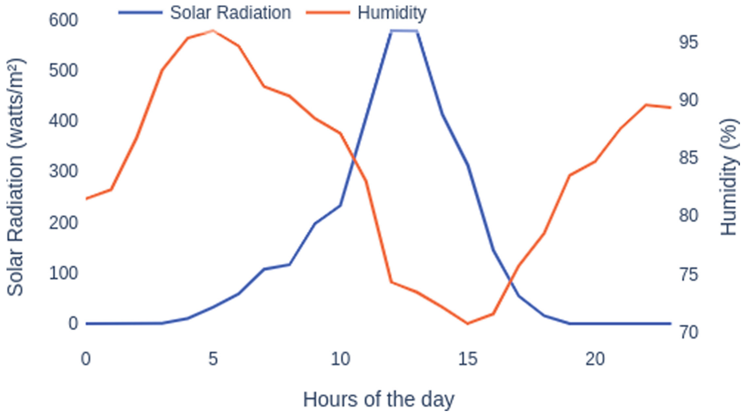


Fig. 3. Solar radiation and humidity of the simulated day.

For this case study, two simulations were made simultaneously. The differences between the simulations are the use of BESS and the simulation speeds. In the simulation that will not be used BESS the speed of the simulation will be 0.2 s per period, while the simulation with BESS will be run in real-time (15 min per period). The real-time simulation is necessary as the BESS are physical devices available in our labs, and, therefore, they cannot handle a rapid increase in the charge/discharge speed. The integration of BESS is done using a Modbus/TCP driver that can monitor and control the BESS according to the agent's needs. The case study also uses HTTP drivers to access the real-time weather data.

The results of the simulations can be seen in Fig. 4. The graph shows the financial balance of consumer 1, which is one of the prosumers with a BESS. The utilization of the BESS can decrease the financial cost throughout the day.

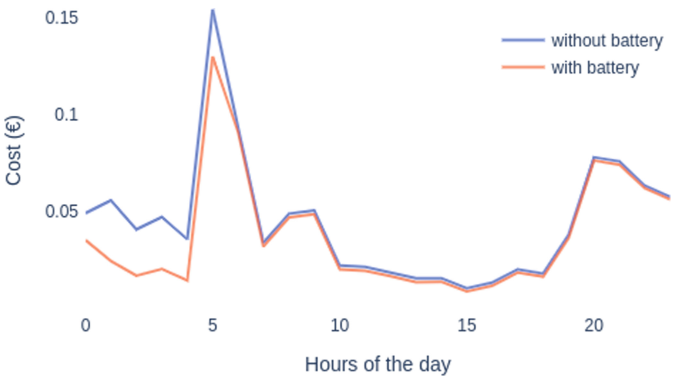


Fig. 4. Financial balance of consumer 1 with and without BESS

Relatively to the simulations, the framework handled both at the same time, each with different speeds, using two MAS, one for each simulation type. The simulations did not overlap each other and the XMPP server was able to handle every message sent inside each simulation.

5 Conclusion

The decentralized architecture, described in this paper, is a great solution when it comes to solving problems regarding the application of multi-agent systems in the energy domain. It is a flexible and easy framework to modulate various multi-agent systems in the same ecosystem and framework, while it provides tools for the proper management of the agents. In addition, it also provides tools for various types of simulations. The preliminary results of the PEAK framework demonstrate a wide range of possibilities that enable the test and validation of management models in smart grids by allowing the individual representation of entities and players.

For future work, the PEAK framework must be developed to be able to handle different types of drivers as well as improve the graphical interface of the PEAK Management module to be possible to see the network more interactively. Regarding the use of battery energy storage systems in energy communities, it would be interesting to see optimization algorithms implemented to minimize energy costs and improve the community's sustainability.

Acknowledgments. This work has received funding from FEDER Funds through COMPETE program and from National Funds through (FCT) under the project PRECISE (PTDC/EEI-EEE/6277/2020). The authors acknowledge the work facilities and equipment provided by GECAD research center (UIDB/00760/2020) to the project team.

References

1. Colak, A.: Introduction to smart grid. (Aug. 2016). https://doi.org/10.1109/ISGWCP.2016.7548265
2. Mengelkamp, E., Notheisen, B., Beer, C., Dauer, D., Weinhardt, C.: A blockchain-based smart grid: towards sustainable local energy markets. Comput. Sci. Res. Dev. **33**(1–2), 207–214 (2017). https://doi.org/10.1007/s00450-017-0360-9
3. Gomes, L., Vale, Z., Corchado, J.M.: Microgrid management system based on a multi-agent approach: An Office building pilot. Measurement **154**, 107427 (2020). https://doi.org/10.1016/J.MEASUREMENT.2019.107427. Mar
4. Faria, P., Vale, Z.: Distributed energy resource scheduling with focus on demand response complex contracts. J. Modern Power Sys. Clean Energy **9**(5), 1172–1182 (2021). https://doi.org/10.35833/MPCE.2020.000317. Sep
5. Pinto, T., Faia, R., Navarro-Caceres, M., Santos, G., Corchado, J.M., Vale, Z.: Multi-agent-based CBR recommender system for intelligent energy management in buildings. IEEE Syst. J. **13**(1), 1084–1095 (2019). https://doi.org/10.1109/JSYST.2018.2876933. Mar

6. Gazafroudi, A.S., et al.: Organization-based multi-agent structure of the smart home electricity system. In: 2017 IEEE Congress on Evolutionary Computation, CEC 2017 - Proceedings, pp. 1327–1334 (Jul. 2017). https://doi.org/10.1109/CEC.2017.7969458
7. Khan, M.W., Wang, J., Ma, M., Xiong, L., Li, P., Wu, F.: Optimal energy management and control aspects of distributed microgrid using multi-agent systems. Sustainable Cities and Society, vol 44, pp. 855–870. Elsevier (Jan. 01 2019). https://doi.org/10.1016/j.scs.2018.11.009
8. Klaimi, J., Rahim-Amoud, R., Merghem-Boulahia, L., Jrad, A.: A novel loss-based energy management approach for smart grids using multi-agent systems and intelligent storage systems. Sustain. Cities Soc. **39**, 344–357 (2018). https://doi.org/10.1016/j.scs.2018.02.038. May
9. Gomes, L., Vale, Z.A., Corchado, J.M.: Multi-agent microgrid management system for single-board computers: a case study on peer-to-peer energy trading. IEEE Access **8**, 64169–64183 (2020). https://doi.org/10.1109/ACCESS.2020.2985254
10. Ren, Y., Fan, D., Feng, Q., Wang, Z., Sun, B., Yang, D.: Agent-based restoration approach for reliability with load balancing on smart grids. Appl. Energy **249**, 46–57 (2019). https://doi.org/10.1016/j.apenergy.2019.04.119. Sep
11. Mansour, A.M., Obeidat, M.A., Abdallah, J.: A Novel Multi-agent Mechanism for Managing Electrical Power Smart Grids (Apr. 2021). https://doi.org/10.1109/IREC51415.2021.9427850
12. Panisson, A.R., Bordini, R.H.: Towards a computational model of argumentation schemes in agent-oriented programming languages. In: Proceedings - 2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, WI-IAT 2020, pp. 9–16 (Dec. 2020). https://doi.org/10.1109/WIIAT50758.2020.00007
13. Chen, F., Ren, W.: On the Control of Multi-Agent Systems: A Survey. Foundations and Trends® in Systems and Control **6**(4), 339–499 (2019). https://doi.org/10.1561/2600000019
14. Reis, I.F.G., Gonçalves, I., Lopes, M.A.R., Antunes, C.H.: A multi-agent system approach to exploit demand-side flexibility in an energy community. Utilities Policy **67**, 101114 (2020). https://doi.org/10.1016/j.jup.2020.101114. Dec
15. Prasad, A., Dusparic, I.: Multi-agent Deep Reinforcement Learning for Zero Energy Communities (Sep. 2019). https://doi.org/10.1109/ISGTEurope.2019.8905628
16. Algarvio, H.: Management of local citizen energy communities and bilateral contracting in multi-agent electricity markets. Smart Cities **4**(4), 1437–1453 (2021). https://doi.org/10.3390/smartcities4040076. Nov
17. An, A.R., Liu, G.P., Tan, C.: Group consensus control for networked multi-agent systems with communication delays. ISA Trans. **76**, 78–87 (2018). https://doi.org/10.1016/J.ISATRA.2018.03.008. May
18. Ge, X., Han, Q.L., Ding, D., Zhang, X.M., Ning, B.: A survey on recent advances in distributed sampled-data cooperative control of multi-agent systems. Neurocomputing **275**, 1684–1701 (2018). https://doi.org/10.1016/J.NEUCOM.2017.10.008. Jan
19. Wang, A., Wang, Z., Chen, M., Wang, W.: Distributed optimization for multi-agent systems with constraints set and communication time-delay over a directed graph. Inf. Sci. **438**, 1–14 (2018). https://doi.org/10.1016/J.INS.2018.01.040. Apr
20. Hasanuzzaman Shawon, M., Muyeen, S.M., Ghosh, A., Islam, S.M., Baptista, M.S.: Multi-agent systems in ICT enabled smart grid: A status update on technology framework and applications. IEEE Access **7** (2019). https://doi.org/10.1109/ACCESS.2019.2929577
21. Bergenti, F., Caire, G., Monica, S., Poggi, A.: The first twenty years of agent-based software development with JADE. Auton. Agent. Multi-Agent Syst. **34**(2), 1–19 (2020). https://doi.org/10.1007/S10458-020-09460-Z/FIGURES/1. Oct

22. Palanca, A., Terrasa, A., Julian, V., Carrascosa, C.: Spade 3: Supporting the new generation of multi-agent systems. IEEE Access **8**, 182537–182549 (2020). https://doi.org/10.1109/ACCESS.2020.3027357
23. de Freitas, B.K., Fritzen Venturini, L., Domingues, M.A., Augusto Da Rosa, M., Issicaba, D.: Exploiting PADE to the simulation of multiagent restoration actions. In: 2019 11th International Symposium on Advanced Topics in Electrical Engineering, ATEE 2019 (Mar. 2019). https://doi.org/10.1109/ATEE.2019.8724852

Deep Learning Approach for the Prediction of the Concentration of Chlorophyll ɑ in Seawater. A Case Study in *El Mar Menor* (Spain)

Javier González-Enrique[1](✉), Juan Jesús Ruiz-Aguilar[1], Eduardo Madrid Navarro[2], Rosa Martínez Álvarez-Castellanos[2], Ivan Felis Enguix[2], José M. Jerez[3], and Ignacio J. Turias[1]

[1] Intelligent Modelling of Systems Research Group, ETSI Algeciras, University of Cádiz, Avd. Ramón Puyol s/n, 11202 Cádiz, Algeciras, Spain
javier.gonzalezenrique@uca.es

[2] Centro Tecnológico Naval y del Mar (CTN), 30320 Fuente Álamo, Murcia, Spain
ivanfelis@ctnaval.com

[3] Department of Computer Science, ETS Computer Science, University of Málaga, Bulevar Louis Pasteur, 35. Campus de Teatinos, 29071 Málaga, Spain
jmjerez@uma.es

Abstract. The goal of this research is to develop accurate and reliable forecasting models for chlorophyll ɑ concentrations in seawater at multiple depth levels in *El Mar Menor* (Spain). Chlorophyll ɑ can be used as a eutrophication indicator, which is especially essential in a rich yet vulnerable ecosystem like the study area. Bayesian regularized artificial neural networks and Long Short-term Memory Neural Networks (LSTMs) employing a rolling window approach were used as forecasting algorithms with a one-week prediction horizon. Two input strategies were tested: using data from the own time series or including exogenous variables among the inputs. In this second case, mutual information and the Minimum-Redundancy-Maximum-Relevance approach were utilized to select the most relevant variables. The models obtained reasonable results for the univariate input scheme with $\overline{\sigma}$ values over 0.75 in levels between 0.5 and 2 m. The inclusion of exogenous variables increased these values to above 0.85 for the same depth levels. The models and methodologies presented in this paper can constitute a very useful tool to help predict eutrophication episodes and act as decision-making tools that allow the governmental and environmental agencies to prevent the degradation of *El Mar Menor*.

Keywords: Chlorophyll · Machine learning · Deep learning · LSTM · Artificial neural networks · Mar Menor · Eutrophication · Bayesian regularization

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 72–85, 2023.
https://doi.org/10.1007/978-3-031-18050-7_8

1 Introduction

El Mar Menor is a hypersaline coastal lagoon located in the Region of Murcia (Spain). With a surface area of 135 km^2, it is the largest saltwater lagoon in Europe [1]. During the last decades, increasing inputs of nutrients (nitrates, as well as phosphate and ammonium) and organic matter into *El Mar Menor* have led to progressive eutrophication of the lagoon, with several unprecedented crises exacerbated by heavy rain events in 2016, 2019, and 2021, which have caused deep ecologic, social, and economic impacts in the system [2]. Since the great crisis of 2016, environmental data such as temperature, salinity, chlorophyll a, and dissolved oxygen, among others, have been systematically acquired in oceanographic campaigns to accurately assess the status of the lagoon [3]. These data are sampled in 24 different measurement locations, making a rich database with both temporal and spatial dependence. However, the potential of this dataset has not yet been much exploited, as only one data-based modeling approach study has been published according to our knowledge [4].

In this study, this dataset has been used to model the dynamics of chlorophyll a in the lagoon, using Bayesian regularized neural networks and LSTMs to extract knowledge from the complex system and ultimately aid in the critical decision-making processes that are currently being debated. To achieve this goal, several models have been created, tested, and compared to obtain accurate predictions of the mass concentrations of chlorophyll a in seawater with a prediction horizon of one week.

Several interesting works devoted to the prediction of the concentration of chlorophyll a in lakes and seawater can be found in the scientific literature. In the work of Yu et al. [5], authors combined LSTMs and wavelets to predict the concentrations in Dianchi lake (China) with good results. Cho and Park [6] proposed a new methodology using merged-LSTM models to predict this concentration in a river in Korea. This approach made it possible to use exogenous variables and outperformed standard LSTMs and multilayer perceptron models. Finally, Shin et al. [7] employed several prediction techniques, including recurrent neural networks (RNNs) and LSTMs, combined with a rolling window approach to predict the chlorophyll a concentration levels in the Nakdong River (Korea).

The rest of this paper is organized as follows. Section 2 presents a description of the database and the study area. Section 3 describes the techniques employed in this work. The experimental procedure is explained in Sect. 4. Section 5 discusses the results. Finally, the main conclusions are indicated in Sect. 6.

2 Area Description and Datasets

El Mar Menor is a saltwater lagoon separated from the Mediterranean Sea by a 22 km sand coastal barrier, known as *La Manga del Mar Menor* (see Fig. 1).

This lagoon and its ecosystem have suffered a continuous degradation process in the last decades due to anthropogenic pressure [8]. One of the main threats can be found in the huge urban growth in the surrounding area, with cities such as Cartagena. Additionally, a very intense agricultural activity in *El Campo de Cartagena* causes the discharge of nutrients. Finally, mining activities near *La Unión* cause the acid drainage of heavy metals into its waters.

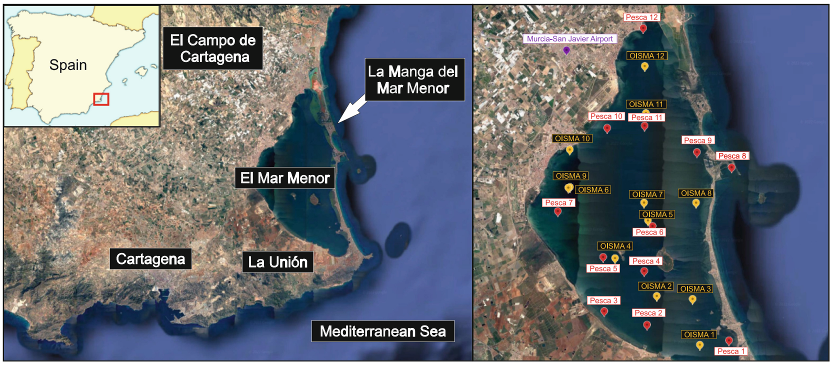


Fig. 1. Localization of *El Mar Menor* and the monitoring stations

The dataset used in this work has been provided by The Scientific Data Server of *El Mar Menor*. This institution collects data from two different sets of monitoring stations deployed in the study area, which can be seen in Fig. 1. One of these sets belongs to the Office for Socioeconomic Promotion of the Environment (stations labeled as OISMA) and the other one to the Fishing Service (stations labeled as PESCA). Data covers the period from May 2017 to October 2021 and includes several variables measured weekly in each of the stations that are shown in Table 1. These measures were originally taken in a range of periods between 5 and 12 days. Thus, they were interpolated to obtain their weekly values. Additionally, these variables are measured in each station at different depth levels: 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, and 5 m. However, these data are offered as an interpolation of the primary data collected in the stations so that values for each coordinate of the surface of the lagoon are provided. These data needed to be preprocessed to obtain values for each specific station and variable pair. Therefore, for each timestep of the variable, the median values of the surrounding coordinates to the specific station in a radius of 100 m were calculated. Additionally, several meteorological variables measured in the same period at the Murcia-San Javier Airport (see Fig. 1), were also included.

Table 1. Description of the variables included in the database

Variable	Unit	Monitoring stations
Mass concentration of chlorophyll α in seawater	mg/m^3	OISMA, PESCA
Mass concentration of oxygen in seawater	mg/l	OISMA, PESCA
Seawater phycoerythrin	ppm	OISMA, PESCA
Seawater salinity	PSU	OISMA, PESCA
Seawater temperature	°C	OISMA, PESCA
Seawater turbidity	FTU	OISMA, PESCA
Mean temperature	°C	Murcia-San Javier Airport

(*continued*)

Table 1. (*continued*)

Variable	Unit	Monitoring stations
Rainfall	mm	Murcia-San Javier Airport
Minimum temperature	°C	Murcia-San Javier Airport
Maximum temperature	°C	Murcia-San Javier Airport
Wind direction	°	Murcia-San Javier Airport
Average wind speed	m/s	Murcia-San Javier Airport
Insolation	hours	Murcia-San Javier Airport

3 Methods

This section describes the techniques and methods used in the present manuscript. A schematic representation of the experimental procedure is presented in Fig. 2.

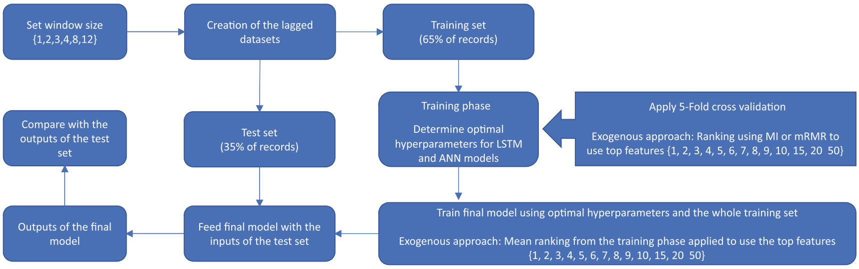

Set window size {1,2,3,4,8,12}
Creation of the lagged datasets
Training set (65% of records)
Compare with the outputs of the test set
Test set (35% of records)
Training phase
Determine optimal hyperparameters for LSTM and ANN models
Apply 5-Fold cross validation
Exogenous approach: Ranking using MI or mRMR to use top features {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20 50}
Outputs of the final model
Feed final model with the inputs of the test set
Train final model using optimal hyperparameters and the whole training set
Exogenous approach: Mean ranking from the training phase applied to use the top features {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20 50}

Fig. 2. Schematic representation of the experimental procedure

3.1 Artificial Neural Networks

Artificial Neural Networks (ANNs) are computational models inspired by the way biological neurons interconnect and can be seen as a non-linear procedure that can map a set of inputs into a set of outputs. The most extensively used design for ANNs is the Feedforward Multilayer Perceptron [9]. In this type of artificial network, the learning process is based on error backpropagation and its architecture consists of an input layer, one or more hidden layers, and an output layer.

Due to their ability to estimate any nonlinear function, Feedforward Neural Networks can be considered universal approximators [10]. Nonetheless, there are significant drawbacks: there is no standard method for determining the optimal number of hidden neurons, and ANN models are susceptible to experiencing overfitting issues.

3.2 Bayesian Regularized Neural Networks

Bayesian regularized artificial neural networks (BRANNs) emerge as an alternative to reduce the drawbacks that were previously mentioned. This type of network uses the Bayesian regularization technique [11] to improve the robustness of the models and reduce the likelihood of overfitting. During the training process, the goal is to reduce the model's sum squared error. Additionally, a penalty term is introduced to the error function to manage the complexity of the models.

3.3 Long Short-Term Memory Neural Networks

Long short-term memory networks [12] are a type of recurrent neural network especially suited for learning long-term dependencies in sequence data. They include a special type of unit, called memory block, to avoid the vanishing gradient problem that affects standard RNNs [13]. As a result, LSTMs are especially appropriate for time-series forecasting.

The typical architecture of a long short-term memory network is composed of an input layer, a hidden layer containing the memory blocks, and an output layer. The hidden layer includes several self-connected memory cells in a chainlike configuration. Each memory block includes an input gate, an output gate and a forget gate in its inner structure. These gates let memory blocks perform read, write, and erase operations and give memory cells the capacity of storing long-term dependencies.

3.4 Mutual Information

The mutual information (MI) [14] calculates the amount of information that one of two random vectors contains about the other. MI is defined by Eq. (1) and is based on Shannon's information theory.

$$MI(x, y) = \iint p(x, y) \log \frac{p(x, y)}{p(x) \cdot p(y)} dxdy \tag{1}$$

where *x* and *y* are two continuous random vectors, *p(x, y)* is their joint probability density, and *p(x)* and *p(y)* are their marginal probability densities.

3.5 Minimum-Redundancy-Maximum-Relevance (mRMR)

The Minimum-Redundancy-Maximum-Relevance method [15] is a feature ranking technique that allows ranking a group of features based on how relevant they are to the target variable. Simultaneously, it penalizes superfluous characteristics. As a result, the highest ranked features are those that offer the finest balance of maximal relevance with the goal variable and the lowest redundancy with the remaining attributes. Mutual information is used to calculate relevancy and redundancy between variables in this technique. The pseudocode corresponding to the mRMR algorithm tailored to regression problems can be found in [16].

4 Experimental Procedure

The objective of this work is to obtain accurate predictions of the mass concentration of chlorophyll a for specific monitoring stations located in *El Mar Menor*, with a prediction horizon of one week. The OISMA-1 monitoring station was selected to conduct the experiments and forecasting models were created to predict the chlorophyll a concentrations in all of its ten depth levels (from 0.5 to 5 m, every 0.5 m). Two techniques were compared to calculate these future concentrations: Bayesian regularized artificial neural networks and sequence-to-sequence LSTMs. This latter technique was tailored to use a rolling window approach in conjunction with cross-validation for time series. A similar approach has been previously used by the authors with very promising results [17]. Additionally, multiple linear regression models (MLR) have also been included.

The database contained weekly measures of mass concentration of chlorophyll a and several other variables, including meteorological variables, measured from 2017 to 2021 (see Sect. 2). All these data were preprocessed, subject to an imputation of missing values procedure, and then standardized and normalized.

4.1 Creation of the Lagged Datasets

A rolling window approach was followed with both forecasting techniques. The input values included present and previous values of the variables in a certain window size *ws*. This term denotes the number of input values that comprise the autoregressive window. Window sizes of 1, 2, 3, 4, 8, and 12 weeks were employed in this study. The prediction horizon k was set to one step ahead (one week in the future). Equation (2) depicts how the outcomes were modeled.

$$\hat{y}(t+1) = F\{y(t), y(t-1), ..., y(t-(ws-1))\} \tag{2}$$

For a given depth level in the OISMA-1 station, two input data approaches were tested to calculate the forecasting models. In the first case, a univariate scheme was employed and only past concentrations of chlorophyll a measured at this specific level were used to make the predictions. The output variable y corresponds to the input values shifted one-time step. The new lagged variables were then created using samples of consecutive observations. The datasets were defined as $\mathcal{D}_{k,ws} = \{x_{ws_i}, y_{k_i}\}_{i=1}^{T}$ where T indicates the number of samples and k represents the prediction horizon used. In the case of the exogenous input scheme, several variables measured at any of the stations and depth levels (see Table 1) were added to the set of initial inputs. Each of these time series had to be transformed into new lagged variables using the same window sizes and procedure as in the univariate input scheme. Feature ranking methods were applied in a later step to determine the most relevant among these lagged input variables. Thus, a group of potential input variables Q_{ws} was created including all the previously created lagged variables. Finally, new datasets were created, including these variables and the output variable: $\mathcal{D}'_{k,ws} = \{Q_{ws_i}, y_{k_i}\}_{i=1}^{T}$.

Once the lagged datasets were created for each window size and input scheme, the training subset consisted of the first 65% of the records while the test subset included the remaining 35%.

4.2 Forecasting Models

For each window size, models were trained using the univariate and exogenous training subsets. Several input cases were tested when the exogenous training subset was used. These cases were defined by the number of relevant features kept from the group of lagged variables (the top 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20 and 50 features), and the feature ranking technique used (MI or mRMR). Three performance indexes were used to evaluate the accuracy of the models: the correlation coefficient (σ), the mean square error (MSE) and the mean absolute error (MAE).

4.2.1 BRANN Models

The networks employed a single fully connected hidden layer and several hidden unit configurations were tested (from 1 to 25). For each of these configurations, a 5-fold cross-validation procedure for time series [18] was followed. In this procedure, data is partitioned into sequential folds of fixed indexes respecting the temporal order. This entire procedure was repeated 20 times, and the average results were calculated and stored. Additionally, each repetition's record was also saved to be used in a later multicomparison procedure.

After completing all the repetitions for each hidden layer configuration, a multicomparison process was used to find the simplest model with no statistically significant changes from the top-performing model. The Friedman test [19] and the Bonferroni method [20] were applied to the sets of 20 records previously stored for each model. It was possible to determine whether there were relevant statistical differences across models with varying numbers of hidden units using the Friedman test. If these differences were confirmed, the Bonferroni method allowed us to determine which of these models were statistically equivalent to the best-performing model. In the case of a set of models that were not statistically different, the model with the fewest hidden units was chosen using the Occam's razor principle.

This same procedure was applied using the univariate and exogenous approaches. However, in this latter approach, only the top variables from the group of potential input lagged variables were utilized. This is achieved by applying a feature ranking method (mutual information or mRMR). Since the training and test sets changed in every inner cross-validation, the ranking of the potential features was recalculated and stored in each case.

After determining the optimal number of hidden units, a final model was trained using the whole lagged training dataset. For the exogenous approach, the ranking corresponding to the input case used was obtained as each variable's mean position in the rankings stored in the training phase. Once trained, the inputs of the unseen test lagged dataset were used to feed this model. Then, forecasting values were obtained, and performance metrics were determined by comparing forecasts to measured values. This process was repeated 50 times, and the average performance results were then calculated.

4.2.2 LSTM Models

LSTM models were created using sequence-to-sequence LSTMs. This type of network can map an input sequence of values of T time instants $x = (x_1, \ldots, x_T)$ to an output sequence of values $y = (y_1, \ldots, y_T)$.

The process followed is similar to the one described for the BRANN models. The training subset was used to determine the optimal hyperparameters for the models using a Bayesian optimization procedure [21]. The Bayesian optimization algorithm attempts to minimize an objective function where its inputs belong to a bounded domain. In our case, the objective function corresponds to each training model, and the bounded inputs correspond to its parameters (see Table 2). The *bayesopt* MATLAB function was employed to achieve this goal with a limit of 250 objective function evaluations for each model. The root mean square error guided this optimization process. For each of them, the lagged training dataset was subjected to a 5-fold cross-validation for time series. As was previously stated, every time the inner cross-validation subsets changed when using the exogenous dataset, the ranking of the features was recalculated and stored. The Adam optimizer was employed to train the models. The parameters used to train the models and the network architecture employed are shown in Table 2.

Table 2. LSTM architecture and parameters employed in the training phase

Layer number	Layer name	Parameter	Range of values
1	Sequence input layer	LSTM neurons	1–1000
2	LSTM layer	Initial learning rate	0.0005–0.09
3	Dropout layer	Dropout probability	0.01–0.99
4	Fully connected layer	Gradient decay factor	0.5–0.999
5	Output layer	L2 regularization factor	0.00005–0.0009
		Minibatch size	$\{2^n\}_{n=2}^{12}$

Following the discovery of the optimal parameters, a sequence-to-sequence final LSTM model was trained using the entire lagged training dataset. In this case, the ranking corresponding to the input case was obtained as the mean position of each variable in the rankings stored during the training phase. Once trained, the unseen test lagged dataset's input sequence was used to feed this model. The predicted values were thus obtained. Performance measures were computed by comparing these values to the test lagged dataset output sequence. This procedure was repeated 50 times, and the average performance results were calculated.

5 Results and Discussion

The results of the experimental procedure are presented in this section. Models were built to predict the mass concentration of chlorophyll α in seawater in the OISMA-1 monitoring station for each of its 10 depth levels (see Fig. 1). A one-step ahead prediction horizon was set (one week in the future, as the database records were measured weekly). BRANNs and LSTMs using a rolling window approach were employed as the forecasting techniques, using different window sizes (1, 2, 3, 4, 8, and 12 weeks). Depending on the dataset utilized, their results were compared in two different scenarios (univariate or exogenous datasets, see Sect. 4). In the exogenous dataset approach, mutual information and the mRMR method were applied to keep only the top features. Thus, for each window size models were built using the top 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, and 50 variables.

Table 3 shows the average performance indexes of the best-performing models using the univariate input approach. In this table, *D* indicates the depth level in meters, *ws* depicts the window size, *nh* is the number of units in the hidden layer, *DP* corresponds to the dropout probability, *GDF* stands for the gradient decay factor, *LR* corresponds to the learning rate, *MB* indicates the minibatch size and *L2R* is the level 2 regularization factor.

Table 3. Best performing models using the univariate inputs approach

D (m)	*Method*	*ws*	*nh*	$\overline{\sigma}$	$\overline{MSE}$	$\overline{MAE}$	*DP*	*MB*	*LR*	*L2R*	*GDF*
0.5	BRANN	2	11	0.722	1.771	0.591	–	–	–	–	–
	LSTM	3	523	0.779	1.465	0.547	0.194	8	0.029	8.297E−04	0.876
	MLR	2	–	0.720	1.800	0.601	–	–	–	–	–
1	BRANN	2	2	0.787	1.745	0.592	–	–	–	–	–
	LSTM	4	487	0.787	1.666	0.814	0.691	32	0.047	2.687E−04	0.526
	MLR	1	–	0.777	1.702	0.674	–	–	–	–	–
1.5	BRANN	2	4	0.809	1.639	0.736	–	–	–	–	–
	LSTM	4	759	0.797	1.794	0.760	0.451	32	0.034	7.827E−04	0.552
	MLR	1	–	0.773	1.970	0.736	–	–	–	–	–
2	BRANN	3	1	0.772	2.128	0.850	–	–	–	–	–
	LSTM	2	453	0.770	2.110	0.890	0.711	16	0.023	8.457E−04	0.635
	MLR	1	–	0.745	2.396	0.859	–	–	–	–	–
2.5	BRANN	2	12	0.747	1.803	0.842	–	–	–	–	–
	LSTM	4	336	0.703	2.104	0.932	0.170	2048	0.049	5.304E−04	0.588
	MLR	4	–	0.643	2.625	0.853	–	–	–	–	–
3	BRANN	1	5	0.591	2.730	1.008	–	–	–	–	–
	LSTM	4	326	0.569	2.815	1.135	0.068	64	0.036	7.023E−04	0.506
	MLR	2	–	0.530	3.249	1.043	–	–	–	–	–
3.5	BRANN	3	1	0.573	3.340	1.069	–	–	–	–	–
	LSTM	3	835	0.595	3.154	1.216	0.891	16	0.011	5.830E−04	0.608
	MLR	2	–	0.552	3.620	1.225	–	–	–	–	–

(*continued*)

Table 3. (*continued*)

D (m)	*Method*	*ws*	*nh*	$\overline{\sigma}$	$\overline{MSE}$	$\overline{MAE}$	*DP*	*MB*	*LR*	*L2R*	*GDF*
4	BRANN	3	1	0.553	3.014	0.984	–	–	–	–	–
	LSTM	3	999	0.545	2.977	1.026	0.048	32	0.061	5.829E−04	0.678
	MLR	1	–	0.479	3.750	1.068	–	–	–	–	–
4.5	BRANN	3	1	0.788	2.192	0.932	–	–	–	–	–
	LSTM	4	841	0.761	2.511	1.030	0.028	32	0.020	6.890E−04	0.518
	MLR	1	–	0.788	2.193	0.992	–	–	–	–	–
5	BRANN	4	1	0.678	3.126	0.875	–	–	–	–	–
	LSTM	2	997	0.665	3.070	0.890	0.017	32	0.050	8.819E−04	0.593
	MLR	3	–	0.662	3.253	0.893	–	–	–	–	–

Table 4 shows the average performance indexes of the best–performing models for the exogenous input scheme. In this table, *RK* denotes the type of feature filter ranking technique, MI indicates the mutual information and MR denotes the mRMR method. Additionally, *VAR* indicates the number of variables selected in the model.

Table 4. Best performing models using the exogenous inputs approach

D(m)	*Method*	*ws*	*RK*	*VAR*	*nh*	$\overline{\sigma}$	$\overline{MSE}$	$\overline{MAE}$	*DP*	*MB*	*LR*	*L2R*	*GDF*
0.5	BRANN	3	MR	2	1	0.829	1.043	0.513	–	–	–	–	–
	LSTM	1	IM	7	315	0.852	0.828	0.427	0.345	16	0.065	3.383E−04	0.530
	MLR	3	MR	5	–	0.803	1.21	0.514	–	–	–	–	–
1	BRANN	3	IM	4	2	0.848	1.176	0.818	–	–	–	–	–
	LSTM	4	MR	10	136	0.893	1.089	0.767	0.879	8	0.019	5.546E−05	0.576
	MLR	1	IM	1	–	0.767	1.720	0.721	–	–	–	–	–
1.5	BRANN	1	IM	1	8	0.854	1.125	0.683	–	–	–	–	–
	LSTM	2	MR	20	140	0.885	1.045	0.695	0.191	16	0.080	4.673E−04	0.872
	MLR	1	IM	6	–	0.810	1.720	0.766	–	–	–	–	–
2	BRANN	8	MR	5	1	0.801	1.394	0.818	–	–	–	–	–
	LSTM	1	IM	10	992	0.860	1.194	0.838	0.162	8	0.007	3.729E−04	0.785
	MLR	2	MR	5	–	0.796	1.924	0.824	–	–	–	–	–
2.5	BRANN	8	IM	2	7	0.722	1.434	0.836	–	–	–	–	–
	LSTM	4	MR	10	658	0.815	1.203	0.812	0.720	16	0.008	5.826E−04	0.851
	MLR	1	MR	3	–	0.651	2.498	0.942	–	–	–	–	–
3	BRANN	1	IM	1	5	0.657	1.575	0.992	–	–	–	–	–
	LSTM	2	IM	8	57	0.722	1.507	1.129	0.012	8	0.008	5.781E−04	0.799
	MLR	2	MR	3	–	0.551	3.054	1.230	–	–	–	–	–

(*continued*)

Table 4. (*continued*)

D(m)	*Method*	*ws*	*RK*	*VAR*	*nh*	$\overline{\sigma}$	$\overline{MSE}$	$\overline{MAE}$	*DP*	*MB*	*LR*	*L2R*	*GDF*
3.5	BRANN	4	MR	5	3	0.640	1.717	1.007	–	–	–	–	–
	LSTM	1	MR	7	610	0.687	1.564	0.945	0.732	8	0.007	8.446E−04	0.865
	MLR	1	MR	3	–	0.600	3.386	0.983	–	–	–	–	–
4	BRANN	2	MR	2	1	0.589	1.812	1.037	–	–	–	–	–
	LSTM	3	MR	20	927	0.663	1.557	0.842	0.167	8	0.056	4.775E−04	0.660
	MLR	1	IM	3	–	0.550	3.90	1.009	–	–	–	–	–
4.5	BRANN	2	MR	2	2	0.816	1.381	0.877	–	–	–	–	–
	LSTM	2	MR	2	972	0.822	1.386	0.859	0.264	32	0.023	1.595E−04	0.504
	MLR	1	MR	1	–	0.788	2.193	0.992	–	–	–	–	–
5	BRANN	2	MR	1	5	0.704	1.663	0.817	–	–	–	–	–
	LSTM	2	MR	4	866	0.702	1.682	0.867	0.695	16	0.030	3.744E−04	0.506
	MLR	4	MR	3	–	0.670	3.042	0.856	–	–	–	–	–

The performance of the models is reasonable in the univariate input scheme for a prediction of one-week ahead values. However, results show how the inclusion of the exogenous variables greatly improves the performance of the models.

Regarding the type of forecasting technique used, BRANNs and LSTMs show similar results in the univariate approach, with slight differences between their models. However, LSTMs using the rolling window approach perform better in most of the levels that were considered, especially with the exogenous approach. The best performing MLR models produce the overall worst performances among all the techniques employed. This tendency is more pronounced in the exogenous approach.

The main advantages of the LSTM models lie in their performance, especially when the exogenous input scheme is employed. However, the computation times for this type of model are the highest of the three techniques considered. In the case of the BRANN models, an advantage consists of providing good all-around performance with lower computation times than the LSTM models. Finally, the main advantage of the MLR models is their fastest computation speeds at the expense of performance.

Another interesting fact is that the window sizes used in the best-performing models are fairly small, been two weeks the most widely used value in the univariate input scheme and three weeks in the exogenous case. It is also worth mentioning that the mRMR method is used in 67% of the exogenous models, which confirms the goodness of this technique in avoiding redundancies in the feature selection process.

Finally, the influence of the depth level is also important, and can be observed that there is a very appreciable worsening of the results between 2.5 and 5 m. This fact is much more notorious in the univariate input scheme.

6 Conclusions

This paper aims to produce accurate and reliable forecasting models to predict the chlorophyll α concentrations in seawater in one of the monitoring stations of *El Mar Menor* at several depth levels. Bayesian regularized neural networks and LSTMs using a rolling window approach were employed as the forecasting techniques with a prediction horizon of one week. Additionally, MLR models were also included. Two different input schemes were also tested: using data from the own time series or including exogenous variables among the inputs. The high number of variables produced in this second case made it unfeasible to use all the lagged variables produced. Hence, mutual information and the mRMR method were used to select the very top variables.

Results showed how the models built provided reasonable results in the univariate input scheme for a prediction horizon of a week, with $\overline{\sigma}$ values above 0.75 in levels between 0.5 and 2 m. The inclusion of exogenous variables boosted these values above 0.85 and nearly 0.90 for the same levels. For levels between 2.5 and 5 m (especially from 3 to 4 m), a decrease in the models' performance can be observed, which is much more accused in the univariate input approach. This fact suggests that the inclusion of exogenous variables provides useful information that could help to explain the chlorophyll a concentrations in seawaters. At these depth levels, LSTM models presented a more accurate behavior than BRANNs. In contrast, MLR models produced the worst performances of all the techniques that were compared.

A plausible reason that could explain this phenomenon is that *El Mar Menor* presents a clear stratification in both its water temperature and current speed, with two distinct layers separated at around 3 m depth level. This intermediate region may be subjected to a higher degree of variability, not only for these variables but for others as well. In fact, at least during the extreme precipitation event in September 2019, the massive amount of freshwater injected into the lagoon caused a stratification in the water column with respect to the salinity, which led to the apparition of two water layers of different salinity and density, with the interface around the 3–4 m depth level. Moreover, dissolved oxygen also stratified around this level as well, overall causing a severe euxinic event that lasted more than two weeks [2]. In addition, when considering the distribution of all chlorophyll a time series, a higher variability is found as depth increases, which could be favored by the higher mixing of the superior layer; that is, the chlorophyll a dynamics of the superficial waters is more stable than those at deeper levels.

Finally, the chlorophyll a concentrations can play a pivotal role as a eutrophication indicator. This becomes extremely important in locations and environments such as *El Mar Menor*. This very rich ecosystem has suffered a continuous degradation process in the last years due to anthropogenic pressure. Therefore, the forecasting models and methodologies presented in this paper can help to predict eutrophication episodes and act as decision-making tools that allow environmental agencies to prevent its degradation. Additionally, these methodologies could also be applied to other possible eutrophication indicators, such as the concentration of oxygen in seawater, by just switching the input datasets.

Acknowledgments. This work is part of the research projects RTI2018–098160-B-I00 supported by MICINN (Ministerio de Ciencia e Innovación-Spain) and "EnviroPorts. Investigación de técnicas de análisis y predicción de series temporales de parámetros ambientales y su interrelación con tráfico marítimo en entornos portuarios". Expediente 2021.08.CT01.000044. Financiado por el Instituto de Fomento de la Región de Murcia (INFO) dentro del PROGRAMA DE AYUDAS DESTINADAS A CENTROS TECNOLÓGICOS DE LA REGIÓN DE MURCIA DESTINADAS A LA REALIZACIÓN DE ACTIVIDADES I+D DE CARÁCTER NO ECONÓMICO. MODALIDAD 1: "PROYECTOS I+D INDEPENDIENTE".

References

1. Perez-Ruzafa, A., Marcos, C., Gilabert, J.: The ecology of the Mar Menor coastal lagoon: A fast changing ecosystem under human pressure. In: Coastal Lagoons: Ecosystem Processes and Modeling for Sustainable Use and Development (2005)
2. Ruiz, J.M., et al.: Informe de evolución y estado actual del Mar Menor en relación al proceso de eutrofización y sus causas. Informe de asesoramiento técnico del Instituto Español de Oceanografía (IEO) (2022)
3. U. P. Cartagena Servidor de Datos Cientificos del Mar Menor. https://marmenor.upct.es/docs/. Accessed 6 May 2022
4. Jimeno-Sáez, P., Senent-Aparicio, J., Cecilia, J., Pérez-Sánchez, J.: Using machine-learning algorithms for eutrophication modeling: case study of Mar Menor Lagoon (Spain). Int. J. Environ. Res. Public Health **17**, 1189 (2020)
5. Yu, Z., Yang, K., Luo, Y., Shang, C.: Spatial-temporal process simulation and prediction of chlorophyll-a concentration in Dianchi Lake based on wavelet analysis and long-short term memory network. J. Hydrol. **582**, 124488 (2020)
6. Cho, H., Park, H.: Merged-LSTM and multistep prediction of daily chlorophyll-a concentration for algal bloom forecast. IOP Conf. Ser. Earth Environ. Sci. **351**, 012020 (2019). https://doi.org/10.1088/1755-1315/351/1/012020
7. Shin, Y.: Prediction of chlorophyll-a concentrations in the Nakdong river using machine learning methods. Water **12**, 1822 (2020)
8. Pérez-Ruzafa, A., Marcos, C., Pérez-Ruzafa, I.M., Barcala, E., Hegazi, M.I., Quispe, J.: Detecting changes resulting from human pressure in a naturally quick-changing and heterogeneous environment: Spatial and temporal scales of variability in coastal lagoons. Estuarine, Coastal Shelf Sci. **75**(1–2), 175–188 (2007). https://doi.org/10.1016/j.ecss.2007.04.030
9. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. In: Parallel distributed processing: explorations in the microstructure of cognition, vol. 1, pp 318–362. MIT Press, Cambridge, MA, USA (1986)
10. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Netw. **2**, 359–366 (1989)
11. MacKay, D.J.C.: Bayesian interpolation. Neural Comput. **4**, 415–447 (1992)
12. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**, 1735–1780 (1997)
13. Hochreiter, S.: The vanishing gradient problem during learning recurrent neural nets and problem solutions. Int. J. Unc., Fuzz. Knowl.-Based Syst. **6**, 107–116 (1998)
14. Shannon, C.E.: A mathematical theory of communication. Bell Syst. Tech. J. **27**, 379–423 (1948)
15. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans. Pattern Anal. Mach. Intell. **27**, 1226–1238 (2005)
16. González-Enrique, J., Ruiz-Aguilar, J.J., Moscoso-López, J.A., Urda, D., Turias, I.J.: A comparison of ranking filter methods applied to the estimation of NO_2 concentrations in the Bay of Algeciras (Spain). Stoch. Environ. Res. Risk Assess. **35**(10), 1999–2019 (2021). https://doi.org/10.1007/s00477-021-01992-4

17. González-Enrique, J., et al.: Artificial neural networks, sequence-to-sequence LSTMs, and exogenous variables as analytical tools for NO_2 (air pollution) forecasting. A case study in the Bay of Algeciras (Spain). Sensors **21**, 1770 (2021)
18. Bergmeir, C., Benítez, J.M.: On the use of cross-validation for time series predictor evaluation. Inf. Sci. (Ny) **191**, 192–213 (2012)
19. Friedman, M.: The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J. Am. Stat. Assoc. **32**, 675–701 (1937)
20. Hochberg, Y., Tamhane, A.C.: Multiple Comparison Procedures. John Wiley & Sons Inc., New York, NY, USA (1987)
21. Snoek, J., Larochelle, H., Adams, R.P.: Practical Bayesian Optimization of Machine Learning Algorithms. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, pp. 2951–2959. Curran Associates, Inc. (2012)

Evolutionary Computing

A Hybrid Discrete Symbiotic Organisms Search Algorithm and List-Based Simulated Annealing Algorithm for Traveling Salesman Problem

Vladimir Ilin(✉), Dragan Simić, Marko Veličković, and Nemanja Garunović

Faculty of Technical Sciences, University of Novi Sad, Trg Dositeja Obradovića 6,
21000 Novi Sad, Serbia
{v.ilin,dsimic,marvel,garunovic}@uns.ac.rs

Abstract. A discrete symbiotic organisms search (DSOS) and list-based simulated annealing (LBSA) are new metaheuristic search algorithms used for solving different complex optimization problems. DSOS mimics the symbiotic relationship strategies adopted by organisms in the ecosystem for survival, while LBSA simplify parameter tuning of the simulated annealing algorithm. In this paper, we propose a hybrid algorithm, named DSOS-LBSA, to solve the well-known traveling salesman problem (TSP) which belongs to the class of NP-hard problems. Additionally, an arbitrary insertion algorithm is introduced to produce organisms in the initial ecosystem. The proposed DSOS-LBSA is implemented in the MATLAB environment and it is tested on symmetric and asymmetric instances from TSPLIB. The overall results demonstrate that the proposed DSOS-LBSA offers promising results, particularly for small-size symmetric instances and large-size asymmetric instances.

Keywords: Discrete symbiotic organisms search algorithm · List-based simulated annealing algorithm · Arbitrary insertion algorithm · Traveling salesman problem · Hybrid approach

1 Introduction

The traveling salesman problem (TSP) is a well-known combinatorial optimization problem in the fields of computer sciences which attracts the attention of researchers for a long period of time. The problem is formulated as follows: given a set of cities and the routes between each pair of cities, there is a task to find the shortest path that passes through a set of n cities so that each city is visited exactly once. The TSP belongs to the class of *NP*-hard problems which is proved by Karp in 1975 [8]. That implies that there are no polynomial-time algorithms for TSP unless $P = NP$.

To solve different TSPs, many exact and approximate algorithms have been developed. Exact solutions include branch and bound, cutting planes, dynamic programming, linear programming and other methods. Approximate algorithms include heuristics, metaheuristics, and various hybrid approaches. In this article, our efforts are

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 89–97, 2023.
https://doi.org/10.1007/978-3-031-18050-7_9

focused towards two newly proposed metaheuristic algorithms for TSP, named a symbiotic organisms search (SOS) algorithm and a list-based simulated annealing (LBSA) algorithm.

A SOS algorithm is an intelligent optimization algorithm inspired by the symbiosis interactions, such as mutualism, commensalism and parasitism, among organisms in real-world ecosystems [2]. The symbiosis interactions offer organisms the opportunity to increase probability for their survival in the evolution process. Similarly to some other metaheuristics, such as genetic algorithm (GA), artificial bee colony algorithm (ACO), particle swarm optimization algorithm (PSO), and harmony search algorithm (HS), SOS belongs to the class of population-based metaheuristic algorithms. In SOS, each new population is produced using symbiosis interactions between organisms. However, SOS differs from other population-based metaheuristics in terms of number of specific parameters for fine-tuning during the implementation [3].

Originally, the SOS algorithm is developed to solve continuous engineering optimization problems [2]. However, several studies have shown its potential to be applied in the field of discrete optimization techniques, such as TSP. In two recent studies, different discrete symbiotic organisms search (DSOS) algorithms are design to solve TSP [3, 13]. The obtained results have shown that DSOSs retain the same advantages as SOS which makes it worth investigating even further. Therefore, in this paper we also propose a DSOS for TSP. The biggest issue of almost all heuristic algorithms, that is, premature convergence to local optimum, has been tackled in a different manner in this article compared to other approaches. Here, we have introduced a list-based simulated annealing (LBSA) algorithm [14] to be applied as an escape mechanism when the search being trapped into local optimum. The LBSA is applied in the following manner. In case that the search in the DSOS does not progress after a predefined number of generations, the LBSA is triggered to try to improve the current best solution. Similarly to DSOS, LBSA also has fewer parameter to be tunned during the search. This paper continuous the authors' previous researches in transportation planning [4–7].

The rest of the paper is organized in the following way. Section 2 and Sect. 3 overview the use of DSOS and LBSA for TSP, respectively. Section 4 introduces a hybrid algorithm named DSOS-LBSA. Experimental results and discussion are presented in Sect. 5, and finally, Sect. 6 provides concluding remarks.

2 A Discrete Symbiotic Organisms Search Algorithm for TSP

A discrete symbiotic organisms search (DSOS) algorithm is a discrete version of the SOS algorithm. In [3], the transformation methodology is based on the MATLAB function *round*. Here we introduce another control mechanism to assure a Hamiltonian cycle (HC) is completed. In case the HC is not completed, that is, numerical computation with the *round* function do not provide HC, then the additional control mechanism is activated. That new control mechanism is the ranking of vertices. In that manner, a HC is guaranteed for all transformations.

The DSOS starts by first randomly generating *n* number of organisms to populate the ecosystem. Each organism in the ecosystem represents a feasible solution with a fitness function. The search for the best solution starts after the organism with the best fitness function is found. Subsequently, the current best organism is updated in each evaluation phase of the algorithm, following a common symbiosis strategy. The three phases of symbiosis include mutualism, commensalism, and parasitism phase. The new organism is accepted to be included in the population only if it has a better fitness function than the previous organism. The search is iteratively repeated until the termination condition is reached. Two main parameters define computation time during the search. Those are outer and inner loop. The outer loop represents maximal number of iterations. The inner loop corresponds to the ecosystem size.

In the following subsections, we describe the mutualism phase, commensalism phase, and parasitism phase in more detail.

2.1 Mutualism Phase

In this phase, a mutualism operator is activated to produce new organisms. First, the organism $X_{i,1 \le i \le m}$ from the ecosystem (m is defined number of organisms) is observed and another random organism $X_{j,1 \le j \le m, j \ne i}$ from the ecosystem is selected for the interaction. The current best organism X_{best} is also used during a mutualistic relationship. The objective is to produce new organisms with better chances for survival in the ecosystem. Those new organisms are meant to learn from each other as well as from the X_{best}. The two new organisms, X_{i_new} and X_{j_new} are computed according to formulas (1) and (2), respectively.

$$X_{i\_new} = X_i + rand(0, 1) \times \left(X_{best} - \left(\frac{X_i + X_j}{2} \right) \times \alpha \right) \tag{1}$$

$$X_{j\_new} = X_j + rand(0, 1) \times \left(X_{best} - \left(\frac{X_i + X_j}{2} \right) \times \beta \right) \tag{2}$$

In formulas (1) and (2), the function *rand* generates a random value between 0 and 1, the term $\frac{X_i + X_j}{2}$ represents the relationships between X_i and X_j, and expressions α and β denote the mutual benefit factors, which represent the level of benefit that both X_i and X_j derive from the mutual association. Factors α and β are calculated using the following expression $\alpha = \beta = 1 + \text{round}[\text{rand}(0, 1)]$. The new organisms X_{i_new} and X_{j_new} are retained only if they provide better values of fitness function than X_i and X_j, respectively.

2.2 Commensalism Phase

In this phase, a commensalism operator is activated to produce new organism. Similar to mutualism phase, X_i, X_j, and X_{best} are first determined. The relationship interaction is such that only X_i tries to benefit from the interaction. The new organism, X_{i_new} is computed according to formula (3).

$$X_{i\_new} = X_i + rand(-1, 1) \times \left(X_{best} - X_j \right) \tag{3}$$

In formula (3), the function *rand* generates a random value between -1 and 1 and the term $\left(X_{best} - X_j \right)$ represents the benefit provided by the organism X_j to improve organism X_i. The new organisms X_{i_new} is retained only if it provides better values of fitness function than organism X_i.

2.3 Parasitism Phase

In this phase, a parasitism operator is activated to produce new organism. First, a parasite vector X_{pv} is generated by mutating X_i, that is X_i serve as a host to the X_{pv}. Then, the organism $X_{j,1 \le j \le m, j \ne i}$ is selected for comparison. The assessment of a new organism X_{pv} is performed in a way that if the value of its fitness function is better than the value of the fitness function of an organism X_j then the organism X_{pv} will be retained in the ecosystem. Here we used the three operators, named swap, inversion, and insertion, as parasitism operators. These operators were proposed by [12] and only one of them is used for computation based on a generated random value and defined probabilities for each operator.

3 A List-Based Simulated Annealing Algorithm for TSP

The LBSA algorithm is introduced to overcome the premature convergence of DSOS, that is, to try to escape the search being trapped into local optimum. Here we describe the pseudocode for the LBSA algorithm for TSP (Algorithm 1).

Algorithm 1 LBSA for TSP

```
Calculate the best current solution (parameter $x = BestCurr$) found by DSOS;
Generate an empty initial temperature list (parameter $L$) and the list length $Lmax$
Initialize the initial acceptance probability (parameter $p_0$)
Initialize an empty new population (parameter $Pnew$)
Initialize the number of inner and outer loops (parameters $M$ and $K$)
for $i = 1$ to $K$
    Fetch the maximum temperature from the list $L$: parameter $t_{max}$
    $c = 0$   % number of acceptances of bad solutions
    $t_{sum} = 0$   % temperature level summation
    for $j = 1$ to $M$
        Apply inverse, insert, swap operator and create a neighbor solution $y$
        if $f(y) \le f(x)$
            $x = y$; Append $Pnew$ with $x$
        else
            $p = e^{-(f(y)-f(x))/t_{max}}$   % $p$ is the acceptance probability
            Generate a random number $r \in [0,1)$
            if $r < p$
                $c = c + 1$
                $t_{sum} = t_{sum} + \left(-\left(f(y) - f(x)\right) / \ln(r)\right)$
                $x = y$; Append $Pnew$ with $x$
            end if
        end if
    end for
    if $c \sim = 0$
        Delete the highest temperature in $L$
        Append $L$ with $t_{sum} / c$
    end if
end for
return $Pnew$
```

We used the LBSA to capture the best solution found during the search. The mechanism that triggers the LBSA is defined on the basis of predefined number of generations in which search does not progress. The main parameters of the LBSA are: p_0, L_{max}, *K,* and *M.* Zhan et al. [14] showed that the performance of the LBSA is not too sensitive to the initial temperature values and the length of the temperature list, while it is highly sensitive to the parameters of inner and outer loops. Considering that the LBSA can be time consuming when combined with the DSOS, values for parameters *K* and *M* should be carefully selected.

4 A Hybrid DSOS-LBSA Algorithm for TSP

The hybrid DSOS-LBSA algorithm starts by first randomly generating *n/2* number of organisms to populate the first half of ecosystem. The second half of ecosystem is populated with organisms generated using Arbitrary insertion algorithm (AIA). An arbitrary insertion algorithm (AIA) was proposed by Rosenkrantz et al. [11].

In the remainder of this section we describe the pseudocode for the hybrid DSOS-LBSA (Algorithm 2).

Algorithm 2 Hybrid DSOS-LBSA for TSP

```
1:  Initialize first half of the ecosystem: randomly created feasible solutions
2:  Initialize second half of the ecosystem: the use of Arbitrary insertion algorithm
3:  Initialize ecosystem size (parameter eco_size)
4:  Initialize maximum iteration (parameter MaxIter)
5:  Define the best solution found for a specific instance (parameter OPTinstance)
6:  Search the current best solution from X (parameter Xbest)
7:  for i = 1 to MaxIter
8:      for j = 1 to eco_size
9:          Mutualism phase
10:         Commensalism phase
11:         Parasitism phase
12:         Update Xbest
13:         if Xbest ==OPTinstance
14:             return Xbest
15:         end if
16:         if Xbest is not improved after predefined number of generations
17:             Apply LBSA and capture new Xbest_new if found
18:             Xbest ← Xbest_new
19:         end if
20:     end for
21: end for
22: return Xbest
```

First, the initial ecosystem is produced using randomly generated organisms and organisms produced using AIA algorithm (Lines 1–2). The idea of using the AIA is to produce organisms with better fitness function compared to those that are generated randomly. The diversity of the first population is provided with randomly created organisms. Then, other setup parameters are defined (Lines 3–5) and fitness function is calculated

(Line 6). After that, for each iteration the search is performed using mutualism phase, commensalism phase, and parasitism phase (Lines 9–11). If the optimum solution is found, the algorithm is terminated (Lines 13–15). In case that after a predefined number of generations the search in DSOS does not progress, the LBSA is applied (Lines 16–19). The LBSA aims at finding the new best organism for setup parameters. Algorithm returns the best solution found during the search (Line 22).

5 Computational Results and Discussion

In order to demonstrate the effectiveness and performance of the hybrid DSOS-LBSA, we have implemented the proposed algorithm in the MATLAB environment and the experiments are run on a desktop computer with an Intel Core i5–2400, 3.1 GHz processor with 8GB RAM. The TSP experimental data sets used in this paper were obtained from MP-TESTDATA, which covers the following: (1) the TSPLIB symmetric TSPs and the best-known solutions for symmetric TSPs and (2) the TSPLIB asymmetric TSPs and the best-known solutions for asymmetric TSPs [10].

5.1 Parameter Settings

The parameter selection may significantly influence the solution's quality of each algorithm performance. In the hybrid DSOS-LBSA, the parameter configurations used for the proposed DSOS and LBSA is presented in Table 1.

Table 1. The parameter settings for both DSOS and LBSA

DSOS		LBSA	
Parameter	Value	Parameter	Value
MaxIter	500	*Lmax*	40
eco_size	50	*p0*	0.9
Swap prob. factor	0.2	*K*	800
Inverse prob. factor	0.5	*M*	400
Insert prob. factor	0.3		

For symmetric instances, the DSOS-LBSA algorithm was executed 20 times for each instance in order to make comparison with the GA-TCTIA-LBSA algorithm [4] and the EW-DHOA algorithm [9]. For asymmetric instances, the DSOS-LBSA algorithm was executed 30 times for each instance in order to make comparison with the GA-TCTIA-LBSA algorithm [4] and the HHS algorithm [1]. In the DSOS-LBSA algorithm, we used an additional parameter to track the number of generations in DSOS without improvement. This parameter is set to 50. When that value is reached, the LBSA algorithm is applied and the search for the improved solution is triggered.

5.2 Computational Results and Analysis

The obtained results for 15 symmetric TSP instances and 9 asymmetric TSP instances are presented in Table 2 and Table 3, respectively. Tested TSP symmetric instances range from 48 to 1002 cities and tested TSP asymmetric instances range from 33 to 443 cities.

The obtained results from the proposed hybrid DSOS-LBSA algorithm are compared with the results obtained from the GA-TCTIA-LBSA algorithm [4], the EW-DHOA algorithm [9], and the HHS algorithm [1].

Table 2. Comparison results between the proposed DSOS-LBSA and EW-DHOA and GA-TCTIA-LBSA on 15 symmetric benchmark instances from TSPLIB

	EW-DHOA [9]			**GA-TCTIA-LBSA [4]**			**DSOS-LBSA**		
Instance	**Best**	**Mean**	**PDav**	**Best**	**Mean**	**PDav**	**Best**	**Mean**	**PDav**
Att48	**33520**	33557	0.11	**33522**	33566.10	0.13	**33522**	33561.10	0.12
Eil51	437.59	469.45	10.20	**426**	426.65	0.15	**426**	426.30	0.07
St70	714	729	8.00	**675**	675.50	0.07	**675**	677.35	0.35
Eil76	601	618	14.87	**538**	543.60	1.04	**538**	540.35	0.44
KroD100	**21294**	21340	0.22	**21294**	21318.10	0.11	**21294**	21379.35	0.40
Eil101	688.99	703.2	11.80	**629**	634.75	0.91	**629**	634.50	0.87
Lin105	14400	14400	0.15	**14379**	14385.10	0.04	**14379**	14390.95	0.08
Ch130	6135.9	6181.8	1.18	**6110**	6183.95	1.21	6147	6204.80	1.55
KroA150	**26524**	26566	0.16	26525	26816.95	1.10	26771	27012.30	1.84
Rat195	**2318.3**	2343.8	1.10	2363	2409.35	3.72	2366	2396.35	3.16
KroA200	29375	29410	0.14	29550	29808.10	1.50	29858	30185.25	2.78
Lin318	**42029**	42059	0.07	42919	43582.20	3.69	42685	43042,55	2.41
Rat575	**6770**	6840	1.03	7141	7178.85	5.99	7028	7091.6	4.75
Rat783	**8800**	8840	0.45	9321	9408.65	6.84	9288	9373.95	6.45
Pr1002	**259000**	259000	0	272724	274804.60	6.08	275258	278944.25	7.70
Average of PDav(%)			3.30			2.17			2.20

Optimal values: Opt(Att48)=33522; Opt(Eil51)=426; Opt(St70)=675; Opt(Eil76)=538; Opt(KroD100)= 21294; Opt(Eil101)=629; Opt(Lin105)=14379; Opt(Ch130)=6110; Opt(KroA150)=26524; Opt(Rat195)=2323; Opt(KroA200)=29368; Opt(Lin318)=42029; Opt(Rat575)=6773; Opt(Rat783)=8806; Opt(Pr1002)=259045.

Table 3. Comparison results between the proposed DSOS-LBSA and HHS and GA-TCTIA-LBSA on 9 asymmetric benchmark instances from TSPLIB

	HHS [1]			**GA-TCTIA-LBSA [4]**			**DSOS-LBSA**		
Instance	**Best**	**Mean**	**PDav**	**Best**	**Mean**	**PDav**	**Best**	**Mean**	**PDav**
ftv33	**1286**	1310.83	1.93	**1286**	1286.00	0	**1286**	1312.43	2.05
ftv38	1532	1542.23	0.80	**1530**	1540.97	0.72	**1530**	1554.9	1.63
ft53	7135	7336.10	6.24	**6905**	7029.30	1.80	7049	7297.17	5.68
krol124p	37262	38540.63	6.38	36241	36625.10	1.09	37085	38159.3	5.32
ftv170	3047	3221.43	16.93	2838	2926.73	6.23	3063	3167.23	14.96
rbg323	1477	1537.00	15.91	1376	1409.37	6.29	1349	1366.13	3.03
rbg358	1369	1414.37	21.61	1228	1249.93	7.47	1186	1204.50	3.57
rbg403	2572	2625.07	6.49	2470	2483.43	0.75	**2465**	2468.87	0.16
rbg443	2862	2921.90	7.42	2731	2749.53	1.09	**2720**	2726.83	0.25
Average of PDav(%)			9.30			2.83			4.07

Optimal values: Opt(ftv33)=1286; Opt(ftv38)=1530; Opt(ft53)=6905; Opt(krol124p)=36230; Opt(ftv170)=2755; Opt(rbg323)=1326; Opt(rbg358)=1163; Opt(rbg403)=2465; Opt(rbg443)=2720.

In Table 2, the computed cumulative average percentage of the deviation (PDav) of the DSOS-LBSA packing for the 15 TSP symmetric instances is very similar to the GA-TCTIA-LBSA [4] and slightly better than the EW-DHOA [9]. Although the EW-DHOA [9] presented the worst performance of all three compared algorithms in terms of the PDav, at the same time, it exhibits the best results for instances greater than 130 cities. If we compare PDav for individual instances, the proposed DSOS-LBSA shows better results than the GA-TCTIA-LBSA [4] for 8 out of 15 tested instances. However, the proposed DSOS-LBSA shows better results than the EW-DHOA [9] only for 5 out of 15 tested instances. In terms of the quality of the best solution found the proposed DSOS-LBSA and the GA-TCTIA-LBSA [4] show similar results. The EW-DHOA [9] displays promising results for large-size TSP instances. In particular, the EW-DHOA [9] exhibits the optimal results for all tested instances greater than 200 cities.

In Table 3, the computed cumulative PDav of the DSOS-LBSA for the 9 TSP asymmetric instances is worse than the GA-TCTIA-LBSA [4] and better than the HHS [1]. If we compare PDav for individual instances, the proposed DSOS-LBSA shows better results than the GA-TCTIA-LBSA [4] only for 4 out of 9 tested instances. However, the proposed DSOS-LBSA shows better results than the HHS [1] for 7 out of 9 tested instances. In terms of the quality of the best solution found the proposed DSOS-LBSA exhibits the best results. In particular, the DSOS-LBSA displays promising results for large-size TSP instances.

6 Conclusion and Future Work

In this study, the hybrid DSOS-LBSA is proposed to solve symmetric and asymmetric TSPs. The DSOS is used to improve current best solution in each iteration. In case the search becomes trapped into local optimum, the LBSA is introduced to try to improve the current best solution. In addition, the AIA is introduced to produce organisms for the initial ecosystem. The presented algorithm is tested using 24 benchmark instances ranging from 33 to 1002 cities. The overall results demonstrate that the proposed DSOS-LBSA exhibits promising results, particularly for small-sized symmetric TSP instances and large-sized asymmetric TSP instances.

The obtained results from the proposed DSOS-LBSA algorithm could have application in various industries, such as logistics and shipping in general. The asymmetric TSPs directly correspond to real-life situations in the cities with one-way or temporarily closed streets.

The future work could focus on extending the research on different strategies of how to improve local search in order to increase the convergence speed of the proposed algorithm. In addition, the computational results section can be extended with the results of some new algorithms and the parameter computational time complexity can be considered for all compared algorithms.

Acknowledgment. This research (paper) has been supported by the Ministry of Education, Science and Technological Development through project no. 451-03-68/2022-14/200156 “Innovative scientific and artistic research from the FTS (activity) domain”.

References

1. Boryczka, U., Szwarc, K.: An effective hybrid harmony search for the asymmetric travelling salesman problem. Eng. Optim. **52**(2), 218–234 (2020)
2. Cheng, M.Y., Prayogo, D.: Symbiotic organisms search: a new metaheuristic optimization algorithm. Comput. Struct. **139**(15), 98–112 (2014)
3. Ezugwu, A.E.S., Adewumi, A.O.: Discrete symbiotic organisms search algorithm for travelling salesman problem. Expert Syst. Appl. **87**(30), 70–78 (2017)
4. Ilin, V., Matijevic, L., Davidovic, T., Pardalos, P.M.: Asymmetric capacitated vehicle routing problem with time window. In: Proc. XLV Symposium on Operations Research, SYM-OP-IS 2018, pp. 174–179. Faculty of Economics in Belgrade (2018)
5. Ilin, V., Simić, D., Simić, D.S., Simić, S., Saulić, N., Calvo-Rolle, J.L.: A hybrid genetic al-gorithm, list-based simulated annealing algorithm, and different heuristic algorithms for travelling salesman problem, Log. J. IGPL (2022, in press)
6. Ilin, V., Simić, D., Simić, S.D., Simić, S.: Hybrid genetic algorithms and tour construction and improvement algorithms used for optimizing the traveling salesman problem. In: Herrero, Á., Cambra, C., Urda, D., Sedano, J., Quintián, H., Corchado, E. (eds.) SOCO 2020. AISC, vol. 1268, pp. 530–539. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-57802-2_51
7. Ilin, V., Simić, D., Tepić, J., Stojić, G., Saulić, N.: A survey of hybrid artificial intelligence algorithms for dynamic vehicle routing problem. In: Onieva, E., Santos, I., Osaba, E., Quintián, H., Corchado, E. (eds.) HAIS 2015. LNCS (LNAI), vol. 9121, pp. 644–655. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19644-2_53
8. Karp, R.M.: On the computational complexity of combinatorial problems. Networks **5**(1), 45–68 (1975)
9. Rajesh Kanna, S.K., Sivakumar, K., Lingaraj, N.: Development of deer hunting linked earthworm optimization algorithm for solving large scale traveling salesman problem. Knowl-Based Syst. **227**, 107199 (2021)
10. Reinelt, G.: TSPLIB. http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/. Last Accessed 05 Apr 2022
11. Rosenkrantz, D., Stearns, R., Lewis, P.: Approximate algorithms for the traveling salesper-son problem. In: Proceedings of the 15th Annual IEEE Symposium of Switching and Au-tomata Theory, pp. 33–42. IEEE (1974)
12. Wang, C.Y., Lin, M., Zhong, Y.W., Zhang, H.: Solving travelling salesman problem using multiagent simulated annealing algorithm with instance-based sampling. Int. J. Comput. Sci. Math. **6**(4), 336–353 (2015)
13. Wang, Y., Wu, Y.W., Xu, N.: Discrete symbiotic organism search with excellence coeffi-cients and self-escape for traveling salesman problem. Comput. Ind. Eng. **131**, 269–281 (2019)
14. Zhan, S.H., Lin, J., Zhang, Z.J., Zhong, Y.W.: List-based simulated annealing algorithm for traveling salesman problem. Comput. Intell. Neurosci. **2016**, 1–12 (2016)

Estimation of Distribution Algorithms Applied to the Next Release Problem

Víctor Pérez-Piqueras(✉), Pablo Bermejo López, and José A. Gámez

Department of Computing Systems, Intelligent Systems and Data Mining Laboratory (I3A), Universidad de Castilla-La Mancha, 02071 Albacete, Spain
{victor.perezpiqueras,pablo.bermejo,jose.gamez}@uclm.es

Abstract. The Next Release Problem (NRP) is a combinatorial optimization problem that aims to find a subset of software requirements to be delivered in the next software release, which maximize the satisfaction of a list of clients and minimize the effort required by developers to implement them. Previous studies have applied various metaheuristics and procedures, being many of them evolutionary algorithms. However, no Estimation of Distribution Algorithms (EDA) have been applied to the NRP. This subfamily of evolutionary algorithms, based on probability modelling, have been proved to obtain good results in problems where genetic algorithms struggle. In this paper we adapted two EDAs to tackle the multi-objective NRP, and compared them against widely used genetic algorithms. Results showed that EDA approaches have the potential to generate solutions of similar or even better quality than those of genetic algorithms in the most crowded areas of the Pareto front, while keeping a shorter execution time.

1 Introduction

Successfully managing software releases is one of the major challenges in Software Engineering. As the product goal grows and project scope gets expanded, the difficulty of delivering to clients what is needed increases substantially. Clients interests are usually defined in terms of software requirements, and these software requirements are built based on clients interests. Thus, when there is more than one client, their concerns are usually different or even opposed. Furthermore, complexity of requirements has to be taken into account, in order to not surpass development capacity at each iteration of the development cycle. Finally, the problem can get even more complex if the possible dependencies between requirements are taken into account.

This problem, named NRP, pursues finding a set of requirements for a release that satisfy clients as much as possible and optimize development efforts. This is a NP-hard problem which does not have a unique and optimal solution, and is usually solved manually by experts judgement. Given that solving the NRP is critical for a software project success, and that it has to be solved iteratively every time a release is planned, it is an interesting candidate to be automated

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 98–108, 2023.
https://doi.org/10.1007/978-3-031-18050-7_10

by means of optimization methods. In previous works, different search techniques have been proposed to tackle the NRP. Most of them are based on metaheuristic techniques, being evolutionary algorithms the ones that showed best performance. In this paper, we introduce the possibility of applying Estimation of Distribution Algorithms (EDA) to the NRP. As a first approach, we have applied two univariate EDAs, and compared them against widely used genetic algorithms (GA). Experimentation results show that EDAs can be successfully applied to the NRP and, while evolved populations are not covering all search space areas, they can overcome GAs solutions in specific sections of the space.

The rest of the paper is structured as follows. In Sect. 2, a summary of previous works and applied procedures is made. Section 3 introduces EDA and describes our two proposed algorithms and solution encoding. Then, in Sect. 4, the evaluation setup is described, listing the algorithms, datasets and methodology used. Section 5 presents and discusses the results of the experimentation. Finally, Sect. 6 summarizes the conclusions of this study and introduces potential lines of research.

2 Next Release Problem

2.1 Related Work

The solution of the NRP is one of the applications of the field of Search-Based Software Engineering (SBSE), where Software Engineering related problems are tackled by means of search-based optimization algorithms.

The NRP was firstly formulated by Bagnall et al. [2]. In its definition, a subset of requirements has to be selected, having as goal meeting the clients needs, minimizing development effort and maximizing clients satisfaction. They applied a variety of metaheuristics techniques, such as simulated annealing, hill climbing and GRASP, but combining the objectives of the problem into a single-objective function.

Other works started formulating the NRP as a multi-objective optimization (MOO) problem, being the first one the proposal of Zhang et al. [15]. This new formulation, Multi-Objective Next Release Problem (MONRP) was based on Pareto dominance [4] and is formally defined in Sect. 2.2. In their proposal, they tackled each objective separately, exploring the non-dominated solutions (NDS). Finkelstein et al. [7] also applied multi-objective optimization considering different measures of fairness. All these studies applied evolutionary algorithms, such as ParetoGA and NSGA-II [5] to solve the MONRP.

Although EDA approaches have been applied to SBSE problems, none has been used to tackle the NRP, to the authors' knowledge. From the most recent reviews, in Ramírez et al. [12] only an EDA application to software testing [13] is referenced; and in Gupta et al. [9] and Alba et al. [1] EDA approaches are not mentioned or matched to any solution of the NRP. Thus, we find it of interest to develop new EDA-based algorithms to be applied to the NRP.

Comparison among study proposals has been generally carried out by analyzing the Pareto fronts and execution time returned by the algorithms. However, later works started using a set of numerical metrics that evaluate different features of the Pareto fronts. Most studies compare the metrics obtained from their proposals against those obtained by other algorithms commonly applied to MOO, analyzing the performance and the advantages and weaknesses of using each algorithm to solve the MONRP. Based on this, we provide a similar comparison framework for our proposal.

2.2 Multi-objective Next Release Problem

As mentioned in the introduction, the NRP requires a combinatorial optimization of two objectives. While some studies alleviate this problem by adding an aggregate (Single-Objective Optimization), others tackle the two objectives by using a Pareto front of NDS, using multi-objective optimization (MOO).

In MOO, there is no unique and optimal solution, but a Pareto front of NDS [4]. The Pareto front is a vector or set of configuration values for the decision variables that satisfies the problem constraints and optimizes the objective functions. Thus, the Pareto front contains a set of solutions that are not dominated by any other. Given a solution vector $x = [x_1, x_2, \ldots, x_j]$ where j is the number of problem objectives, it dominates a solution vector $y = [y_1, y_2, \ldots, y_j]$ if and only if y is not better than x for any objective $i = 1, 2, \ldots, j$. In addition, there must exist at least one objective x_i that is better than the respective y_i of y. Conversely, two solutions are non-dominated as long as neither of them dominates the other.

Defining the NRP as a multi-objective optimization problem gives the advantage that a single solution to the problem is not sought, but rather a NDS set. In this way, one solution or another from this set can be chosen according to the conditions, situation and restrictions of the software product development. This new formulation of the problem is known as MONRP.

The MONRP can be defined by a set $R = \{r_1, r_2, \ldots, r_n\}$ of n candidate software requirements, which are suggested by a set $C = \{c_1, c_2, \ldots, c_m\}$ of m clients. In addition, a vector of costs or efforts is defined for the requirements in R, denoted $E = \{e_1, e_2, \ldots, e_n\}$, in which each e_i is associated with a requirement r_i [10]. Each client has an associated weight, which measures its importance. Let $W = \{w_1, w_2, \ldots, w_m\}$ be the set of client weights. Moreover, each client gives an importance value to each requirement, depending on the needs and goals that this has with respect to the software product being developed. Thus, the importance that a requirement r_j has for a client c_i is given by a value v_{ij}, in which a zero value represents that the client c_i does not have any interest in the implementation of the requirement r_j. A $m \times n$ matrix is used to hold all the importance values in v_{ij}. The overall satisfaction provided by a requirement r_j is denoted as $S = \{s_1, s_2, \ldots, s_n\}$ and is measured as a weighted sum of all importance values for all clients, that is: $s_j = \sum_{i=1}^{m} w_i \times v_{ij}$. The MONRP consists of finding a decision vector X, that includes the requirements to be implemented for the next software release. X is a subset of R, which contains the requirements that maximize clients satisfaction and minimize development efforts.

The MONRP objectives are the following:

$$Maximize\ S(X) = \sum_{j \in X} s_j \quad Minimize\ E(X) = \sum_{j \in X} e_j \tag{1}$$

In addition, requirements in vector X might have to satisfy the constraints of the problem. These constraints are related to the interactions between requirements and to the total effort of the development.

3 Proposal: Univariate EDAs for the MONRP

EDAs are evolutionary algorithms based on probabilistic modelling and were designed as an alternative to genetic algorithms (GAs). As GAs, EDAs are population-based algorithms, however instead of relying upon the goodness of genetic operators, EDAs apply a more normative approach, which consists of learning a probability distribution from a set of promising individuals of the current population and sampling the estimated distribution in order to obtain the next population [11]. As no crossover nor mutation operator is needed, the number of hyperparameters decreases, thus simplifying their configuration. The complexity of an EDA is related with the degree of explicit interrelations (dependencies) it allows. Thus, in univariate EDAs no explicit dependency is allowed and interrelations are implicitly catched by the evaluation function (as in GAs), but when multivariate EDAs are used, the dependencies among the variables are explicitly modeled.

Formally, univariate EDAs assume that the n-dimensional joint probability distribution (JPD) factorizes like a product of n univariate and independent probability distributions, that is $p_l(x) = \prod_{i=1}^{n} p_l(x_i)$. Thus framework fits well in our goal for this study as we are tackling the MONRP without dependencies between requirements, therefore we propose to adapt two univariate EDAs to work in the domain of this multi-objective problem: UMDA and PBIL. Before describing each algorithm, let us to consider some common issues: an individual is represented by a vector of booleans of length n, where each value indicates the inclusion or not of a requirement of the set R; both the satisfaction and effort of each requirement are scaled using a min-max normalization; since we only consider univariate EDA, we have not modeled cost restrictions nor interactions between requirements (as in related works [6,7,15]); finally, as we are tackling the MONRP, algorithms must return only NDS, so at each iteration of the execution algorithms update a NDS set with the new generated individuals.

Algorithm 1. MONRP-UMDA pseudocode

```
procedure MONRP-UMDA($maxGenerations$)
    $nds \leftarrow \emptyset$                    ▷ empty set of non-dominated solutions
    $P \leftarrow$ generateRandomPopulation()
    for $i = 0$ to $maxGenerations$ do
        $individuals \leftarrow$ selectIndividuals($P$)
        $probModel \leftarrow$ learnProbModel($individuals$)
        $P \leftarrow$ sampleNewPopulation($probModel$)
        evaluate($P$)
        $nds \leftarrow$ updateNDS($P, nds$)
    end for
    return $nds$
end procedure
```

3.1 MONRP-UMDA

In Univariate Marginal Distribution Algorithm (UMDA) [11, Chapter 3] the JPD is factorized as the product of marginal distributions: $p_l(x;\theta^l) = \prod_{i=1}^{n} p_l(x_i;\theta_i^l)$. The MONRP-UMDA (Algorithm 1) starts creating a random population and, at each generation, it selects the non-dominated individuals of the population, learns the probability model p_l from them, and samples a new population using p_l. Finally, new individuals are evaluated and the global NDS set updated by adding the new non-dominated individuals found. After execution, it returns the NDS set.

3.2 MONRP-PBIL

Probability Based Incremental Learning (PBIL) [11, Chapter 3] combines the mechanisms of a generational GA with simple competitive learning. Differently to UMDA, in which populations are transformed into a probability model whose only purpose is to sample new populations, PBIL algorithm attempts to create a probability model which can be considered a prototype for high evaluation vectors for the function space being explored. In a manner similar to the training of a competitive learning network, the values in the probability model are gradually shifted towards representing those in high evaluation vectors.

MONRP-PBIL (Algorithm 2) is quite similar to MONRP-UMDA, except in the updating of the probabilistic model. First, instead of selecting always the best individual found so far, at each iteration we randomly sampled an individual from the NDS set. Then, we update the probability model (vector) in two steps:

(1) $V_i = V_i \cdot (1.0 - LR) + bestIndividual_i \cdot LR$, being i the i^{th} gene position, and LR the learning rate hyperparameter, ranging from 0 to 1.
(2) If $(Prob_{rand} < Prob_{mut})$: $V_i = V_i \cdot (1.0 - MS) + r \cdot MS$, r being a random number $\in \{0, 1\}$ and MS the mutation shift hyperparameter, ranged from 0 to 1.

Algorithm 2. MONRP-PBIL pseudocode

```
procedure MONRP-PBIL($maxGenerations, LR, Prob_{mut}, MS$)
    $nds \leftarrow \emptyset$                              ▷ empty set of non-dominated solutions
    $probModel \leftarrow$ initProbModel()                  ▷ set all vector initial values to 0.5
    for $i = 0$ to $maxGenerations$ do
        $P \leftarrow$ sampleNewPopulation($probModel$)
        evaluate($P$)
        $bestInd \leftarrow$ selectBestIndividual($P$)
        $probModel \leftarrow$ updateProbModel($probModel, bestInd, LR, Prob_{mut}, MS$)
        $nds \leftarrow$ updateNDS($P, nds$)
    end for
    return $nds$
end procedure
```

4 Experimental Evaluation

In this section, we present the experimental method used in the evaluation. Then, we describe other algorithm approaches used to be compared against our proposal, along with the datasets used to evaluate the algorithms. The source code for the algorithms, implemented in Python 3.8.8, along with the experimentation setup and datasets used is available at the following repository: https://github.com/uclm-simd/monrp/tree/soco22.

4.1 Algorithms

Our experimentation framework includes the following algorithms to compare performance and effectiveness of the two multi-objective univariate EDA versions:

- **Random search.** This is a baseline algorithm, expected to be outperformed by all algorithms. The procedure generates as many random solutions as the maximum number of evaluation functions specified. Then, it returns the NDS set.
- **Single-Objective GA.** It combines the two objective functions of the MONRP into a single objective. Then, updates the NDS set with new individuals after each generation.
- **NSGA-II.** The Non-dominated Sorting Genetic Algorithm-II [5] is a state-of-the-art multi-objective GA. It uses elitism and ranks each individual based on the level of non-dominance.

The ranges of parameters used in the experimentation for each algorithm are described in Sect. 4.3, for further detail.

4.2 Datasets

Algorithms performance has been tested using two widely used public datasets (P1 and P2), taken from previous NRP works. Due to the lack of datasets with high number of clients, we decided to create another one (S3) synthetically, to test algorithms in a significantly larger instance. Dataset P1 [8] includes 20 requirements and 5 clients. Dataset P2 [14] includes 100 requirements and 5 clients. Dataset S3 includes 140 requirements and 100 clients. Each dataset contains a set of proposed requirements, defined by a vector of efforts, one effort value for each requirement. Clients are also included, defined by a vector of importances. The priority that each client gives to each requirement is also contained in the dataset, by means of a matrix of values in which each value represents the importance of a requirement for a client.

4.3 Methodology

We tested a set of hyperparameter configurations for each algorithm and dataset. Each configuration was executed 10 times.

For the Single-Objective GA and NSGA-II, populations were given values in the range 20 to 200 and number of generations took values among 100 to 2000. Crossover probabilities were assigned values among 0.6 to 0.9 and mutation probabilites in the range from 0 to 1. Both algorithms used a binary tournament selection, a one-point crossover scheme and an elitist replacement scheme. Both EDA approaches used population sizes and number of generations among 50 to 200. UMDA used two replacement schemes: a default one and an elitist replacement. PBIL used learning rates, mutation probabilities and mutation shifts with values between 0.1 to 0.9.

The stop criterion used by other works [3,14,15] is the number of function evaluations, commonly set to 10000. To adapt our experiments to this stop criterion, we restricted the execution of our algorithms to: $Pop.\ size * \#Gens. \leq 10000$.

We normalized datasets satisfaction and effort values, scaling them between 0 and 1. To evaluate the results, we compared the obtained Pareto fronts and a set of metrics. These metrics are quality indicators of the results generated by the algorithms and their efficiency: Hypervolume (HV), Δ-Spread, Spacing and execution time.

Mean values of these metrics have been calculated and compared pairwise between algorithms using a non-parametric statistical test, more specifically, the Mann-Whitney U test with Holm correction for multiple comparison. This is a non-parametric statistical hypothesis test that allows to assess whether one of two samples of independent observations tends to have larger values than the other. All the experiments were run in the same runtime environment. The machines used had 32 Gb RAM, of which only 8 Gb were used, and two 3.00 GHz 4-core Intel Xeon E5450 processors. The operating system used was a CentOS Linux 7 with a 64-bit architecture.

5 Results and Analysis

5.1 Best Configurations

The Single-Objective GA's best hyperparameter configuration includes a population size of 100 individuals, a number of generations of 100 (maximum number to stay under the 10,000 limit) and a $P_c = 0.8$. The mutation operator that showed a better performance was the *flip1bit*. This operator gives a chance of flipping only one bit of the booleans vector. The best-performing probability is $P_m = 1$, which means that we always mutates one random bit of each individual. That probability is equivalent to using $P_m = \frac{1}{n}$ at gene level, n being the number of genes (scheme used in [14]). The best hyperparameter configuration for the NSGA-II used a population size of 100 individuals and 100 generations. The best crossover probability (P_c) was the lowest, 0.6, and the best mutation operator was the *flip1bit*, using a $P_m = 1$.

Regarding UMDA, best hyperparameter configuration included a population size of 200, 50 generations and an elitist replacement scheme. With the upper limit of individual evaluations set and the datasets used, it seemed to generate better results when setting a higher population size than increasing the number of generations. Regarding the replacement scheme, elitism tends to produce wider Pareto fronts.

With respect to PBIL, a population size and number of generations of 100 was used, with learning and mutation rates of 0.5 and a mutation shift of 0.1. In this case, population size and number of generations did not show a significant difference in performance. However, learning rate and mutation configurations did affect the results. A higher learning rate than 0.5 caused the algorithm to underperform. Mutation worked similarly, enhancing results when increasing the probability up to a limit, and generating worse results with probability values above 0.5. For the mutation shift, high values showed bad performance, indicating that high variations in PBIL probability vector are not suiting this problem.

5.2 Pareto Front Results

To analyze Pareto front results, a random execution of the best algorithm configurations is plotted for each dataset (see Fig. 1). Single-Objective GA shows bad performance, being similar to that of the random procedure. This occurs due to the low number of generations set to keep the maximum number of function evaluations. Configuring a number of generations of one or two magnitude orders higher increases the quality of its Pareto front. Regarding the Pareto front distribution, the GA's aggregation of objectives biases the search, leaving unexplored areas. NSGA-II generates Pareto fronts of better quality: better solutions and more distributed along the search space. As expected, the crowding operator of the algorithm helps exploring the search space. However, as the dataset size increases, its performance decreases significantly. The reason is the limited number of generations, as this algorithm is expected to perform better in larger datasets when compared against other search methods. Regarding EDAs, for

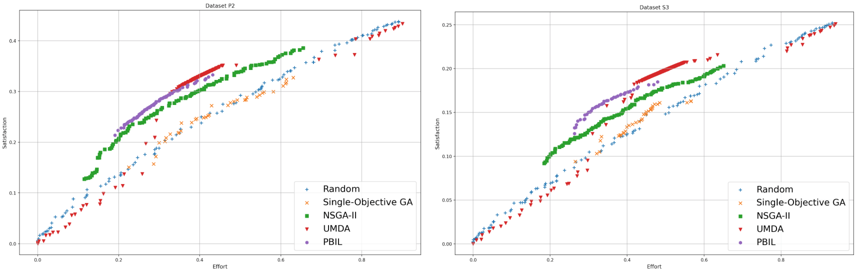


Fig. 1. Results from best configuration of each algorithm for datasets P2 (left) and S3 (right).

datasets P2 and S3 both algorithms struggled to generate a wide Pareto front. In the case of UMDA, it tends to group the best solutions in a certain area of the space, presumably where the probability vector drifted. In the remaining areas of the Pareto, only the random initial solutions are found, which were never updated by the algorithm. Regarding PBIL, as it does not generate an initial set of random solutions, but an initial vector, it did not create solutions in other areas of the space apart from the region were the probability vector was focused. It is interesting to highlight that both EDAs are unable to generate wide, high-quality Pareto fronts. Instead, both algorithms focused on a small region of the search space, progressively restricting the area with more consecutive generations, caused by the probability vector values stabilizing at extreme probabilities for 0 or 1 values. Despite this effect, it is worth mentioning that both EDAs have been able to overcome the Pareto front solutions returned by NSGA-II in the most balanced area of the heap.

5.3 Metrics Results

The mean values of the metrics obtained for each algorithm and dataset after 10 independent executions have been statistically compared, as explained in Sect. 4.3. Each metric mean value has been compared pair-wise between algorithms, denoting the best value in **bold** and indicating the values that are statistically worse ($P < 0.05$) with a ↓ symbol, as depicted in Table 1.

For the HV metric, NSGA-II obtained the best possible value for dataset P1, but UMDA generated best results for the other two datasets. This is caused by the initial random solutions generated. PBIL did not obtain a high HV metric for large datasets. For the Δ-Spread metric, best values were achieved by the Single-Objective GA for P1 and P2 datasets, but PBIL reached the lowest (best) Δ-Spread in dataset S3. Regarding Spacing, no EDA approach could surpass NSGA-II values, both algorithms obtaining similar results for each dataset. Finally, regarding execution time, EDA approaches resulted to run very fast, being PBIL the fastest algorithm for all datasets. These results show that EDA can lead to a very efficient and specific search through the Pareto front, being quicker than GAs to generate and evolve solutions, thanks to the simplified

Table 1. Average metrics of the best hyperparameter configurations for each algorithm and dataset

Dataset	Algorithm	HV	Δ-Spread	Spacing	Execution time (s)
P1	Single-Objective GA	0.594↓	**0.615**	0.323↓	17.967↓
	NSGA-II	**1.000**	0.963↓	**0.382**	180.991↓
	UMDA	0.878↓	0.744↓	0.330↓	5.494↓
	PBIL	0.554↓	0.692↓	0.316↓	**3.036**
P2	Single-Objective GA	0.157↓	**0.637**	0.128↓	82.713↓
	NSGA-II	0.407↓	0.969↓	**0.245**	616.415↓
	UMDA	**0.975**	1.008↓	0.111↓	21.489↓
	PBIL	0.138↓	0.670	0.114↓	**4.099**
S3	Single-Objective GA	0.102↓	0.720	0.105↓	125.702↓
	NSGA-II	0.286↓	0.970↓	**0.206**	859.928↓
	UMDA	**0.981**	1.025↓	0.103↓	23.079↓
	PBIL	0.089↓	**0.678**	0.105↓	**3.802**

methods they use, most of them applied only to the probability vector, instead of to the population.

6 Conclusions and Future Works

In this work, we have presented new algorithm proposals to the MONRP field, introducing the application of EDA techniques. Formerly, this problem has been commonly tackled by means of evolutionary algorithms, mainly GAs. This has been the first time that the EDA family of algorithms has been used to solve MONRP. For this purpose, we have considered the use of univariate EDAs as a first approach. In this paper we have adapted UMDA and PBIL algorithms and compared them against two GAs (Single-Objective GA and NSGA-II) by means of Pareto front results and quality metrics. Our proposals were able to overcome GAs in the most crowded regions of the Pareto front, despite generating low-quality or no solutions in other areas of the search space. Regarding quality metrics, our EDA proposals have been able to get rather good results compared to those of the GAs, showing the best performance in the execution time. Future research will focus on the applicability of multivariate EDA algorithms, aiming to manage the MONRP with modeled dependencies between requirements.

Acknowledgements. This work has been partially funded by the Regional Government (JCCM) and ERDF funds through the projects SBPLY/17/180501/000493 and SBPLY/21/180501/000148.

References

1. Alba, E., Ferrer, J., Villalobos, I.: Metaheuristics and software engineering: past, present, and future. Int. J. Software Eng. Knowl. Eng. **31**(09), 1349–1375 (2021)
2. Bagnall, A.J., Rayward-Smith, V.J., Whittley, I.M.: The next release problem. Inf. Softw. Technol. **43**(14), 883–890 (2001)
3. Chaves-González, J.M., Pérez-Toledano, M.A., Navasa, A.: Software requirement optimization using a multiobjective swarm intelligence evolutionary algorithm. Knowl.-Based Syst. **83**(1), 105–115 (2015)
4. Coello Coello, C.A., Lamont, G.B., Veldhuizen, D.A.V.: Evolutionary Algorithms for Solving Multi-Objective Problems. Springer, Heidelberg (2007). https://doi.org/10.1007/978-0-387-36797-2
5. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
6. Durillo, J.J., Zhang, Y., Alba, E., Harman, M., Nebro, A.J.: A study of the bi-objective next release problem. Empir. Softw. Eng. **16**(1), 29–60 (2011)
7. Finkelstein, A., Harman, M., Mansouri, S.A., Ren, J., Zhang, Y.: A search based approach to fairness analysis in requirement assignments to aid negotiation, mediation and decision making. Requirements Eng. **14**(4), 231–245 (2009)
8. Greer, D., Ruhe, G.: Software release planning: an evolutionary and iterative approach. Inf. Softw. Technol. **46**, 243–253 (2004)
9. Gupta, P., Arora, I., Saha, A.: A review of applications of search based software engineering techniques in last decade. In: 2016 5th International Conference on Reliability, Infocom Technologies and Optimization, ICRITO 2016: Trends and Future Directions, vol. 978, pp. 584–589 (2016)
10. Harman, M., McMinn, P., de Souza, J.T., Yoo, S.: Search based software engineering: techniques, taxonomy, tutorial. In: Meyer, B., Nordio, M. (eds.) Empirical Software Engineering and Verification: International Summer Schools, pp. 1–59. Springer, Berlin Heidelberg (2012). https://doi.org/10.1007/978-3-642-25231-0_1
11. Larrañaga, P., Lozano, J.A.: Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer Academic Publishers, New York (2001)
12. Ramírez, A., Delgado-Pérez, P., Ferrer, J., Romero, J.R., Medina-Bulo, I., Chicano, F.: A systematic literature review of the SBSE research community in Spain. Prog. Artif. Intell. **9**(2), 113–128 (2020)
13. Sagarna, R., Lozano, J.A.: On the performance of estimation of distribution algorithms applied to software testing. Appl. Artif. Intell. **19**(5), 457–489 (2005)
14. del Sagrado, J., del Águila, I.M., Orellana, F.J.: Multi-objective ant colony optimization for requirements selection. Empir. Softw. Eng. **20**(3), 577–610 (2013). https://doi.org/10.1007/s10664-013-9287-3
15. Zhang, Y., Harman, M., Mansouri, A.: The multi-objective next release problem. In: GECCO 2007: Genetic and Evolutionary Computation Conference, pp. 1129–1137 (2007)

An Extremal Optimization Approach to the Pairwise Connectivity Critical Node Detection Problem

Noémi Gaskó[1,2(✉)], Tamás Képes[1], Mihai Suciu[1,2], and Rodica Ioana Lung[2]

[1] Faculty of Mathematics and Computer Science, Babeş-Bolyai University,
Cluj, Romania
{noemi.gasko,tamas.kepes,mihai.suciu}@ubbcluj.ro

[2] Centre for the Study of Complexity, Babeş-Bolyai University, Cluj, Romania
rodica.lung@econ.ubbcluj.ro

Abstract. The critical node detection is a computational challenging problem with several applications in biology, sociology, etc. Minimizing the pairwise connectivity after removing k critical nodes is one of the most studied problem. In this paper we approach this problem by using a standard Extremal Optimization algorithm, and another variant with incorporated network shifting mechanism. Network centrality measures are used to speed up the search, the variants are analyzed on synthetic and real-world problems. Numerical results indicate the potential of the proposed approach.

1 Introduction

Complex networks have gained a lot of interest since random scale-free networks were introduced [5], having a wide range of applications. Network formation problem [27], the graph partitioning problem [9] or the influence maximization problem [11] are some of the problems that attempt to better explain complex networks. The critical node detection problem, a subclass of the node deletion problem [16], is an important problem in complex networks. When using different measures, the nodes of a network may have different importance. The problem of identifying important nodes in a network according to the network measure used can be a computationally challenging task, as the importance may vary with the measure used.

The critical node detection problem (CNDP) in a network is: find a set of k nodes, from the given network, that when deleted will maximally degrade the network according to a given measure $\sigma(G)$. Due to its large applicability, CNDP has gained popularity; such applications are in network immunization [13], network risk management [3], social network analysis [8], etc. In [14,17], the measure σ uses different network centrality measures (e.g. betweenness centrality, closeness centrality, page rank) to explore the node importance.

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 109–118, 2023.
https://doi.org/10.1007/978-3-031-18050-7_11

In this article we study the pairwise connectivity as a measure of node importance for the CNDP (presented in more detail in the next section). The main goal of the article is to propose an Extremal Optimization (EO) based approach to solve the problem. To variants of the EO algorithms are proposed, a simple one and a noisy based approach. We study four different variants for the selection processes (for both EO variants) based on different graph centrality measures.

Section 2 describes the problem studied and related work. The next section presents the proposed Extremal Optimization algorithm. The fourth section describes numerical experiments and comparisons with other algorithms. The last section presents conclusions and further work.

2 Related Work and Problem Formulation

Based on the detailed survey by Lalou et al. [15] the Critical Node detection problem can be classified in two main groups: k-vertex-CNDP and β-connectivity CNDP. k-vertex-CNDP is the classic variant of the problem presented above. In the case of β-connectivity CNDP the objective function needs to be bound to a value β, minimizing the number of deleted nodes.

The most studied version of the k-vertex-CNDP is the minimization of pairwise connectivity, called the Critical Node Problem. Given a graph $G = (V, E)$, where V denotes the set of nodes, E denotes the set of edges and a positive integer k, the problem consists of finding a subset of the vertices $S \subseteq V$ at most k nodes, such that the deletion of these nodes minimizes the pairwise connectivity in the remaining graph $G = (V \setminus S, E)$. Pairwise connectivity can be expressed as [15]:

$$f(S) = \sum_{C_i \in G[V \setminus S]} \frac{\delta_i(\delta_i - 1)}{2}, \tag{1}$$

where C_i, $i = 1 \ldots k$ are the connected components of the remaining graph $G = (V \setminus S, E)$, δ_i represents the size of the component C_i.

This variant is an NP-complete problem on general graphs [2]. As solving methods, an exact approach using Integer Linear Programming was proposed [2], in [25] a mixed integer programming formulation is described for several variants of the CNDP. As stochastic approaches, we mention a Greedy Randomized Adaptive Search Procedure [20], a Memetic Algorithm [29], and an evolutionary framework [1].

Another variant based on the pairwise connectivity is the bi-objective critical node detection problem, proposed in [24]. In [4] a cardinality-constrained variant is proposed. [26] proposed and formalizes the distance-based critical node detection problem.

3 Extremal Optimization

Extremal optimization (EO) [6,7] is an efficient optimization algorithm with multiple applications. For example, a variant of EO [18] was successfully used for the community detection problem, another variant was used to maximize connected components [12].

The standard variant of the EO algorithm uses two individuals: s and s_{best}; s_{best} preserves the best solution found so far by s based on an overall fitness $f()$. The component with the worst fitness is replaced randomly with a new generated solution (in the case of the CNDP problem, a new node from the network). The outline of the standard EO algorithm is presented in Algorithm 1. Algorithm 2 outlines the steps how the worst node is obtained.

To escape from local optima another variant of the EO is used, the NoisyEO [18]. It uses a network shifting mechanism to induce diversity in the search. This mechanism was used successfully for the maximization of connected components, Critical Nodes -EO (CN-EO) in [12].

The network is shifted, which means that edges are randomly deleted with a probability p_{shift}. The shifted network is used for a number of generations G. The outline of the algorithm is presented in Algorithm 3.

Algorithm 1. Extremal Optimization

```
Initialize $s, s_{best}$;
$gen := 0$;
while $gen < MaxGen$ do
    $worst$ =GetLeastCriticalBySubtraction($s$);
    switch $worst$ from $s$, with a node* from the network $G$
    if f($s$) > f($s_{best}$) then
        $s_{best} := s$
    end if
    $gen := gen + 1$
end while
```

* randomly selected, based on degree centrality, betweenness centrality, closeness centrality

Algorithm 2. GetLeastCriticalBySubtraction (s)

```
for all nodes in $s$ do
    $s' := s$
    remove the currently evaluated node from $s'$
    $result :=$ f($s$) − f($s'$)
    if $result < min$ or we are at the first node evaluated then
        $min := result$
    end if
end for
return the node with $result$ of $min$
```

Algorithm 3. *NEO* algorithm

```
Initialize $(s, s_{best})$;
gen:=0;
while $gen < MaxGen$ do
  if $s_{best}$ does not change in $G$ generations and there is no noise then
    Induce noise with probability $p_{shift}$;* ;
    Reinitialize $s_{best}$ with the current $s$ value
  end if
  if There has been noise for $G$ generations then
    Return to the original network
  end if
  Find the node with the worst fitness and replace it randomly** with another one;
  if $(f(s) > f(s_{best}))$ then
    set $s_{best} := s$.
  end if
  gen:=gen+1;
end while
Return $s_{best}$ with highest fitness achieved on a non-noisy network.
```

* Modify network by randomly deleting edges with probability p_{shift}

** randomly selected, based on degree centrality, betweenness centrality, closeness centrality

Encoding
Integer encoding is used: the individual s is represented as a vector of integers of size k (representing the critical nodes), each value is chosen in the interval $[1, n]$ where n is the number of nodes in the graph.

Fitness Function
Two fitness functions are used: the first evaluates the individual s and the second fitness function evaluates each individual component of s, i.e. each potential critical node. The fitness value of $s = (s_1, \ldots, s_k)$ is computed as the value of the pairwise connectivity of the remaining graph after removing the k nodes, C_i represents the remaining components and δ_i their size (as presented in Eq. 1):

$$f(s) = \sum_{C_i \in G[V \setminus (s_1,\ldots,s_k)]} \frac{\delta_i(\delta_i - 1)}{2} \qquad (2)$$

The second fitness function used for the evaluation of each component is computed as the marginal contribution of a node i in s to $f(s)$:

$$f_i(s) = f(s) - f(s \setminus i), \qquad (3)$$

where $s \setminus i$ denotes the set of nodes in s without node i.

Improvement of EO
In the standard version of EO the worst component (in our case the node) is randomly replaced. A natural way to improve this step is to guide the replacement to an "important" node. More central nodes (based on three network measures) will be chosen as new components, that is nodes which have their centrality above the average centrality value of the network will be chosen.

The following centrality measures are used:

- degree centrality - measures the degree of each node:
$$C_D(v) = \frac{deg(v)}{|V| - 1},$$
where $deg(v)$ is the degree of node v and $|V|$ is the number of nodes.
- closeness centrality - captures the average shortest distances from a node to all other nodes:
$$C_C(v) = \frac{1}{\sum_{w \neq v} d(w, v)},$$
where $d(w, v)$ denotes the shortest distance between node v and w.
- betweenness centrality - measures how a certain node lies on the shortest paths of other nodes:
$$C_B(v) = \sum_{s \neq v \neq t} \frac{g_{st}(v)}{g_{st}},$$
where $s, t \in V$, g_{st} is the total number of shortest paths, $g_{st}(v)$ is the number of shortest paths, where v lies.

For NEO the same centrality measures are used, but the selection of the new node is based on a p_{rand} probability: with a probability of p_{rand} the nodes with the centrality value over the average value of the network are selected, in the rest a random value is chosen.

4 Numerical Experiments

Parameter setting For each variant of the EO algorithm the maximum number of generations (iterations) is set to 5000. The algorithm does not have other parameters that need setting. In the case of the NEO MaxGen is set to 5000, G is set to 10, p_{shift} to 0.01, and p_{rand} to 0.8. The values of these parameters are based on a study about the Noisy Extremal Optimization approach [18].

Benchmarks. Table 1 describes basic network measures of the synthetic[1] and real-world[2] benchmarks used: number of nodes ($|V|$), number of edges ($|E|$), average degree ($\langle d \rangle$), density of the graph (ρ), and average path length (l_G).

[1] Downloaded from http://individual.utoronto.ca/mventresca/cnd.html.

[2] Downloaded from https://networkrepository.com/.

Table 1. Synthetic and real-world benchmark test graphs and basic properties.

Graph	$\|V\|$	$\|E\|$	$\langle d \rangle$	ρ	l_G	Ref.
BA500	500	499	1.996	0.004	5.663	[23]
BA1000	1000	999	1.998	0.002	6.045	[23]
ER250	235	350	2.979	0.013	5.338	[23]
ER500	466	700	3.004	0.006	5.973	[23]
FF250	250	514	4.112	0.017	4.816	[23]
FF500	500	828	3.312	0.007	6.026	[23]
Bovine	121	190	3.140	0.026	2.861	[21]
Circuit	252	399	3.167	0.012	5.806	[19]
EColi	328	456	2.780	0.008	4.834	[28]
USAir97	332	2126	12.807	0.038	2.738	[22]
TrainsRome	255	272	2.133	0.008	43.496	[10]

Comparisons with Other Methods. [1] presents three algorithms for the CNDP: two variants of a greedy algorithm (G_1 and G_2), one of them is based on node deletion and the other one is based on node addition and a third algorithm is a genetic algorithm from an evolutionary algorithm framework using greedy rules (GA). To the four variants of the EO algorithm, we refer them as: $R-EO$ the base algorithm, C_D-EO is the degree based approach, C_B-EO is the approach that uses betweenness and C_C-EO is the closeness based algorithm. Similarly, the four variants of the NEO algorithm are denoted by $R-NEO$, C_D-NEO, C_B-NEO, respectively C_C-NEO.

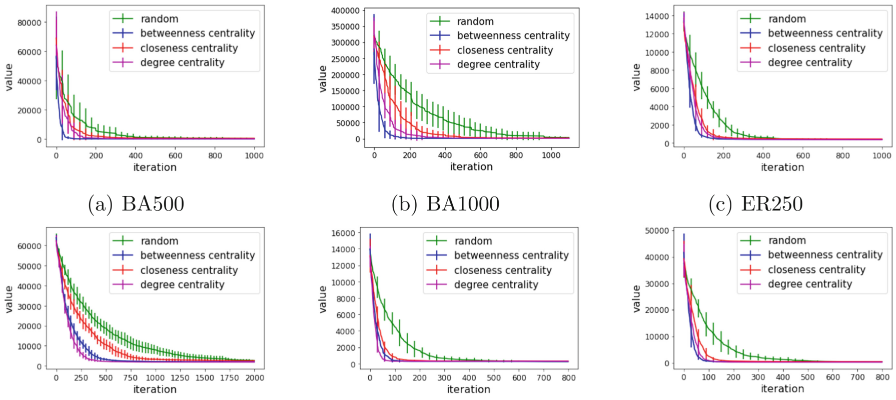
Figures 1 and 2 presents the evolution of the pairwise connectivity value (denoted in Oy axe value) over the number of generations (iterations) with the standard EO in the case of synthetic and real-world networks. In most cases the slowest evolution has the $R-EO$.

Table 2 presents the results obtained for the synthetic and real-world networks. The average value and standard deviation is reported for ten independent runs by each variant of the algorithm. The performance depends on the structure of the network, in most cases C_D-EO performs the best besides the standard $R-EO$. The noisy variant of the algorithm performs better for four networks, as the standard EO. In the case of the other three variants based on centrality measures in nine cases performs better (for one network all variants find the same value). The better performance is due to the noise and the introduced p_{rand} parameter, which ensures that with a small probability random nodes are selected, which ensures not to remain blocked in local optima. Critical nodes can be not only nodes with high centrality values.

Although we present best known results from the literature, the direct comparison with our methods is not fair as we report the averages and standard deviation over ten independent runs while the results reported by the literature are probably best obtained values (the authors of the study do not report average values, number of independent runs).

Table 2. Results for different real and synthetic networks (mean and standard deviation for ten independent runs) for the EO and NEO variants. The column "Best known results" presents the best results found in the literature according to [1], the authors do not specify if these values represent means or minimum values over multiple runs

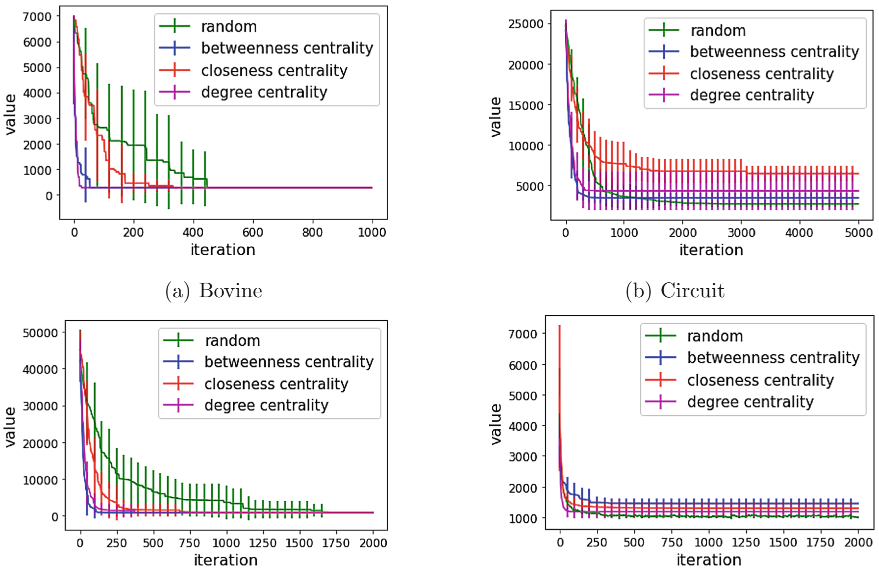
Network	k	$R - EO$	$C_B - EO$	$C_C - EO$	$C_D - EO$	Best known result
		$R - NEO$	$C_B - NEO$	$C_C - NEO$	$C_D - NEO$	
BA500	50	195.00(±0.00)	199.00(±0.00)	516.80(±26.84)	195.00(±0.00)	195 (GA)
		201.00(±3.07)	195.80(±.98)	211.40(±11.82)	195.00(±.00)	
BA1000	75	563.30(±8.69)	558.60(±0.70)	1395.50(±56.46)	558.70(±0.82)	558 (GA)
		617.20(±76.19)	558.50(±.50)	708.20(±56.20)	559.20(±.87)	
ER250	50	324.70(±17.98)	374.30(±32.10)	412.40(±45.51)	340.60(±30.23)	295 (GA)
		322.00(±13.98)	301.80(±4.12)	308.20(±8.45)	302.90(±10.72)	
ER500	80	2068.60(±325.52)	2032.10(±123.75)	2535.70(±301.49)	1877.30(±112.17)	1560 (GA)
		2218.67(±160.60)	1748.40(±75.01)	1845.70(±100.09)	1674.90(±44.42)	
FF250	50	197.60(±4.03)	248.40(±9.61)	286.40(±15.26)	273.80(±21.67)	194 (GA)
		199.50(±0.71)	195.00(±2.05)	195.50(±2.16)	196.00(±2.10)	
FF500	110	262.10(±4.12)	416.10(±28.77)	455.40(±30.74)	276.90(±5.09)	257 (GA)
		277.50(±12.77)	282.40(±7.34)	277.40(±6.36)	262.90(±1.64)	
Bovine	3	268.00(±0.00)	268.00(±0.00)	268.00(±0.00)	268.00(±0.00)	268 (G_1,G_2,GA)
		268.00(±.00)	268.00(±.00)	268.00(±.00)	268.00(±.00)	
Circuit	25	2730.20(±428.16)	3483.60(±664.97)	6461.10(±982.88)	4345.80(±2326.52)	2099 (G_1,GA)
		2317.00(±130.81)	2284.00(±111.93)	2388.40(±361.39)	2238.40(±97.01)	
EColi	15	840.40(±18.13)	830.30(±21.41)	919.60(±29.95)	828.80(±20.45)	806(G_1,GA)
		837.40(±16.74)	818.90(±19.71)	823.20(±21.07)	827.50(±21.50)	
TrainsR	26	1004.60(±71.30)	1440.80(±150.75)	1286.90(±206.00)	1176.20(±202.76)	921 (GA)
		937.20(±8.70)	964.80(±19.14)	948.70(±17.19)	943.60(±7.70)	



(a) BA500 (b) BA1000 (c) ER250

(d) ER500 (e) FF250 (f) FF500

Fig. 1. Errorbars of evolution of pairwise connectivity value on synthetic networks



(a) Bovine

(b) Circuit

(c) Ecoli

(d) Treni Roma

Fig. 2. Errorbars of evolution of pairwise connectivity value on real-world networks

As a general conclusion we can establish that using different centrality measures speeds up the convergence of the algorithm to the solutions, but it does not ensure better results in all cases. Another important conclusion is, that we cannot use only best nodes regarding a certain measure, critical nodes can appear between other nodes, as well. For example nodes with highest closeness centrality are not as good as randomly selected nodes.

5 Conclusions

Two variants of the Extremal Optimization algorithm are proposed to the Critical Node Problem (minimizing the pairwise connectivity). Besides the standard EO, in the NEO a network shifting mechanism is used to escape from local optima. The node replace is analyzed with several network centrality measures.

Comparisons with other methods are conducted both on synthetic and real world networks. Experiments demonstrate the potential of the proposed method. As further work other centrality measure will be investigated.

Acknowledgements. This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS - UEFISCDI, project number PN-III-P1-1.1-TE-2019-1633.

References

1. Aringhieri, R., Grosso, A., Hosteins, P., Scatamacchia, R.: A general evolutionary framework for different classes of critical node problems. Eng. Appl. Artif. Intell. **55**, 128–145 (2016)
2. Arulselvan, A., Commander, C.W., Elefteriadou, L., Pardalos, P.M.: Detecting critical nodes in sparse graphs. Comput. Oper. Res. **36**(7), 2193–2200 (2009)
3. Arulselvan, A., Commander, C.W., Pardalos, P.M., Shylo, O.: Managing network risk via critical node identification. Risk management in telecommunication networks. Springer (2007)
4. Arulselvan, A., Commander, C.W., Shylo, O., Pardalos, P.M.: Cardinality-constrained critical node detection problem. In: Gülpınar, N., Harrison, P., Rüstem, B. (eds.) Performance Models and Risk Management in Communications Systems. SOIA, vol. 46, pp. 79–91. Springer, New York (2011). https://doi.org/10.1007/978-1-4419-0534-5_4
5. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science **286**(5439), 509–512 (1999)
6. Boettcher, S., Percus, A.G.: Optimization with extremal dynamics. Phys. Rev. Lett. **86**, 5211–5214 (2001)
7. Boettcher, S., Percus, A.G.: Extremal optimization: an evolutionary local-search algorithm. In: Bhargava, H.K., Ye, N. (eds.) Computational Modeling and Problem Solving in the Networked World. Operations Research/Computer Science Interfaces Series, vol. 21, pp. 61–77. Springer, Heidelberg (2003). https://doi.org/10.1007/978-1-4615-1043-7_3
8. Borgatti, S.P.: Identifying sets of key players in a social network. Comput. Math. Organ. Theory **12**(1), 21–34 (2006)
9. Buluç, A., Meyerhenke, H., Safro, I., Sanders, P., Schulz, C.: Recent advances in graph partitioning. In: Kliemann, L., Sanders, P. (eds.) Algorithm Engineering. LNCS, vol. 9220, pp. 117–158. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49487-6_4
10. Cacchiani, V., Caprara, A., Toth, P.: Scheduling extra freight trains on railway networks. Transp. Res. Part B Methodol. **44**(2), 215–231 (2010)
11. Chen, W., Wang, Y., Yang, S.: Efficient influence maximization in social networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 199–208 (2009)
12. Gaskó, N., Képes, T., Suciu, M., Lung, R.I.: Critical node detection for maximization of connected components: an extremal optimization approach. In: Sanjurjo González, H., Pastor López, I., García Bringas, P., Quintián, H., Corchado, E. (eds.) SOCO 2021. AISC, vol. 1401, pp. 502–511. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-87869-6_48
13. He, J., Liang, H., Yuan, H.: Controlling infection by blocking nodes and links simultaneously. In: Chen, N., Elkind, E., Koutsoupias, E. (eds.) WINE 2011. LNCS, vol. 7090, pp. 206–217. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25510-6_18
14. Iyer, S., Killingback, T., Sundaram, B., Wang, Z.: Attack robustness and centrality of complex networks. PLoS ONE **8**(4), e59613 (2013)
15. Lalou, M., Tahraoui, M.A., Kheddouci, H.: The critical node detection problem in networks: a survey. Comput. Sci. Rev. **28**, 92–117 (2018)
16. Lewis, J.M., Yannakakis, M.: The node-deletion problem for hereditary properties is NP-complete. J. Comput. Syst. Sci. **20**(2), 219–230 (1980)

17. Lozano, M., García-Martínez, C., Rodriguez, F.J., Trujillo, H.M.: Optimizing network attacks by artificial bee colony. Inf. Sci. **377**, 30–50 (2017)
18. Lung, R.I., Suciu, M., Gaskó, N.: Noisy extremal optimization. Soft. Comput. **21**(5), 1253–1270 (2017)
19. Milo, R., et al.: Superfamilies of evolved and designed networks. Science **303**(5663), 1538–1542 (2004)
20. Purevsuren, D., Cui, G., Win, N.N.H., Wang, X.: Heuristic algorithm for identifying critical nodes in graphs. Adv. Comput. Sci. Int. J. **5**(3), 1–4 (2016)
21. Reimand, J., Tooming, L., Peterson, H., Adler, P., Vilo, J.: GraphWeb: mining heterogeneous biological networks for gene modules with functional significance. Nucleic Acids Res. **36**, 452–459 (2008)
22. Rossi, R.A., Ahmed, N.K.: The network data repository with interactive graph analytics and visualization. In: AAAI (2015)
23. Ventresca, M.: Global search algorithms using a combinatorial unranking-based problem representation for the critical node detection problem. Comput. Oper. Res. **39**(11), 2763–2775 (2012)
24. Ventresca, M., Harrison, K.R., Ombuki-Berman, B.M.: The bi-objective critical node detection problem. Eur. J. Oper. Res. **265**(3), 895–908 (2018)
25. Veremyev, A., Prokopyev, O.A., Pasiliao, E.L.: An integer programming framework for critical elements detection in graphs. J. Comb. Optim. **28**(1), 233–273 (2014)
26. Veremyev, A., Prokopyev, O.A., Pasiliao, E.L.: Critical nodes for distance-based connectivity and related problems in graphs. Networks **66**(3), 170–195 (2015)
27. Watts, A.: A dynamic model of network formation. Games Econom. Behav. **34**(2), 331–341 (2001)
28. Yang, R., Huang, L., Lai, Y.C.: Selectivity-based spreading dynamics on complex networks. Phys. Rev. E **78**(2), 026111 (2008)
29. Zhou, Y., Hao, J.K., Glover, F.: Memetic search for identifying critical nodes in sparse graphs. IEEE Trans. Cybern. **49**(10), 3699–3712 (2018)

Neural Networks and Data Mining

Dimensional Reduction Applied to an Intelligent Model for Boost Converter Switching Operation

Luis-Alfonso Fernandez-Serantes[1](✉), José-Luis Casteleiro-Roca[1], Paulo Novais[2], Dragan Simić[3], and José Luis Calvo-Rolle[1]

[1] University of A Coruña, CTC, CITIC, Department of Industrial Engineering, Ferrol, A Coruña, Spain
luis.alfonso.fernandez.serantes@udc.es

[2] Department of Informatics, University of Minho, Algoritmi Center, Braga, Portugal

[3] Faculty of Technical Sciences, University of Novi Sad, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia

Abstract. The dimensional reduction algorithms are applied to a hybrid intelligent model that distinguishes the switching operating mode of a boost converter. Thus, the boost converter has been analyzed and both operating mode are explained, distinguishing between Hard-switching and Soft-switching modes. Then, the dataset is created out of the data obtained from simulation of the real circuit and the hybrid intelligent classification model is implemented. Finally, the dimensional reduction of the input variables is carried out and the results are compared. As result, the proposed model with the applied dimensional reduced dataset is able to distinguish between the HS and SS operating modes with high accuracy.

Keywords: Hard-switching · Soft-switching · Boost converter · Power electronics · Classification · Dimensional reduction

1 Introduction

Nowadays, multiple research approaches are applied in the power electronics field, where the focus is kept on increasing the efficiency of the power converter; thus, reducing the size and weight of the circuits. The recent studies centre the attention on the use of the wide band-gap (WBG) materials such as Silicon Carbide (SiC) and Gallium Nitride (GaN) and the use of soft-switching techniques [1,4]. The introduction of SiC and GaN materials in the power converters initiated replacement of the silicon as manufacturing material of the power transistors [8,23]. They are more competitive, provide better performance and characteristics in comparison with silicon transistors, such as higher switching speeds, higher breakdown voltages, lower on-state resistance, etc. Additionally, in the last years the production prices have reduced, making them more interesting for the industry [8,15,17].

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 121–133, 2023.
https://doi.org/10.1007/978-3-031-18050-7_12

In order to improve the efficiency of the existing converters, the soft- switching techniques are widely introduced. These techniques allow a reduction of the switching losses during the converter operation. Moreover, the new materials, SiC and GaN, make more interesting these techniques in addition to their intrinsic characteristics [9,23].

Along with the introduction of the Artificial Intelligence (AI) in other fields, the AI starts gaining also importance in the power electronics. These techniques are used for supporting the development and design processes, as described in [2,26] where the AI are used to design magnetic components. Or, another application, to improve the performance of the power converters with the development of new control schemes, as done in [12,14,25].

With the aim of controlling and maintaining the converter operating in soft-switching and, therefore, delivering the maximum efficiency, the classification of the operating mode needs to be realized. When the converter operates in SS mode in comparison with HS mode, the switching losses are reduced. Thus, assuring that the converter operates in the desired mode becomes of importance.

In this work, the proposed method to detect the operating mode is based on AI. By measuring different signals of the converter, the AI is able to detect the operating mode, helping the designer to optimize the converter by reducing the switching losses and increasing the transfer of energy.

A dimensional reduction of the dataset used by the model is presented. This reduction of data helps to increase the speed and reduce computational cost of classifier, and further improve the performance of the model to detect the operating mode of the power converter.

The paper is structured as follows: first, an analysis of a synchronous rectified boost converter is explained in Sect. 2. The applied dimensional reduction to the proposed model is described in the next Sect. 3, along with the generated dataset and classification techniques. Then, the performance and efficiency of the proposed model with the different dimensional reduction methods is presented in Sect. 4 and finally, conclusions are drawn in Sect. 5.

2 Case Study

The analysis of a synchronous rectified boost converter is done in this section. The converter topology is shown in the Fig. 1. The components used in this converter are two transistors, high-side and low-side transistors, which operate in a complementary manner. The transistors generate a pulsed voltage that varies from input voltage and ground, at the switching node (Vsw), which is then filtered. An input capacitor is used to filter the peak currents drawn by the converter. The output filter is made up of an inductor and a capacitor, which filters the pulse voltage of the switching node obtaining a constant output voltage.

Traditionally, the described converter operates in Hard-Switching (HS) mode: meaning that losses occur during the switching transitions due to the current and voltage at the transistor during commutation. When the transistor is turned-off,

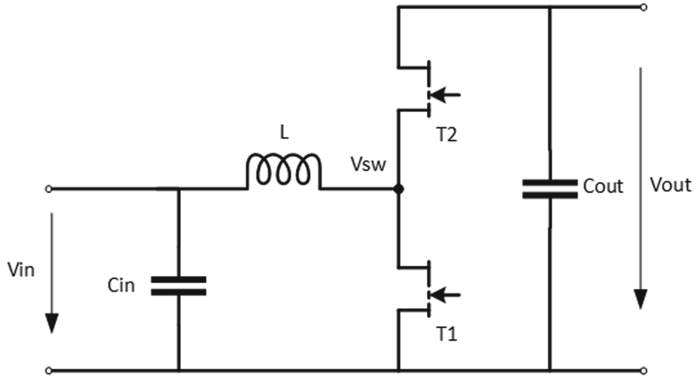


Fig. 1. Synchronous rectified boost converter.

the voltage is blocked and no current is flowing. When a signal is applied to the transistor's gate and the commutation starts, the resistance of the channel starts to drop as the current starts flowing through the transistor. During this time, the voltage drops while the current rises, occurring switching losses as $P = V \cdot I$. In HS mode, this process happens during turn-on and turn-off commutation. Moreover, when the transistor is switched on, the losses are caused by the on-state resistance times the flowing current: $P_{conduction} = I^2 \cdot R_{on-state}$.

In addition to the losses caused by the concurrent of current and voltage, the parasitic capacitance of the transistor (Coss) is reloaded and discharged through the transistor channel, causing further switching losses. These losses can be calculated as $P = \frac{1}{2} \cdot C_{oss} \cdot V^2$.

In the Fig. 2, the converter losses in HS mode are represented, where the switching losses can be seen.

Intending to improve the efficiency of the synchronous rectified boost converter, the other operating mode is introduced: soft-switching (SS) mode. In this case, either the current or the voltage through the transistor channels is brought to zero. If the current is zero at the switching instant, it is called Zero-Current-Switching (ZCS); on the other hand, if the voltage across the transistor at the switching instant is zero, the SS mode is due to Zero-Voltage-Switching (ZVS).

When the converter operates in SS, the losses during the commutation are nearly zero, as either the voltage or current are zero: $P = V \cdot I = 0 \cdot I$ or $P = V \cdot 0$. The Fig. 2, shows the transitions in the SS operating mode.

The conditions of SS are achieved thanks to the resonance between the components of the circuit, such as a resonance LC tank. Moreover, in the proposed converter, the resonance happens between the filter inductor and the parasitic output capacitance of the transistor (Coss), using this capacitance as a non-dissipative snubber [3,22].

In the synchronous rectified boost converter, the used method for the SS operation is the ZVS and it is achieved during turn-on of the high-side switch.

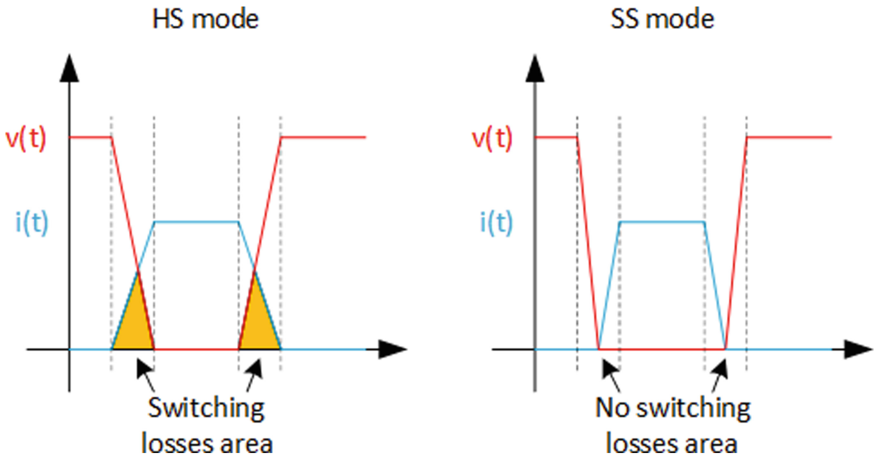


Fig. 2. Hard- and Soft-switching transitions with different current ripple.

In the ZVS method, the switches turn-on or -off when the voltage across the transistor is zero. Thus, when the high-side transistor switches off, the current moves to the low-side switch. During the interlock delay, time where both switches are off, the current flows through the antiparallel diode of the low-side transistor, equalizing the voltage across this transistor, causing a ZVS switching of the low-side transistor.

During the time that the low-side transistor is on, the current starts decreasing in the inductor until the current reaches a negative value, flowing from the load to the switches. At this instant, when the current is negative, the low-side switch turns-off. The current at the inductor does not have a path to flow, so it will charge the output capacitance of the transistors, rising the voltage at the switching node.

Once the voltage at the switching node reaches the input voltage, the antiparallel of the high-side switch starts conducting, instant when this switch can turn-on with ZVS.

At this point, the high-side transistor can switch lossless, as the voltage across is just a few volts from the forward diode voltage [19,22].

When using this topology in SS mode, the design of the inductor is very important, as it needs to allow high ripple current..

Traditionally, the designer and developers keep the inductor ripple low, between 10% and 30% of the output current. The definition of the inductance value is done according to the Eq. 1, where the inductor value depends on the switching frequency, output voltage and the allowed current ripple.

$$L = \frac{V_{in} \cdot (V_{out} - V_{in})}{f \cdot I_{ripple} \cdot V_{out}} \tag{1}$$

where L is the inductance value of the inductor, V_{in} is the input voltage to the circuit, V_{out} is the output voltage from the converter, f is the switching frequency and I_{ripple} is the current ripple in the inductor.

With the introduction of this converter operating in SS mode, the design of the inductor needs to be reconsidered. In this mode, the ripple of the current allows the current to drop to zero and beyond, defining the Triangular Current Mode (TCM) [5,7,13].

As mentioned above, when the current ripple is kept low, as shown in the Fig. 3, the converter operates in HS mode, as the ZVS condition is never reached. In opposition, when the current ripple allows the current to drop below zero, the switching losses in the converter can be reduced due to the ZVS mode.

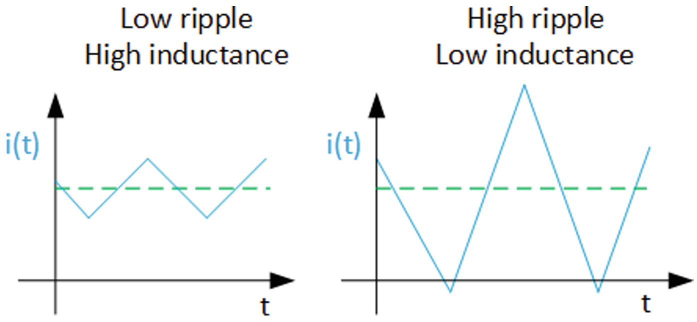


Fig. 3. Current ripple with different filter inductors.

The SS operating mode allows a reduction of the switching losses but with the drawback of increasing the conduction losses, as the Root Means Square (RMS) of the current increases. In order to take advantage of this operating mode, the switching frequency of the converter is increased, reducing in this manner the conduction losses and the filter components; therefore further increasing the power densities of the converter.

3 Model Approach

In this research, a dimensional reduction of the data used by the classification model has been done. The classification model aims to detect and distinguish the operating mode of the converter, between HS and SS mode. Three different dimensional reduction techniques are applied with the aim of reducing the computational cost of the classifier and improving the performance and efficiency.

In the Fig. 4, the steps followed to build the classification model are shown. First, the simulation data is obtained and pre-processed to obtain more significant parameters which provide detail information about the converter operating mode. Then, the reduction of the data is applied and finally the classification algorithms are used to determine if the converter is in either HS or SS mode.

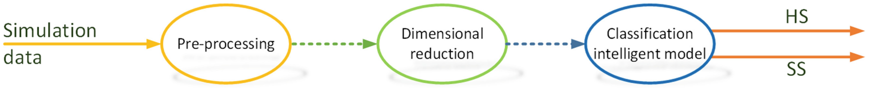


Fig. 4. Model approach.

3.1 Dataset

The dataset used in this work is obtained from simulation results of the synchronous rectified boost converter by using the LTSpice simulation software. With the aim of obtaining closer results to the real circuit, the transistors models from the manufacturer have been used.

Up to 80 simulations have been done with the proposed circuit of the Fig. 1. The converter is operated in both HS and SS modes. To allow reproducibility of the experiments and results, the converter keeps unchanged and just the applied load is varied. The complete dataset is compound of 50% of HS data and the other 50% of SS data.

The dataset is made up from the following signals measured in the circuit:

- Input voltage: a constant input voltage of 200 V is applied to the circuit.
- Output voltage: the output voltage of the converter is kept at 400 V, allowing a ripple from 390 V up to 410 V.
- Switching node voltage (Vsw node Fig. 1): at this node, the voltage varies from 0 V when the low-side switch is on up to 400 V when the high-side switch is on. The generated signal is square with a frequency varying from 80 kHz up to 2 MHz.
- Inductor current: is a triangular waveform. The average current depends on the output load and the current ripple depends on the switching frequency. For a constant inductance value, when the switching frequency is higher, the current ripple is lower, while it increases as the switching frequency decreases. In HS mode, the ripple is between 10% to 30% while in SS mode, the ripple increases above 200%.
- Output current: is constant, filtered by the inductor and capacitor, and its value depends on the connected load to the converter.

Then, from this initial dataset, a pre-processing is done for the purpose of obtaining more significant measurements: the base for this pre-processing is the raw data of the switching node voltage (Vsw); then, the first and second derivative are calculated, removing the on- and off-states while keeping the information of the commutations.

Furthermore, the rising and falling edges of the Vsw are isolated, as shown in the Fig. 5. This allows to focus the model on the transitions, which provides details of the operating mode, either HS or SS mode. The rising and falling times are also obtained (tr and tf).

Moreover, the first and second derivatives are also applied to the rising and falling edge signal explained above. Additionally, the integral of the edges is calculated.

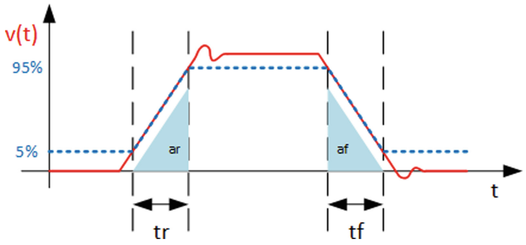


Fig. 5. Rising and falling edge of the switching node voltage, in dashed blue, and the original signal in continuous red. (Color figure online)

As described, 8 signals are obtained from the Vsw: the raw data (red signal in Fig. 5), the first and the second derivatives of the raw data, the rising/falling edge data (dotted blue signal in Fig. 5), the first and second derivatives of rising/falling edge data, the rising edge integral (area at the rising edge, *ar*, in Fig. 5) and the falling edge integral (area at the falling edge, *af*, in Fig. 5).

Moreover, the following statistics and indicators are calculated from the 8 obtained signals: average, standard deviation, variance, co-variance, Root Mean Square and Total Harmonic Distortion (THD). Resulting in a matrix of 8×6 for each of the 80 simulations.

Moreover, previous application of the dimensional reduction algorithms, the data has been parameterized.

3.2 Methods

The dimensional reduction algorithms used in this research are the self-organizing map (SOM), the correlation matrix and the principal component analysis (PCA).

3.2.1 Self-organizing Map (SOM)

Self-organizing maps learn to cluster data based on similarity, topology, with a preference of assigning the same number of instances to each class. Self-organizing maps are used both to cluster data and to reduce the dimensionality of data [24].

In SOM method, the dimensional reduction is completed by visualization techniques. The procedure is executing the SOM algorithm and with the obtained map results for each variable and taking into account the similarity, the reduction is accomplished by discarding the less convenient variable.

3.2.2 Correlation Matrix

A correlation matrix is a table showing correlation coefficients between variables. Each cell in the table shows the correlation between two variables. A correlation matrix is used to summarize data, as an input into a more advanced analysis, and as a diagnostic for advanced analyses [18].

3.2.3 Principal Component Analysis (PCA)

Principal component analysis (PCA) is one of the classical dimensionality reduction algorithms, which guarantees the minimum mean square error and gains linearly independent vectors as the basis of subspace [18].

3.3 Classification Model

3.3.1 Multilayer Perceptron

A multilayer perceptron is an artificial neural network with multiple hidden layers of neurons. The structure is the following: one input layer, with the input features to the algorithm, then the multiple hidden layer which have neurons with an activation function, and one output layer, which number of neurons depends on the desired outputs. All these layers are connected in between by weighted connections that are tuned with the aim of decreasing the error of the output [7,11,25].

3.3.2 Linear Discrimination Analysis

This method projects the data from a high dimensional space into a low-dimensional space. This method uses a weight vector W, which projects the given set of data vector E in such a way that maximizes the class separation of the data but minimizes the intra-class data [10,27]. The separation is good when the projections of the class involve exposing long distance along the direction of vector W [6,21].

3.3.3 Support Vector Machine

The algorithm tries to find two parallel hyper-planes that maximize the minimum distance between two class of samples [20]. Therefore, the vectors are defined as training instances presented near the hyperplane and the projection of the dataset is done in a high dimensional feature space using a kernel operator [20].

3.3.4 Ensemble

The term ensemble is used to define multiple classification methods which are used in combination with the aim of improving the performance over single classifiers [16]. They are commonly used for classification tasks. The ensemble does a regularization, process of choosing fewer weak learners in order to increase predictive performance [28].

3.4 Experiments Description

The experiments carried out aims to show the performance of the classification algorithms and to validate the improvement achieved by the dimensional reduction.

First, the dataset is divided into two different groups. The first group, that contains 75% of the generated data, is used to train the proposed models; while the rest, 25 % of the dataset, is used to validate the proposed algorithms. It is important to remark that the separation is done randomly.

The following algorithms are trained and then validated with the previously mentioned datasets:

- Multilayer perceptron (MLP): uses the Levenberg-Marquardt backpropagation algorithm with an hidden layer with 1 up to 10 neurons.
- Linear discrimination analysis (LDA): the used type is the regularized LDA, where all the classes have identical co-variance matrix.
- Support vector machine (SVM): The linear kernel function has been used. The classifier has been trained using the standardized predictors, which centers and scales each predictor variable by the corresponding mean and standard deviation.
- Ensemble: the adaptive logistic regression has been used. This type is commonly applied to binary classification. The number of cycles has been varied in steps of 10 from 10 up to 100. The weak-learners used function is the decision tree.

Finally, once the different models are trained, the models are validated using the previously separated 25% of the data. This is done to verify the correct functionality and performance of the proposed models. The classification results obtained from the models are then compared with the validation data and by using a confusion matrix, the different statistics are calculated.

In a following step, the proposed dimensional reduction methods are applied to the dataset. The different models are used again with this reduced data and the performance is measured. The dimensional reduction methods used are:

- SOM: Train with size of 20 two-dimensional map.
- Correlation matrix: the values close to 1 or −1 indicate that there is a linear relationship between the data columns and they can be removed, leaving the values that are equal or close to 0, that suggest that these data are not related. The chosen limit to differentiate the variables is 0.33.
- PCA: a new matrix is created with different weights of the data. The data in this matrix has no relation with the input data.

4 Results

The Table 1 shows the classification performance of the model approach by applying different dimensional reduction methods. When no reduction is used, the best achieved performance was by the MLP7 with 97.06% of accuracy while when using the dimensional reduction methods, the accuracy increases up to 100% of right classification.

The obtained data reduction by the different applied techniques is the following:

- SOM: a reduction of 33 variables has been achieved, meaning that 70% of the data is removed.

- Correlation matrix: in this case, 37 variables are discarded, reducing the size of the data matrix by 78%.
- PCA: a new matrix of data has been created with different weights of the parameters.

As summary, the performance and accuracy in percentage that each classification technique achieves with the dimensional reduction of the input data is shown in the Table 1.

Table 1. Summary of the results.

Classification method	Dimensional reduction methods			
	Without reduction	SOM	Corr. Matrix	PCA
MLP1	81.54	89.47	91.89	28.57
MLP2	73.53	94.74	94.59	36.36
MLP3	62.26	94.74	94.59	27.78
MLP4	55.88	100	100	42.10
MLP5	79.41	86.11	94.60	35.00
MLP6	60.29	92.11	92.15	52.94
MLP7	97.06	100	97.30	23.53
MLP8	73.53	92.11	94.59	42.10
MLP9	86.77	97.37	100	38.89
MLP10	52.94	100	97.30	21.74
SVM	77.94	94.74	91.89	39.29
LDA	60.29	86.84	94.59	46.43
Ensemble10	51.47	97.37	89.20	96.43
Ensemble20–100	51.47	94.74	86.48	96.43

5 Conclusions and Future Works

In this research different dimensional reduction methods are applied to a classification model used to detect the operating mode of a synchronous rectified boost converter with the aim of increasing the classification accuracy and reducing the computational cost.

The used dataset is obtained from the simulation of the synchronous rectified boost converter and the most significant variables are selected to perform the classification.

In order to compare the performance of the dimensional reduction, three different algorithms have been applied: SOM, correlation matrix and PCA. The achieved dimensional reduction of the data is 70% when using SOM and 78% when using the correlation matrix. In case of using PCA, a new matrix has been built.

The results achieved by the classification models are compared, when no dimensional reduction model is used and when the proposed methods are applied. When SOM algorithm is used, the performance is increase up to 100% with the MLP4, MLP5 and MLP10 classifiers. Moreover, the correlation matrix increases the accuracy of the MLP4 and MLP9 classifiers up to 100%. Although the accuracy with the PCA method applied to the models does not achieve the 100%, the performance of the Ensemble classifiers is increased from 51% up to 96%.

In addition to the increase of the accuracy, the dimensional reduction reduces the input dataset by 70% in case of using SOM and by 78% while using the correlation matrix.

The research in this field will follow by the development of a hybrid intelligent model that aims to further improve the accuracy of the classifiers. Moreover, the real circuit will be implemented and the models will be applied to real measured data from the power converter.

Acknowledgements. CITIC, as a Research Center of the University System of Galicia, is funded by Consellería de Educación, Universidade e Formación Profesional of the Xunta de Galicia through the European Regional Development Fund (ERDF) and the Secretaría Xeral de Universidades (Ref. ED431G 2019/01).

References

1. Al-bayati, A.M.S., Alharbi, S.S., Alharbi, S.S., Matin, M.: A comparative design and performance study of a non-isolated DC-DC buck converter based on Si-MOSFET/Si-diode, SiC-JFET/SiC-Schottky diode, and GaN-transistor/SiC-Schottky diode power devices. In: 2017 North American Power Symposium (NAPS), pp. 1–6 (2017). https://doi.org/10.1109/NAPS.2017.8107192
2. Aláiz-MoretónH, H., et al.: A fault detection system for a geothermal heat exchanger sensor based on intelligent techniques. Sensors **19**(12), 2740 (2019)
3. Casado-Vara, R., et al.: Edge computing and adaptive fault-tolerant tracking control algorithm for smart buildings: a case study. Cybern. Syst. **51**(7), 685–697 (2020). https://doi.org/10.1080/01969722.2020.1798643
4. Casteleiro-Roca, J.L., Javier Barragan, A., Segura, F., Luis Calvo-Rolle, J., Manuel Andujar, J.: Intelligent hybrid system for the prediction of the voltage-current characteristic curve of a hydrogen-based fuel cell. Revista Iberoamericana de Automática e Informática industrial **16**(4), 492–501 (2019)
5. Fernandez-Serantes, L.A., Berger, H., Stocksreiter, W., Weis, G.: Ultra-high frequent switching with GaN-HEMTs using the cross-capacitances as non-dissipative snubbers. In: PCIM Europe 2016; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management, pp. 1–8. VDE (2016)
6. Fernandez-Serantes, L.A., Casteleiro-Roca, J.L., Berger, H., Calvo-Rolle, J.L.: Hybrid intelligent system for a synchronous rectifier converter control and soft switching ensurement. Eng. Sci. Technol. Int. J. 101189 (2022)
7. Fernandez-Serantes, L.A., Casteleiro-Roca, J.L., Calvo-Rolle, J.L.: Hybrid intelligent system for a half-bridge converter control and soft switching ensurement. Revista Iberoamericana de Automática e Informática industrial (2022). https://doi.org/10.4995/riai.2022.16656

8. Fernández-Serantes, L.A., Estrada Vázquez, R., Casteleiro-Roca, J.L., Calvo-Rolle, J.L., Corchado, E.: Hybrid intelligent model to predict the SOC of a LFP power cell type. In: Polycarpou, M., de Carvalho, A.C.P.L.F., Pan, J.-S., Woźniak, M., Quintian, H., Corchado, E. (eds.) HAIS 2014. LNCS (LNAI), vol. 8480, pp. 561–572. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07617-1_49
9. García-Ordás, M.T., et al.: Clustering techniques selection for a hybrid regression model: a case study based on a solar thermal system. Cybern. Syst. 1–20 (2022). https://doi.org/10.1080/01969722.2022.2030006
10. Gonzalez-Cava, J.M., et al.: Machine learning techniques for computer-based decision systems in the operating theatre: application to analgesia delivery. Log. J. IGPL. **29**(2), 236–250 (2020). https://doi.org/10.1093/jigpal/jzaa049
11. Jove, E., Casteleiro-Roca, J., Quintián, H., Méndez-Pérez, J., Calvo-Rolle, J.: Anomaly detection based on intelligent techniques over a bicomponent production plant used on wind generator blades manufacturing. Revista Iberoamericana de Automática e Informática industrial **17**(1), 84–93 (2020)
12. Jove, E., et al.: Comparative study of one-class based anomaly detection techniques for a bicomponent mixing machine monitoring. Cybern. Syst. **51**(7), 649–667 (2020). https://doi.org/10.1080/01969722.2020.1798641
13. Jove, E., Casteleiro-Roca, J.L., Quintián, H., Méndez-Pérez, J.A., Calvo-Rolle, J.L.: A fault detection system based on unsupervised techniques for industrial control loops. Expert. Syst. **36**(4), e12395 (2019)
14. Jove, E., Casteleiro-Roca, J.L., Quintián, H., Simić, D., Méndez-Pérez, J.A., Luis Calvo-Rolle, J.: Anomaly detection based on one-class intelligent techniques over a control level plant. Log. J. IGPL **28**(4), 502–518 (2020)
15. Jove, E., Casteleiro-Roca, J.L., Quintián, H., Méndez-Pérez, J.A., Calvo-Rolle, J.L.: A new method for anomaly detection based on non-convex boundaries with random two-dimensional projections. Inf. Fusion. **65**, 50–57 (2021). https://doi.org/10.1016/j.inffus.2020.08.011, https://www.sciencedirect.com/science/article/pii/S1566253520303407
16. Jove, E., et al.: Modelling the hypnotic patient response in general anesthesia using intelligent models. Log. J. IGPL **27**(2), 189–201 (2019)
17. Jove, E., et al.: Hybrid intelligent model to predict the remifentanil infusion rate in patients under general anesthesia. Log. J. IGPL. **29**(2), 193–206 (2020). https://doi.org/10.1093/jigpal/jzaa046
18. Kaski, S., Sinkkonen, J., Klami, A.: Discriminative clustering. Neurocomputing **69**(1–3), 18–41 (2005)
19. Leira, A., et al.: One-class-based intelligent classifier for detecting anomalous situations during the anesthetic process. Log. J. IGPL (2020). https://doi.org/10.1093/jigpal/jzaa065
20. Liu, M.Z., Shao, Y.H., Li, C.N., Chen, W.J.: Smooth pinball loss nonparallel support vector machine for robust classification. Appl. Soft Comput. **98**, 106840 (2020). https://doi.org/10.1016/j.asoc.2020.106840
21. Marchesan, G., Muraro, M., Cardoso, G., Mariotto, L., da Silva, C.: Method for distributed generation anti-islanding protection based on singular value decomposition and linear discrimination analysis. Elect. Power Syst. Res. **130**, 124–131 (2016). https://doi.org/10.1016/j.epsr.2015.08.025
22. Mohan, N., Undeland, T.M., Robbins, W.P.: Power Electronics: Converters, Applications, and Design. John Wiley & Sons, Hoboken (2003)
23. Neumayr, D., Bortis, D., Kolar, J.W.: The essence of the little box challenge-part a: key design challenges solutions. CPSS Trans. Power Electron. App. **5**(2), 158–179 (2020). https://doi.org/10.24295/CPSSTPEA.2020.00014

24. Qin, A.K., Suganthan, P.N.: Enhanced neural gas network for prototype-based clustering. Pattern Recogn. **38**(8), 1275–1288 (2005)
25. Tahiliani, S., Sreeni, S., Moorthy, C.B.: A multilayer perceptron approach to track maximum power in wind power generation systems. In: TENCON 2019–2019 IEEE Region 10 Conference (TENCON), pp. 587–591 (2019). https://doi.org/10.1109/TENCON.2019.8929414
26. Liu, T., Zhang, W., Yu, Z.: Modeling of spiral inductors using artificial neural network. In: Proceedings of 2005 IEEE International Joint Conference on Neural Networks, 2005, vol. 4, pp. 2353–2358 (2005). https://doi.org/10.1109/IJCNN.2005.1556269
27. Thapngam, T., Yu, S., Zhou, W.: DDOS discrimination by linear discriminant analysis (LDA). In: 2012 International Conference on Computing, Networking and Communications (ICNC), pp. 532–536. IEEE (2012)
28. Uysal, I., Gövenir, H.A.: An overview of regression techniques for knowledge discovery. Knowl. Eng. Rev. **14**, 319–340 (1999)

Intuitionistic Fuzzy Sets in J-CO-QL$^+$?

Paolo Fosci and Giuseppe Psaila(✉)

University of Bergamo, Viale Marconi 5, 24044 Dalmine (BG), Italy
{paolo.fosci,giuseppe.psaila}@unibg.it
http://www.unibg.it

Abstract. Intuitionistic Fuzzy Sets extend the classical notion of Fuzzy Sets, so as to represent "hesitation": indeed, an item has both a membership degree and a non-membership degree, whose sum could be less than 1; the difference denotes the "hesitation" about the fact that the item belongs or not to the fuzzy set. Similarly, Intuitionistic Fuzzy Relations involve two domains.

Supposing that Intuitionistic Fuzzy Sets and Relations are provided as JSON data sets, is there a stand-alone tool to process them? This paper studies if the constructs currently provided by *J-CO-QL*$^+$ (the query language of the *J-CO* Framework) for managing fuzzy sets can actually deal with Intuitionistic Fuzzy Sets and Relations. The results will suggest how to extend *J-CO-QL*$^+$ to deal with classical and Intuitionistic Fuzzy Sets in an integrated way.

Keywords: Intuitionistic Fuzzy Sets and relations · Soft Querying on JSON data sets · Managing Intuitionistic Fuzzy Sets in J-CO-QL$^+$

1 Introduction

An Intuitionistic Fuzzy Set (IFS) is an extension of the classical notion of Fuzzy Set (FS) proposed by Zadeh [16], so as to represent "hesitation": an item of an IFS has both a membership degree and a non-membership degree, whose sum could be less than 1; the difference denotes the "hesitation" about the fact that the item belongs or not to the fuzzy set. The classical notion of relation can be generalized as Intuitionistic Fuzzy Relation (IFR) as well.

Supposing that *JSON* data sets are provided as IFSs and IFRs, is there a stand-alone tool to process them? At University of Bergamo (Italy), we are developing the *J-CO* Framework: it is a pool of software tools whose goal is to allow analysts to integrate and query *JSON* data sets, possibly stored within *JSON* document stores, like *MongoDB* (the framework was inspired by our previous work [5]); The *J-CO-QL*$^+$ query language supports the evaluation of membership degrees of documents to fuzzy sets (this idea was inspired by our previous work [6]). In this paper, we study if and how *J-CO-QL*$^+$ is currently able to manage IFSs and IFRs. The outcomes of this study will suggest how to extend (as a future work) *J-CO-QL*$^+$ to deal with FSs and IFSs in an integrated way.

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 134–145, 2023.
https://doi.org/10.1007/978-3-031-18050-7_13

The goal of this paper is not to discuss if IFSs are better than FSs or possibility theory [9] (this discussion is outside the scope of the paper); in contrast, it evaluates how far away *J-CO-QL*$^+$ is from providing a full support to IFSs.

The paper is organized as follows. Section 2 briefly introduces FSs (Sect. 2.1), basic concepts on IFSs and IFRs (Sect. 2.2) and a simple case study (Sect. 2.3). Section 3 studies if and how it is possible to deal with IFSs and IFRs within *J-CO-QL*$^+$. Finally, Sect. 4 discusses the lessons we learned towards a further extension of *J-CO-QL*$^+$ that addresses multiple models of FSs at the same time; it also draws the conclusions.

2 Background

In this section, we briefly present the background of our work, i.e., basic notions on FSs (Sect. 2.1) and on IFSs and IFRs (Sect. 2.2), practically explained (in Sect. 2.3) through an example.

2.1 Classical Fuzzy Sets

First of all, we report basic notions about classical Fuzzy Sets [16].

Let us denote by U a non-empty universe, either finite or infinite.

Definition 1. A fuzzy set $A \subseteq U$ is a mapping $A : U \rightarrow [0, 1]$. The value $A(x)$ is referred to as the *membership degree* of the x item to the A fuzzy set.

Therefore, a FS A in U is characterized by a membership function $A(x)$ that associates each item $x \in U$ with a real number in the interval $[0, 1]$; in other words, given x, the value $A(x)$ is the degree with which x is member of A (alternatively, the notation $\mu_A(x)$ is used, in place of $A(x)$). Thus, $A(x) = 1$ means that x fully belongs to A; $A(x) = 0$ means that x does not belong to A; $0 < A(x) < 1$ means that x belongs to A, but only partially. Consequently, FSs express vague or imprecise concepts on real-world entities, as well as they express vague or imprecise relationships among real-world entities.

Definition 2. Consider a universe U and two fuzzy sets A and B in U. The *union* of A and B is $S = A \cup B$, where $S(x) = max(A(x), B(x))$ (alternatively written as $\mu_S(x) = max(\mu_A(x), \mu_B(x))$). The *intersection* of A and B is $I = A \cap B$, where $I(x) = min(A(x), B(x))$ (alternatively written as $\mu_I(x) = min(\mu_A(x), \mu_B(x))$).

In this paper, U is the universe set of *JSON* documents. Thus, given a document $d \in U$, the membership degree $A(d)$ denotes the extent with which d belongs to the A FS (d represents a real-world entity or a relation).

2.2 Intuitionistic Fuzzy Sets and Relations

We consider the extension of FSs called Intuitionistic Fuzzy Sets [1]. Classical FSs relies on the "Axiom for Excluded Middle" [2]: given an item $x \in U$ and a FS A, given $A(x)$ its degree of membership, the degree of "non membership" to A is necessarily $1 - A(x)$; so, introducing a function $\nu_A(x)$ that explicitly represents the non-membership degree, $\nu_A(x) = 1 - \mu_A(x)$. But this assumption does not cope with the idea that gray zones can characterize practical situations: is a given symptom s_1 shown by a patient a clear evidence that the patient has a specific disease d_3? A physician would "feel" this, based on past experience and knowledge, that is, the physician is confident that the s_1 symptom denotes the d_3 disease with strength $\mu_{d_3}(s_1)$, but the strength $\nu_{d_3}(s_1)$ of the feeling that s_1 does not denote d_3 is such $\nu_{d_3}(s_1) < 1 - \mu_{d_3}(s_1)$, meaning that the physician "hesitates" in making a decision. In other words, the physician relies on intuitions, which are characterized by a degree of hesitation between the two opposite cases.

Definition 3. Given a universe U, an Intuitionistic Fuzzy Set (IFS) A in U is defined as a function $A : U \rightarrow [0,1]_\mu \times [0,1]_\nu$, where $[0,1]_\mu$ is the domain of membership degrees and $[0,1]_\nu$ is the domain of non-membership degrees. Thus, given an item $x \in U$, $A(x) = \langle \mu, \nu \rangle$, where μ and ν are, respectively, the membership degree of x to A (written also as $\mu_A(x) \in [0,1]$) and the non-membership degree of x to A (written also as $\nu_A(x) \in [0,1]$). For all items $x \in U$ and an Intuitionistic Fuzzy Set A in U, the following constraint must be satisfied: $A(x).\mu + A(x).\nu \leq 1$ (alternatively written as $\mu_A(x) + \nu_A(x) \leq 1$). Finally, given an item $x \in U$ and an IFS A in U, $\pi_A(x) = 1 - (A(x).\mu + A(x).\nu)$ is called the *hesitation* of x.

Given an IFS A in U, if for all items $x \in U$ it is $A(x).\mu + A(x).\nu = 1$, then A is a classical FS. Thus, classical FSs are included in the domain of IFSs.

Classical set-oriented operations can be adapted for IFSs.

Definition 4. Consider a universe U and two IFSs A and B in U.
The *union* of A and B is $S = A \cup B$, where
$S(x) = \langle max(A(x).\mu, B(x).\mu), min(A(x).\nu, B(x).\nu) \rangle$
(alternatively, it is possible to write $\mu_S(x) = max(\mu_A(x), \mu_B(x))$ and $\nu_S(x) = min(\nu_A(x), \nu_B(x))$).
The *intersection* of A and B is $I = A \cap B$, where
$I(x) = \langle min(A(x).\mu, B(x).\mu), max(A(x).\nu, B(x).\nu) \rangle$
(alternatively, it is possible to write $\mu_S(x) = min(\mu_A(x), \mu_B(x))$ and $\nu_S(x) = max(\nu_A(x), \nu_B(x))$).

As the reader can see, IFSs are extensions of FSs.

Intuitionistic Fuzzy Relations. The notion of Intuitionistic Fuzzy Relation [3,8] was used to model medical knowledge in a decision-making application (see more in Sect. 2.3).

Definition 5. Consider two sets X and Y. An *Intuitionistic Fuzzy Relation* (IFR) R from X to Y is an IFS in $U = X \times Y$ ($R : X \times Y \rightarrow [0,1]_\mu \times [0,1]_\nu$), thus $R(\langle x, y\rangle) = \langle \mu, \nu\rangle$ (alternatively, $\mu_R(\langle x, y\rangle)$ and $\nu_R(\langle x, y\rangle)$).

In the literature, the following notation is often used: $R(X \rightarrow Y)$.

What happens if an IFS A is defined on X? It is reasonable to "compose" A and R, so as to obtain a derived IFS B on Y.

Definition 6. Consider an IFR $R(X \rightarrow Y)$ and an IFS A in X. The *max-min-max* composition is denoted as $B = A \bullet R$, where B is an IFS in Y. Given an item $y \in Y$,
$B(y).\mu = max_{x\in X}(min(A(x).\mu, R(\langle x, y\rangle).\mu))$,
$B(y).\nu = min_{x\in X}(max(A(x).\nu, R(\langle x, y\rangle).\nu))$.

Similarly, it is possible to define the composition of two IFRs.

Definition 7. Consider two IFRs $R(X \rightarrow Y)$ and $Q(Y \rightarrow Z)$, where X, Y and Z are three domains. The *max-min-max* composition of R and Q is denoted as $T = R \bullet Q$, where T is an IFR in $X \times Z$. Given an item $\langle x, z\rangle \in X \times Z$,
$T(\langle x, z\rangle).\mu = max_{y\in Y}(min(R(\langle x, y\rangle).\mu, Q(\langle y, z\rangle).\mu))$,
$T(\langle x, z\rangle).\nu = min_{y\in Y}(max(R(\langle x, y\rangle).\nu, Q(\langle y, z\rangle).\nu))$.

In the literature, the notation of the two compositions is $B = R \circ A$ and $T = Q \circ R$ (in our opinion, this notation is a little bit anti-intuitive, because the order of terms is reversed with respect to the way they are composed).

2.3 Example: Representing Medical Knowledge

From [8], we borrow the example for which it is possible to apply IFRs, i.e., representing medical knowledge. Consider three domains: P is the domain of patients, S is the domain of symptoms and D is the domain of diseases. Medical knowledge is described by an IFR $K(S \rightarrow D)$: membership degrees denote the confidence of the physician as far as a given symptom denotes a given disease, as well as the non-membership degree expresses the non-confidence; Fig. 1 shows a sample instance of K, on the right-hand side.

When a patient is visited, symptoms shown by the patient are recorded; since the way they are reported or observed is imprecise, they are represented by an IFR called $O(P \rightarrow S)$; Fig. 1 shows a sample instance of O, on the left-hand side. Composing the two relations O and K, we obtain $PD = O \bullet K$, where PD is an IFR $PD(P \rightarrow D)$ that associates a patient to his/her potential diseases; Fig. 2 shows the instance of the PD IFR obtained by composing the instances of O and K shown in Fig. 1.

Based on this new relation, for each patient $p \in P$ we can build the classical FS D_p in the set D of diseases, which maps a disease $d \in D$ (that is possibly affecting p) to the membership degree, by menas of the following membership function: $D_p(d) = PD(\langle p, d\rangle).\mu - PD(\langle p, d\rangle).\nu \cdot \pi_{PD}(\langle p, d\rangle)$. In words, the membership degree is compensated by the product of the non-membership degree by

O

$P \times S$	$\langle \mu, \nu \rangle$
⟨ John, Temperature ⟩ →	$\langle 0.7, 0.2 \rangle$
⟨ John, Headache ⟩ →	$\langle 0.7, 0.1 \rangle$
⟨ John, Cough ⟩ →	$\langle 0.5, 0.3 \rangle$
⟨ Jane, Temperature ⟩ →	$\langle 0.0, 0.4 \rangle$
⟨ Jane, Headache ⟩ →	$\langle 0.4, 0.5 \rangle$
⟨ Jane, Cough ⟩ →	$\langle 0.2, 0.6 \rangle$

K

$S \times D$	$\langle \mu, \nu \rangle$
⟨ Temperature, Flu ⟩ →	$\langle 0.6, 0.1 \rangle$
⟨ Temperature, COVID-19 ⟩ →	$\langle 0.7, 0.2 \rangle$
⟨ Temperature, Typhoid ⟩ →	$\langle 0.4, 0.5 \rangle$
⟨ Headache, Flu ⟩ →	$\langle 0.4, 0.5 \rangle$
⟨ Headache, COVID-19 ⟩ →	$\langle 0.5, 0.3 \rangle$
⟨ Headache, Typhoid ⟩ →	$\langle 0.5, 0.2 \rangle$
⟨ Cough, Flu ⟩ →	$\langle 0.2, 0.3 \rangle$
⟨ Cough, COVID-19 ⟩ →	$\langle 0.4, 0.5 \rangle$
⟨ Cough, Typhoid ⟩ →	$\langle 0.1, 0.8 \rangle$

Fig. 1. IFRs O (symptoms shown by patients) and K (medical knowledge).

the hesitation degree; in [15], the *normalized Hamming distance* is used as membership function for D_p, which is probably better but this issue is outside the scope of this paper. The final disease chosen for each patient p is the one having the greatest membership degree in D_p. Figure 2 shows, on the right-hand side, the D_p FSs obtained for patients John and Jane, whose symptoms are reported in the instance of the O IFR shown in Fig. 1.

3 Intuitionistic Fuzzy Sets and *J-CO-QL*$^+$

We now address the core contribution of the paper. First of all, we clearly define the problem addressed in the study.

Definition 8 - Problem 1. Suppose that a collection `SymptomsDiseases` of *JSON* documents represents the IFR denoted as K in Sect. 2.3; also suppose that a collection `PatientsSymptoms` of *JSON* documents represents the IFR called O in Sect. 2.3. By means of *J-CO-QL*$^+$, process these two collections as IFRs, so as to obtain, for each patient, the D_p FS and associate each patient with the most likely disease, on the basis of shown symptoms.

Currently, *J-CO-QL*$^+$ is undergoing the revision of statements and the extension of them with constructs to support evaluation of FSs; specifically, for each document d, it is possible to evaluate its membership degrees to several FSs. Consequently, since *J-CO-QL*$^+$ currently does not support IFSs, in order to solve Problem 1, it is necessary to think in terms of a couple of FSs for each $\langle \mu, \nu \rangle$ pair: the membership degree to a FS called `R_Membership` denotes $R(d).\mu$; the membership degree to a FS called `R_NonMembership` denotes $R(d).\nu$ (where R is the name of the IFR).

$PD = O \bullet K$

$P \times D$		$\langle \mu, \nu \rangle$
⟨ John, Flu ⟩	→	$\langle 0.6, 0.1 \rangle$
⟨ John, COVID-19 ⟩	→	$\langle 0.7, 0.2 \rangle$
⟨ John, Typhoid ⟩	→	$\langle 0.5, 0.2 \rangle$
⟨ Jane, Flu ⟩	→	$\langle 0.2, 0.4 \rangle$
⟨ Jane, COVID-19 ⟩	→	$\langle 0.4, 0.4 \rangle$
⟨ Jane, Typhoid ⟩	→	$\langle 0.4, 0.5 \rangle$

D_P

John: D		μ
Flu	→	0.56
COVID-19	→	0.68
Typhoid	→	0.44

Jane: D		μ
Flu	→	0.04
COVID-19	→	0.32
Typhoid	→	0.35

Fig. 2. IFR $PD = O \bullet K$ and classical fuzzy sets D_P.

3.1 *J-CO-QL*$^+$ Data Model and Execution Model

By means of *J-CO-QL*$^+$, it is possible to write complex queries or scripts, which are able to process collections of *JSON* documents taken from a *JSON* database like *MongoDB*. A query $q = (i_1, \ldots, i_n)$ is a sequence of n (with $n > 0$) instructions i_j (with $1 \leq j \leq n$). Each instruction i_j receives a *query-process state* $s_{(j-1)}$ and produces an output query-process state s_j.

A query-process state is a tuple $s = \langle tc, IR, DBS, FO, JSF \rangle$. Specifically, tc is called *temporary collection* and contains the current collection of *JSON* documents to process; IR is a local and volatile database, where the query can temporarily save *Intermediate Results*; DBS is the set of *database descriptors*, so as to connect to external databases; FO is the pool of *fuzzy operators*, which are used to evaluate membership degrees of documents to FSs (see Sect. 3.2); JSF is the set of user-defined JavaScript functions, defined to empower the query with additional computational capabilities (see [12]).

A query works on collections of *JSON* documents, which can be either retrieved from an external source or database, or generated as temporary collection by previous instructions. So, the basic item to process is a *JSON* document d, as defined by the standard [7]. However, *J-CO-QL*$^+$ gives a special meaning to fields whose name begins with the ~ character (which are still compliant with *JSON* naming rules for fields). Specifically, the `~fuzzysets` field behaves a a key/value map: it is a nested document where the field name corresponds to a FS name, while the value of the field is the membership degree of d to the FS. Furthermore, the `~geometry` field denotes the geometry of the real-world entity described by the document (in this paper, we do not make use of this field; the interested reader can refer to [4,14]).

```
{
    "Symptom"       : "Temperature",
    "Disease"       : "Flu",
    "Confidence"    : 0.6,
    "NonConfidence" : 0.1
}
```

(a) Collection `SymptomsDiseases`.

```
{
    "Patient"       : "John",
    "Symptom"       : "Temperature",
    "Confidence"    : 0.7,
    "NonConfidence" : 0.2
}
```

(b) Collection `PatientsSymptoms`.

Fig. 3. Sample documents in the source collections.

3.2 *J-CO-QL*$^+$ Script

Suppose that two collections are stored in a *MongoDB* database called `MyDB`. The first collection is called `SymptomssDiseases`: its documents represent medical knowledge, thus they correspond to items in the K IFR presented in Sect. 2.3 and shown in the right-hand side of Fig. 1. Figure 3a shows an example document in this collection: notice the fields named `Confidence` and `NonConfidence`, by means of which physicians express their confidence/non-confidence (which are going to become membership/non-membership to an IFR, but for physicians it is better to think in terms of confidence/non-confidence) for a sympotm to denote a given disease.

The second collection is named `PatientsSymptoms` and its documents represent medical observations (the O IFR introduced in Sect. 2.3 and shown in the left-hand side of Fig. 1). Figure 3b shows an example doument: the reader can see the presence of the fields named `Confidence` and `NonConfidence` too.

The script we wrote to address Problem 1 is reported in Listings 1 and 2; it is necessary to break it in two distinct parts, due to its length.

- The `USE DB` instruction on line 1 connects the query process to the *MongoDB* database called `MyDB`, managed by a server running on the local machine. It changes the *DBS* member in the query-process state; the other members (included the temporary collection) remain empty.

Listing 1 *J-CO-QL*$^+$ script, Part 1.

```
1. USE DB MyDB  ON SERVER MongoDB 'http://127.0.0.1:27017';

2. JOIN OF COLLECTIONS  PatientSymptom@MyDB AS O, SymptomsDisease@MyDB AS K
     CASE  WHERE  .O.Symptom = .K.Symptom;

3. CREATE FUZZY OPERATOR Fuzzify
     PARAMETERS     Membership TYPE FLOAT
     PRECONDITION   Membership >= 0 AND Membership <= 1
     EVALUATE       Membership
     POLYLINE       [ (0, 0),  (1, 1) ];

4. FILTER
     CASE  WHERE  WITH .K.Confidence, .K.NonConfidence, .O.Confidence, .O.NonConfidence
       GENERATE
         CHECK FOR
           FUZZY SET K_Membership    USING Fuzzify(.K.Confidence),
           FUZZY SET K_NonMembership USING Fuzzify(.K.NonConfidence),
           FUZZY SET O_Membership    USING Fuzzify(.O.Confidence),
           FUZZY SET O_NonMembership USING Fuzzify(.O.NonConfidence);

5. FILTER
     CASE  WHERE  KNOWN FUZZY SETS K_Membership, K_NonMembership,
                                   O_Membership, O_NonMembership
         GENERATE
           CHECK FOR
             FUZZY SET Pair_Membership     USING O_Membership      AND  K_Membership,
             FUZZY SET Pair_NonMembership  USING O_NonMembership   OR   K_NonMembership
           BUILD  { .Patient : .O.Patient,
                    .Symptom : .O.Symptom,
                    .Disease : .K.Disease }
           KEEPING FUZZY SETS Pair_Membership, Pair_NonMembership;

6. GROUP
     PARTITION WITH .Patient, .Disease
       BY .Patient, .Disease   INTO .Items;
```

- The `JOIN` instruction on line 2 has a twofold goal: it retrieves the two collections and joins their documents, aliased as `O` (the `PatientsSymptoms` collection) and as `K` (the `SymptomsDiseases` collection) for simplicity; as specified by the `WHERE` selection condition, for each pair of source documents o (from the `O` collection) and k (from the `K` collection), a new document is built if o and k has the same value for the `Symptom` fields. The output document contains two fields, named `O` and `K`, which contain the source paired documents o and k, respectively. Figure 4a shows a sample document, which is obtained by pairing the two documents reported in Figs. 3b and 3a.
  This way, we can say that the structural part of the IFR composition $O \bullet K$ (see Sect. 2.3) has been done; nevertheless, we have not managed membership/non-membership yet.
- So as to deal with FSs, line 3 defines a "fuzzy operator" called `Fuzzify`, which will be used for evaluating membership degrees to FSs. It receives a single parameter called `Membership`; the `PRECONDITION` imposes that its value must be between 0 and 1 (otherwise, an exception is raised).
  The `EVALUATE` expression says that the value of the `Membership` parameter is used as x-axis value for the linear function described by the `POLYLINE` clause (in [10,11,13] the reader can see that complex polylines can be specified as membership functions). The corresponding y-axis value is the membership degree returned by the operator.

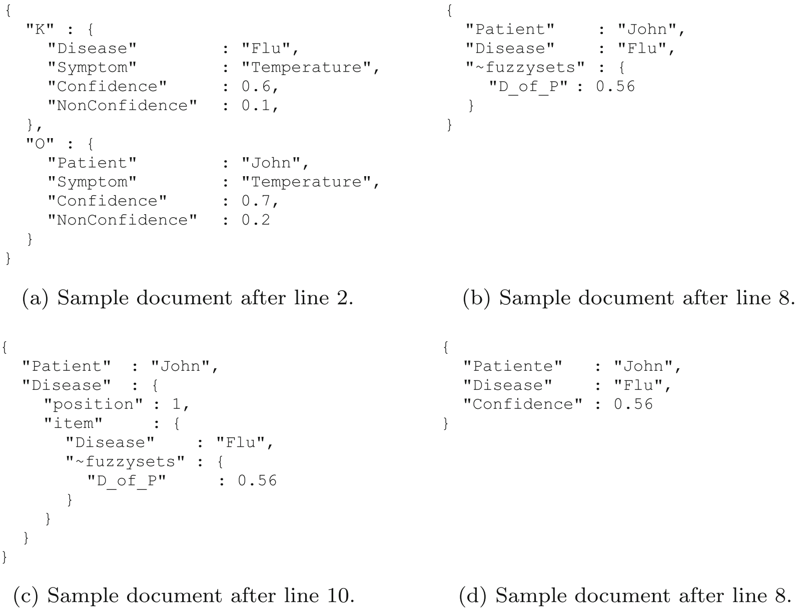
Listing 2 *J-CO-QL*$^+$ script, Part 2.

```
7. CREATE FUZZY OPERATOR toClassical
     PARAMETERS
         Membership     TYPE FLOAT,
         NonMembership  TYPE FLOAT
     PRECONDITION Membership     >= 0 AND Membership     <= 1 AND
                  NonMembership  >= 0 AND NonMembership  <= 1
     EVALUATE     Membership - NonMembership*(1 - Membership - NonMembership)
     POLYLINE     [ (0, 0), (1, 1) ];

8. FILTER
     CASE  WHERE  WITH ARRAY .Items
       GENERATE
         CHECK FOR
           FUZZY SET PD_Membership
             USING Fuzzify(MAX_ARRAY(.Items,NUMERIC,DOCUMENTS,.~fuzzysets.Pair_Membership)),
           FUZZY SET PD_NonMembership
             USING Fuzzify(MIN_ARRAY(.Items,NUMERIC,DOCUMENTS,.~fuzzysets.Pair_NonMembership)),
           FUZZY SET D_of_P
             USING toClassical(MEMBERSHIP_OF(PD_Membership), MEMBERSHIP_OF(PD_NonMembership))
         BUILD {  .Patient : .Patient,
                  .Disease : .Disease   }
         KEEPING FUZZY SETS D_of_P;

9. GROUP
     PARTITION WITH .Patient
       BY .Patient  INTO .Diseases   DROP GROUPING FIELDS
       SORTED BY .~fuzzysets.D_of_P TYPE NUMERIC DESC;

10. EXPAND
     UNPACK WITH .Diseases
       ARRAY .Diseases  TO .Disease;

11. FILTER
     CASE  WHERE  .Disease.position = 1
       GENERATE
         CHECK FOR FUZZY SET D_of_P USING Fuzzify(.Disease.item.~fuzzysets.D_of_P)
         BUILD {  .Patient     : .Patient,
                  .Disease     : .Disease.item.Disease,
                  .Confidence  : MEMBERSHIP_OF(D_o_P)  }
         DEFUZZIFY;

12. SAVE AS Diseases_Of_Patients@MyDB;
```

- Lines 4 and 5 actually evaluate FSs on documents.
  The `FILTER` instruction on line 4 evaluates the membership degrees to four FSs: `K_Membership` and `K_NonMembership` are obtained by means of the `Fuzzify` fuzzy operator in the `USING` clauses, moving from the `Confidence` and `NonConfidence` fields coming from the `K` input collection (thus, their membership degrees correspond to μ_K and ν_K); similarly, the `O_Membership` and `O_NonMembership` FSs represent μ_O and ν_O, resp... Line 5 actually computes the *min* (respectively *max*) value of the membership (respectively, non-membership) of each pair, by obtaining the membership degree to the `Pair_Membership` (resp. `Pair_NonMembership`) FS. The final `BUILD` block simplifies the document structure, keeping only the `Pair_Membership` and `Pair_NonMembership` FSs.
- Line 5 computes the inner part of equations in Definition 7; to complete their computation, it is necessary to aggregate all documents associating the same patient to the same disease, through different symptoms. The `GROUP` instruction on line 6 does that: documents having the same values for the `Patient` and `Disease` fields are grouped together; all grouped documents are put into an array field called `Items`.

To complete the process, it is necessary to continue with the second part of the script, which is reported in Listing 2.

- Line 7 creates a second fuzzy operator, called `toClassical`: its goal is to combine membership and non-membership degrees to obtain one single membership degree for a classical FS (to compute the D_p FSs introduced in



```
{
  "K" : {
    "Disease"        : "Flu",
    "Symptom"        : "Temperature",
    "Confidence"     : 0.6,
    "NonConfidence"  : 0.1,
  },
  "O" : {
    "Patient"        : "John",
    "Symptom"        : "Temperature",
    "Confidence"     : 0.7,
    "NonConfidence"  : 0.2
  }
}
```

(a) Sample document after line 2.

```
{
  "Patient"    : "John",
  "Disease"    : "Flu",
  "~fuzzysets" : {
    "D_of_P" : 0.56
  }
}
```

(b) Sample document after line 8.

```
{
  "Patient"  : "John",
  "Disease"  : {
    "position" : 1,
    "item"     : {
      "Disease"    : "Flu",
      "~fuzzysets" : {
        "D_of_P"     : 0.56
      }
    }
  }
}
```

(c) Sample document after line 10.

```
{
  "Patiente"   : "John",
  "Disease"    : "Flu",
  "Confidence" : 0.56
}
```

(d) Sample document after line 8.

Fig. 4. Sample documents in the temporary collections.

Sect. 2.3). Notice that the operator receives two parameters, `Membership` and `NonMembership`; the `EVALUATE` clause combines them, and the resulting value is returned as membership degree by the operator (see again, the simple polyline defined in the `POLYLINE` clause).

- The `FILTER` instruction on line 8 has a twofold goal. First of all, it completes the computation of membership/non-membership degrees for the IFR composition; this is done by computing the membership degrees to the FSs named `PD_Membership` and `PD_NonMembership`, for which in the `USING` clauses the `Fuzzify` fuzzy operator is called with the value obtained by the `MAX_ARRAY` and `MIN_ARRAY` aggregation functions, which provide the maximum (respectively, minimum) membership degrees to the `Pair_Membership` (respectively, `Peir_NonMebership`) FS, by directly accessing the `~fuzzysets` field of the documents in the `Items` array; in fact, only top-level `~fuzzysets` fields are implicitly managed as membership degrees to FSs by *J-CO-QL*$^+$, but they are regular fields, so can be explicitly accessed by the query. The resulting membership degrees to the `PD_Membership` and `PD_NonMembership` FSs represent μ_{PD} and ν_{PD}, respectively (see the left-hand side of Fig. 2).
  The second goal is to obtain the membership degree to the D_p classical FS, called `D_of_P`: this is done by calling the `toClassical` fuzzy operator, providing, as actual parameters, the membership degrees to the `PD_Membership` and `PD_NonMembership` FSs.
  The final `BUILD` block further simplifies the structure of documents, keeping only the `Patient` and `Disease` fields and the `D_of_P` FS (see the sample document in Fig. 4b).
- The last step to do is to select, for each patient, the disease with the highest membership degree. On line 9, documents are grouped on the basis of the `Patient` field, so as to have one single document for each patient; grouped documents are sorted in reverse order of membership degree to the `D_of_P` FS (notice that it is necessary again to explicitly refer to the `~fuzzysets` field inside the grouped documents).
  The `EXPAND` instruction on line 10 unnests again all grouped documents: this way, as it is possible to see in Fig. 4c, the novel `Disease` field contains both the unnested item (the `item` field) and the `position` field, denoting the position occupied by the item in the original array.
  This structure is suitable to select (line 11) only documents whose field `.Disease.position` is 1, meaning that the document occupied the first position in the source array. For those documents, the membership to the `D_of_P` FS is evaluated again by using the former membership degree (i.e., the maximum one) by explicitly accessing for the third time the nested `~fuzzysets` field; finally the `BUILD` block creates the final structure of output documents (see an example in Fig. 4d), which looses (see the `DEFUZZIFY` keyword) membership degrees, because the `Confidence` field is generated with the membership degree of the `D_of_P` FS.
  The final line 12 saves the last temporary collection into the *MongoDB* database called `MyDB`, with name `Diseases_Of_Patients`.

4 Learned Lessons and Conclusions

Intuitionistic Fuzzy Sets and Relations seem to have a strong potential to represent intuitionistic knowledge and to help processing data with uncertainty when events are judged by humans. Although in the literature it is possible to find many papers that describe their exploitation in practical applications, their adoption is limited by the lack of general-purpose tools.

In Sect. 3, we studied if and how it is possible to adopt *J-CO-QL*$^+$ and its fuzzy constructs to compose two intuitionistic fuzzy relations and choose the most likely option for a given item in a domain. We demonstrated that it is possible to process IFRs, which is good news. Moreover, we saw that the process is tricky, because a specific support is missing. Synthetically, we have learned the following lessons, to pursue as future work.

(i) Managing membership and non-membership degrees as two independent classical fuzzy sets is not the best solution, because they must be dealt with in an integrated way; thus, we guess that a different and polymorphic structure for the special `~fuzzysets` field is necessary, so as to manage classical and intuitionistic fuzzy sets in a unified way.

(ii) Novel constructs are necessary within *J-CO-QL*$^+$, once the data model is adapted, to deal with classical and intuitionistic fuzzy sets at the same time. In particular, novel constructs to transform intuitionistic to classical (and vice versa) fuzzy sets are necessary.

(ii) Fuzzy aggregators over arrays of nested documents with membership degrees to fuzzy sets are currently missing in *J-CO-QL*$^+$; we are aware of this lack; then, it becomes mandatory to define them in such a way they can deal both with classical and intuitionistic fuzzy sets.

The *J-CO* Framework is available on a public GitHub repository[1].

References

1. Atanassov, K.: Intuitionistic fuzzy sets. Fuzzy Sets Syst. **20**, 187–96 (1986)
2. Atanassov, K.: Intuitionistic fuzzy sets. Bioautomation **20**, 1 (2016)
3. Biswas, R.: Intuitionistic fuzzy relations. Bull. Sous. Ens. Flous. Appl. (BUSEFAL) **70**, 22–29 (1997)
4. Bordogna, G., Ciriello, D.E., Psaila, G.: A flexible framework to cross-analyze heterogeneous multi-source geo-referenced information: The J-CO-QL proposal and its implementation. In: Proceedings of the International Conference on Web Intelligence, pp. 499–508 (2017)
5. Bordogna, G., Cuzzocrea, A., Frigerio, L., Psaila, G., Toccu, M.: An interoperable open data framework for discovering popular tours based on geo-tagged tweets. Intell. Inf. Database Syst. **10**(3–4), 246–268 (2017)
6. Bordogna, G., Psaila, G.: Soft aggregation in flexible databases querying based on the vector p-norm. Uncerta. Fuzzi. Knowl. Based Syst. **17**(01), 25–40 (2009)

[1] Github repository: https://github.com/JcoProjectTeam/JcoProjectPage.

7. Bray, T.: The JavaScript object notation (JSON) data interchange format (2014). https://www.rfc-editor.org/rfc/rfc7159.txt
8. De, S.K., Biswas, R., Roy, A.R.: An application of intuitionistic fuzzy sets in medical diagnosis. Fuzzy Sets Syst. **117**(2), 209–213 (2001)
9. Dubois, D., Prade, H.: Possibility theory and its applications: where do we stand. Mathw. Comput. **18**(1), 18–31 (2011)
10. Florescu, D., Fourny, G.: JSONiq: the history of a query language. IEEE Internet Comput. **17**(5), 86–90 (2013)
11. Fosci, P., Marrara, S., Psaila, G.: Soft querying GeoJSON documents within the J-CO framework. In: 16th International Conference on Web Information Systems and Technologies (WEBIST 2020), pp. 253–265 (2020)
12. Fosci, P., Psaila, G.: Powering soft querying in J-CO-QL with JavaScript functions. In: Sanjurjo González, H., Pastor López, I., García Bringas, P., Quintián, H., Corchado, E. (eds.) SOCO 2021. AISC, vol. 1401, pp. 207–221. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-87869-6_20
13. Fosci, P., Psaila, G.: Towards flexible retrieval, integration and analysis of JSON data sets through fuzzy sets: a case study. Information **12**(7), 258 (2021)
14. Psaila, G., Fosci, P.: J-CO: a platform-independent framework for managing georeferenced JSON data sets. Electronics **10**(5), 621 (2021)
15. Szmidt, E., Kacprzyk, J.: Intuitionistic fuzzy sets in some medical applications. In: Reusch, B. (ed.) Fuzzy Days 2001. LNCS, vol. 2206, pp. 148–151. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45493-4_19
16. Zadeh, L.A.: Fuzzy sets. Inf. control **8**(3), 338–353 (1965)

Assessing the Efficient Market Hypothesis for Cryptocurrencies with High-Frequency Data Using Time Series Classification

Rafael Ayllón-Gavilán(✉), David Guijo-Rubio, Pedro A. Gutiérrez, and César Hervás-Martínez

Department of Computer Science and Numerical Analysis, University of Córdoba, Córdoba, Spain
{i72aygar,dguijo,pagutierrez,chervas}@uco.es

Abstract. This work analyzes the performance of several state-of-the-art Time Series Classification (TSC) techniques in the cryptocurrency returns modeling field. The data used in this study comprehends the close price of 6 of the principal cryptocurrencies, collected with a frequency of 5 minutes from January 1st to September 21th of 2021. The aim of this work is twofold: 1) to study the weak form of the Efficient Market Hypothesis (EMH) and 2) to examine the veracity behind the theory of the Random Walk Model (RWM). For this, two datasets are built. The first uses autoregressive values, whereas the second dataset is constructed by introducing randomized past values from the time series. Then, a comparison of the performances achieved by the different TSC techniques is carried out. Results obtained show a pronounced difference in terms of performance obtained by all the TSC models when applied to the original dataset against the randomized one. The results achieved by the models applied to the original dataset are significantly better in terms of Area Under ROC Curve (AUC) and Recall. Therefore, the EMH is refused in its weak form, and indisputable evidence against the RWM in a high-frequency scope is provided.

Keywords: Random walk model · High frequency trading · Efficient market hypothesis · Cryptocurrency forecasting

1 Introduction

Since the creation of Bitcoin in 2009, and its exponential growth until today, there has been several projects developed based on its structure, which was devised in 2008 by an unknown entity under the name of Satashi Nakamoto, the blockchain [20]. Initially, the blockchain was conceived as a technology that allowed direct transactions between people, thus preventing the participation of third-party entities. However, the reality of today is that they have become purely speculative tools in many cases, especially in assets of markets with great value and volatility. In this scenario, developing prediction models capable of anticipating variations in the price of these assets becomes a challenge of growing interest.

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 146–155, 2023.
https://doi.org/10.1007/978-3-031-18050-7_14

The prediction of financial assets is a field in which a great variety of challenges have arisen, mainly due to the high level of noise, uncertainty, and external dependencies that characterize the stock market, which are even more substantial for cryptocurrencies.

This work focuses on predicting a binary variable that indicates whether there is or not a pronounced increase in the price. *Intraday* data at a frequency of 5 minutes is used, contextualizing the work on the field of high frequency trading. This field consists in holding, buying or selling positions in a short time horizon, making profit through numerous low-profit trades performed with a high frequency.

The prediction task is carried out with the aim of proving whether the sequence of values of a given time series is independent of each other, that is, if it follows a Random Walk Model (RWM). This is closely related to the Efficient Market Hypothesis (EMH), which, as noted by [17], states that new information regarding market assets has an immediately and direct impact on the current price. Therefore, anticipations to price fluctuations basing our decisions on historical data can not be successful. As we have seen, the concepts of RWM and EMH differ in definition, but both imply the same consequence, which is the unpredictability of the time series in question.

The hypothesis establishing that the prices of cryptocurrencies follow a RWM started only a few years ago, and yet there is much research to be done. In 2019, strong evidence was found against the RWM [1]. In this work, the authors analyzed the market efficiency of daily Bitcoin returns for the period of July 2010 to March 2018, rejecting the RWM through multiple unit root tests and volatility persistence measures. In a later work in 2020, the RWM was rejected again for the daily returns of 10 popular cryptocurrencies, attributing this market inefficiency to the presence of asymmetric volatility clustering [21]. These two papers cover practically all the research done on the study of RWM in cryptocurrencies.

To the best of our knowledge, this is the first time that the RWM is studied along with the EMH for cryptocurrencies in a high frequency scope. For this, a set of experiments are carried out by extracting randomized copies of each cryptocurrency price sequence, and then comparing the model results between the original and shuffled ones. A similar methodology can be found in [16], in which a z-test is performed to evaluate the results achieved by the shuffled and the original price sequences in terms of accuracy. In this sense, the authors of this work evaluated 6 different stock prices, concluding that they were predictable with a 95% confidence interval.

The rest of this paper is organized as follows: related works are described in Sect. 2; Sect. 3 presents the TSC methods selected for the experiments; in Sect. 4, the experimental setup is detailed as well as the performance results and the statistical tests; and, finally, Sect. 5, concludes the work.

2 Literature Review

This Section introduces the two main concepts used in this work: the Efficient Market Hypothesis (EMH) and the Random Walk Model (RWM), as well as some previous studies where they have been analyzed.

One of the first studies where EMH was considered is [11], where the author divided this hypothesis in three levels depending on the set of information considered: 1) *weak* *form*, in which the information set is limited only to historical prices records; 2) *semi-strong form*, that also considers all public available information; and 3) *strong form*, where all historical prices, public and private information are taken into account.

In this previous work, even though statistically significant dependence in series of returns was found, the author concluded that this was not enough to consider that the market is inefficient. Therefore, strongly supporting the weak form of the EMH. In a second work of this author, the weak and semi-strong forms of the EMH were questioned, given that using both public information and lagged values of returns led to a higher predictability [12]. As we will only consider historical prices when training the proposed models, this work is limited to the EMH in its weak form.

On the other hand, the relation between the EMH and the RWM, presented in the previous Section, was introduced in [8], and it was also discussed in [11]. It states that if a market asset meets the EMH, its returns sequence must follow a RWM.

Focusing on the cryptocurrency market, one of the first papers to study its informational efficiency was [24], in which 6 statistical tests were applied to Bitcoin data. The author concluded that Bitcoin does not satisfy the weak form of the EMH, but the existence of a trend towards efficiency as the Bitcoin matures in the market was argued. Later, the previous work was extended in [19], in which a total of 8 different tests to Bitcoin returns were applied. The conclusion of this second work was that the returns of bitcoin met the EMH.

In this line, [15] studied the time-varying long-term memory of daily Bitcoin prices for the period of 2010 to 2017 using a rolling-window approach and computing a new efficiency index. The conclusions drawn from the results differ from those accepted so far, showing that the efficiency of Bitcoin did not vary over time and proving a consistent lack of efficiency in the market. This lack of efficiency was also reported by subsequent works such as [2, 18, 23].

Regarding the time frequency adopted in experiments, not much research was done for high frequency trading, which comprehends *intraday* information of cryptocurrencies. The very first work in this line was [7], which took the hourly prices of the 2 most voluminous cryptocurrencies in the market, from July to August in 2017, with the aim of studying the EMH. The authors concluded that the EMH was unstable over time, showing both periods of high and low efficiency.

Other works such as [3] studied the relation between sampling frequency and price efficiency in 4 of the highest capitalized cryptocurrencies. A set of long memory tests was applied to reveal a U-shaped pattern in the efficiency over frequency function, showing the highest efficiency in 5 and 10 minutes frequencies, and the lowest in 1 and 60 min data.

Based on all of the above, this work aims to contribute to the existing literature on analyzing cryptocurrencies returns from a machine learning perspective, adopting a binary classification problem. For the first time in the literature, the RWM together with the EMH are studied in high-frequency data. With the emerging status of cryptocurrencies, it is crucial to thoroughly study the nature of these market assets, so investors and portfolio managers can be aware of the risk involved in trading them.

3 Methods

In this study, four different Time Series Classification (TSC) techniques belonging to the literature have been used. Brief descriptions are given below:

Time Series Forest (TSF) is an ensemble of decision trees, in which random intervals are sampled for each tree, and, then, the mean, standard deviation and slope for each of those intervals are found, concatenating this information to build a new dataset [10]. Finally, considering estimates of the averaged probability, the ensemble of trees is constructed.

Supervised Time Series Forest (STSF) is similar to TSF [5]. The main difference is on the way the decision trees are built, given that the intervals are now selected through a supervised process.

Random Convolutional Kernel Transform (ROCKET) consists in convolving the data with a fixed number of kernels generated with random properties [9]. The result of this convolution step is introduced to a ridge regression classifier.

InceptionTime is an ensemble of five deep learning models called Inception networks [13]. These networks are equal in architecture, but are initialized with different random weight values. They are applied with the aim of extracting relevant features from the time series, that finally serve as inputs to a fully connected layer.

Additionally, a **persistence** model is built on the data, whose behavior consists in considering the value of the time series at time t to be equal to the previous obverved value (time $t-1$). This model is considered as a baseline when dealing with time series prediction tasks.

4 Experiments and Results

4.1 Datasets Used

The dataset used in the experiments is taken from a Kaggle competition[1] whose goal was to model and predict the returns of 6 of the main cryptocurrencies. The dataset comprises the prices from January 1st to September 21st of 2021, collected with frequency of 5 minutes. The cryptocurrencies considered are Binance Coin, Bitcoin, Dogecoin, Ethereum, Litecoin and Tron.

In order to make the time series stationary, the prices returns are considered. We define the return in a time period t as $R_t = p_t - p_{t-1}$, where p_t and p_{t-1} are the close price in t and $t-1$, respectively.

In this way, a cryptocurrency returns time series is defined as $R = \{R_t, t \in T\}$, where T is a set of equally separated timestamps. For this definition, it is said that R follows a RWM when it fulfills $f(R_{t+1}|\Phi_t) = f(R_{t+1})$, where f denotes a density function, that must be the same for all t, and Φ_t represents the set of information considered in time t [11].

As 5 min frequency data is examined, Φ_t is set equal to the last hour of returns information. On this basis, the feature vector is defined as $X \in \mathbb{R}^{12}$, where $X = \{R_{t-12}, R_{t-11}, \dots, R_{t-1}\}$.

[1] https://www.kaggle.com/competitions/g-research-crypto-forecasting/data.

Our target variable Y is a binary discretization of the return in time stamp t, which is computed through the following function:

$$Y(R_t) = \begin{cases} 0, \text{ if } R_t \leq \theta, \\ 1, \text{ if } R_t > \theta, \end{cases} \tag{1}$$

where θ denotes a threshold value.

As the transaction cost in the majority of brokers is around 0.2%, more than the double of this value is considered to ensure capturing a big enough price fluctuation. Based on that, the threshold value θ is set to 0.0045 (0.45%). As this fluctuation corresponds to an unusual price variation, the discretization with the $Y(r_t)$ function produces a highly imbalanced dataset. Specifically, the BinanceCoin data includes 73774 positives and 1949 negatives labels, 75074 and 674 for Bitcoin, 78838 and 1910 for Litecoin, 74408 and 1337 for Tron, 71004 and 4743 for Dogecoin, 74457 and 1290 for Ethereum, respectively. In order to alleviate data imbalance, a Random UnderSampling procedure (RUS) [6] is applied to keep a proportion of 3 negatives to 1 positive. The RUS technique is only applied to the training datasets, whereas the testing datasets remain invariant.

Furthermore, two versions of the datasets for each cryptocurrency are built: 1) the original version corresponding to the sequence of prices correctly ordered in time; and 2) a randomized version, where the sequence of prices is disordered with respect to time. If any of the cryptocurrency returns time series has any hidden pattern that can be modeled by the proposed methods, it will disappear in the randomized dataset, leading to significant differences in performance. Hence, it means that the cryptocurrencies time series with high frequency data does not follow an RWM.

Once the features and targets are constructed, a 70% of the patterns are randomly selected for training, whereas the remaining 30% are kept in the generalization set, used for computing the performance metrics. Note that this random process does not disorder the returns in time, but rather the already constructed instances. The process followed for the preprocessing of the data is summarized in Fig. 1.

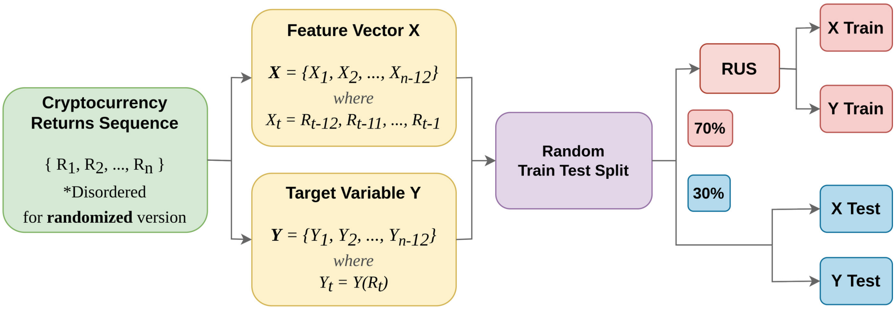

Cryptocurrency Returns Sequence
{ $R_1, R_2, ..., R_n$ }
*Disordered for **randomized** version
Feature Vector X
$\boldsymbol{X} = \{X_1, X_2, ..., X_{n-12}\}$
where
$X_t = R_{t-12}, R_{t-11}, ..., R_{t-1}$
Target Variable Y
$\boldsymbol{Y} = \{Y_1, Y_2, ..., Y_{n-12}\}$
where
$Y_t = Y(R_t)$
Random Train Test Split
70%
30%
RUS
X Train
Y Train
X Test
Y Test

Fig. 1. Flowchart of the preprocessing steps carried out to the original cryptocurrencies datasets.

4.2 Experimental Settings and Performance Measures

Regarding the machine learning techniques, those explained in Sect. 3 are used. Specifically, the stochastic ones are run 5 times, in order to remove their possible bias. Moreover, given that randomness also takes part in the sampling of the training and testing datasets, 20 stratified resamples without replacement of the original datasets are considered. Hence, a total of $20 \times 5 = 100$ fits are performed for each method and cryptocurrency. It is worthy of mention that all the techniques except the persistence have a stochastic nature.

In addition, two performance measures are used for assessing the results achieved: 1) Recall [22], which is defined as $Recall = TP/P$, where TP is the number of correctly classified positive patterns and P indicates the total number of positive patterns, and 2) the Area Under the ROC Curve (AUC), defined in [4].

In order to demonstrate whether statistical differences exist, the hypothesis test is defined as:

$$H = \begin{cases} H_0, & \text{if } p_\mathrm{a} \geq p_\mathrm{o}, \\ H_1, & \text{if } p_\mathrm{a} < p_\mathrm{o}, \end{cases} \tag{2}$$

where p_a and p_o represent the performance of the methods on the randomized and original versions of the datasets, respectively.

In this case, rejecting the null hypothesis means that historical prices provide useful information when anticipating future price fluctuations, which is the opposite to what states the weak form of the EMH. For this, a non-parametric Wilcoxon test [25], is applied to check the existence of significant differences in median of Recall scores, among the best methodology applied to the original dataset versus the same one applied to randomized dataset. A level of significance of $\alpha = 0.05$ is considered, and the corresponding correction for the number of comparisons is also included. As only two algorithms are compared for each of the cryptocurrencies time series dataset, the total number of comparisons is 6, hence, the corrected level of significance is $\alpha = 0.05/6 = 0.0083$. Likewise, the same hypothesis test is proposed for studying the existence of significant differences in the median values of AUC.

4.3 Results

Tables 1 and 2 show the results achieved for all the methods and cryptocurrencies in terms of Recall and AUC, respectively. Values are represented as $mean_{std}$, where std is the standard deviation. Also, best results for each cryptocurrency are included in bold, whereas second best results are in italics. As can be seen, results are included for each cryptocurrency using both versions (original and randomized) except for the Persistence, given that the prediction would be the same.

Moreover, to give a visual insight into the results, Fig. 2 includes the performances in terms of both metrics (Recall and AUC) obtained by each method.

Table 1. Results achieved in terms of Recall ($mean_{std}$ of the 100 fits).

	Persistence	InceptionTime	ROCKET	STSF	TSF
BinanceCoin	$0.1153_{0.0000}$	$0.5419_{0.0495}$	$\mathit{0.5428_{0.0277}}$	$\mathbf{0.6313_{0.0464}}$	$0.5075_{0.0358}$
BinanceCoin randomized		$\mathbf{0.1861_{0.0196}}$	$\mathit{0.0552_{0.0113}}$	$0.0082_{0.0089}$	$0.0082_{0.0164}$
Bitcoin	$0.0455_{0.0000}$	$\mathit{0.4717_{0.0330}}$	$0.4697_{0.0398}$	$\mathbf{0.5374_{0.0556}}$	$0.4313_{0.0409}$
Bitcoin randomized		$\mathbf{0.1663_{0.0328}}$	$\mathit{0.0915_{0.0242}}$	$0.0204_{0.0183}$	$0.0188_{0.0284}$
Dogecoin	$0.2134_{0.0000}$	$0.4859_{0.0279}$	$\mathit{0.5037_{0.0124}}$	$\mathbf{0.5960_{0.0321}}$	$0.4673_{0.0300}$
Dogecoin randomized		$\mathbf{0.1771_{0.0208}}$	$\mathit{0.0201_{0.0048}}$	$0.0057_{0.0066}$	$0.0027_{0.0068}$
Ethereum	$0.0895_{0.0000}$	$0.5250_{0.0323}$	$\mathit{0.5330_{0.0319}}$	$\mathbf{0.5924_{0.0509}}$	$0.4629_{0.0443}$
Ethereum randomized		$\mathbf{0.1896_{0.0292}}$	$\mathit{0.0746_{0.0164}}$	$0.0090_{0.0105}$	$0.0055_{0.0120}$
Litecoin	$0.0746_{0.0000}$	$0.4357_{0.0381}$	$\mathit{0.4375_{0.0285}}$	$\mathbf{0.5024_{0.0493}}$	$0.3715_{0.0484}$
Litecoin randomized		$\mathbf{0.1843_{0.0245}}$	$\mathit{0.0494_{0.0126}}$	$0.0069_{0.0087}$	$0.0087_{0.0162}$
Tron	$0.0922_{0.0000}$	$\mathit{0.4762_{0.0377}}$	$0.4698_{0.0271}$	$\mathbf{0.5489_{0.0592}}$	$0.3978_{0.0409}$
Tron randomized		$\mathbf{0.1827_{0.0267}}$	$\mathit{0.0591_{0.0161}}$	$0.0118_{0.0107}$	$0.0100_{0.0197}$

Table 2. Results achieved in terms of AUC ($mean_{std}$ of the 100 fits).

	Persistence	InceptionTime	ROCKET	STSF	TSF
BinanceCoin	$0.5464_{0.0000}$	$0.7117_{0.0186}$	$\mathit{0.7269_{0.0137}}$	$\mathbf{0.7595_{0.0192}}$	$0.7183_{0.0153}$
BinanceCoin randomized		$\mathbf{0.5036_{0.0082}}$	$0.4993_{0.0056}$	$\mathit{0.5004_{0.0023}}$	$0.5002_{0.0017}$
Bitcoin	$0.5183_{0.0000}$	$0.6803_{0.0159}$	$\mathit{0.6968_{0.0194}}$	$\mathbf{0.7244_{0.0246}}$	$0.6894_{0.0181}$
Bitcoin randomized		$0.4926_{0.0148}$	$0.4985_{0.0116}$	$\mathit{0.4994_{0.0068}}$	$\mathbf{0.5012_{0.0056}}$
Dogecoin	$0.5804_{0.0000}$	$0.6777_{0.0092}$	$\mathit{0.7030_{0.0061}}$	$\mathbf{0.7343_{0.0132}}$	$0.6938_{0.0122}$
Dogecoin randomized		$0.4976_{0.0050}$	$0.4995_{0.0022}$	$\mathbf{0.5003_{0.0012}}$	$\mathit{0.5000_{0.0008}}$
Ethereum	$0.5375_{0.0000}$	$0.7037_{0.0150}$	$\mathit{0.7248_{0.0158}}$	$\mathbf{0.7444_{0.0223}}$	$0.6998_{0.0198}$
Ethereum randomized		$\mathbf{0.5022_{0.0117}}$	$\mathit{0.5009_{0.0078}}$	$0.5001_{0.0031}$	$0.5002_{0.0023}$
Litecoin	$0.5265_{0.0000}$	$0.6504_{0.0154}$	$\mathit{0.6740_{0.0135}}$	$\mathbf{0.6958_{0.0199}}$	$0.6540_{0.0201}$
Litecoin randomized		$\mathbf{0.5035_{0.0107}}$	$0.4955_{0.0057}$	$\mathit{0.5005_{0.0026}}$	$0.4995_{0.0036}$
Tron	$0.5384_{0.0000}$	$0.6734_{0.0165}$	$\mathit{0.6868_{0.0140}}$	$\mathbf{0.7174_{0.0237}}$	$0.6649_{0.0168}$
Tron randomized		$\mathbf{0.5034_{0.0119}}$	$0.4945_{0.0082}$	$\mathit{0.5002_{0.0036}}$	$0.5001_{0.0033}$

Furthermore, in order to check whether significant differences exist between the four TSC techniques in terms of both Recall and AUC, two critical differences diagrams are shown in Fig. 3. As can be seen, the best results are achieved by the STSF technique, achieving ranks of 1.00 for both performance metrics. Hereinafter, the STSF technique is used for analyzing whether statistical differences exist between the original and the randomized datasets.

Finally, the results for the statistical tests regarding the differences between the original and the randomized versions returned a p-value equal to zero for all cryptocurrencies, rejecting the null hypothesis for all of them. Hence, RWM is not satisfied, given that using historical data in our models does provide useful information when making

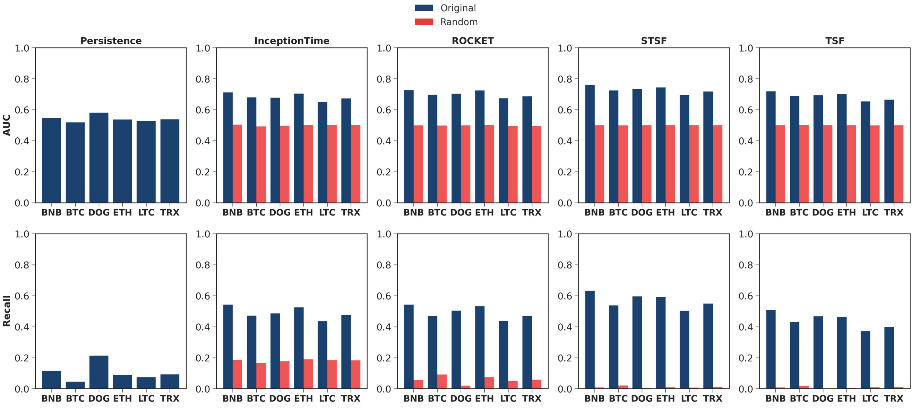
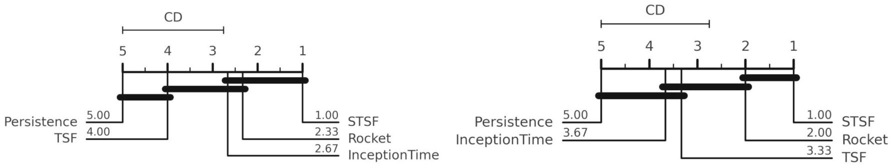


Fig. 2. Results achieved by the four TSC techniques and the persistence.



(a) Recall

(b) AUC

Fig. 3. Critical difference diagrams obtained for both performance measures using the four TSC techniques and the Persistence.

predictions. This fact is not compatible with the EMH, since it shows the existence of inefficiencies throughout the data, which are exploited to obtain better results in the original version than in the randomized one.

5 Conclusions

This paper studies the application of different Time Series Classification (TSC) techniques to 6 cryptocurrencies. The main goal of this work is to analyze the concepts of the Efficient Market Hypothesis (EMH) in its weak form and the Random Walk Model (RWM). For this, two datasets are used: the original, in which the sequence of previous values maintains the order, and the randomized, in which the sequence is disordered. The experiments carried out confirm that the returns time series of the 6 cryptocurrencies under study do not follow an RWM, given that the results achieved for the original datasets are significantly better than those obtained for randomized ones. In view of this fact, the EMH is also rejected, given that it has been proved that considering historical information of the asset leads to a higher predictability. The predictability of these inefficiencies is moderate for all the chosen methods, finding the best results for the Supervised Time Series Forest (STSF). In view of the performance obtained, strong evidence of market inefficiency is found for high-frequency data, because we can consistently anticipate to price fluctuations. Hence, this enforces the conclusions given in previous works like [1] and [21], where high inefficiencies were also found for daily returns time series data.

As future improvements, a larger time frame should be considered to obtain more robust results. In addition, instead of considering the problem from the nominal TSC point of view, it would be of high interest to tackle it from the ordinal TSC [14] one. Thus, trying to predict variations of different magnitudes, differentiating between increases or decreases in prices.

Acknowledgements. This work has been supported by "Agencia Española de Investigación (España)" (grant reference: PID2020-115454GB-C22/AEI/10.13039/501100011033); the "Consejería de Salud y Familia (Junta de Andalucía)" (grant reference: PS-2020-780); and the "Consejería de Transformación Económica, Industria, Conocimiento y Universidades (Junta de Andalucía) y Programa Operativo FEDER 2014-2020" (grant references: UCO-1261651 and PY20_00074). David Guijo-Rubio's research has been subsidized by the University of Córdoba through grants to Public Universities for the requalification of the Spanish university system of the Ministry of Universities, financed by the European Union - NextGenerationEU (grant reference: UCOR01MS).

References

1. Aggarwal, D.: Do bitcoins follow a random walk model? Res. Econ. **73**(1), 15–22 (2019)
2. Alaoui, M.E., Bouri, E., Roubaud, D.: Bitcoin price-volume: a multifractal cross-correlation approach. Financ. Res. Lett. **31** (2019)
3. Aslan, A., Sensoy, A.: Intraday efficiency-frequency nexus in the cryptocurrency markets. Financ. Res. Lett. **35**, 101298 (2020)
4. Bradley, A.P.: The use of the area under the roc curve in the evaluation of machine learning algorithms. Pattern Recogn. **30**(7), 1145–1159 (1997)
5. Cabello, N., Naghizade, E., Qi, J., Kulik, L.: Fast and accurate time series classification through supervised interval search, pp. 948–953 (2020)
6. Charte, F., Rivera, A.J., del Jesus, M.J., Herrera, F.: Addressing imbalance in multilabel classification: measures and random resampling algorithms. Neurocomputing **163**, 3–16 (2015)
7. Chu, J., Zhang, Y., Chan, S.: The adaptive market hypothesis in the high frequency cryptocurrency market. Int. Rev. Financ. Anal. **64**, 221–231 (2019)
8. Cootner, P.H.: The Random Character of Stock Market Prices. Massachusetts Institute of Technology Press, Cambridge (1964)
9. Dempster, A., Petitjean, F., Webb, G.I.: Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. Data Min. Knowl. Disc. **34**, 1454–1495 (2020)
10. Deng, H., Runger, G., Tuv, E., Vladimir, M.: A time series forest for classification and feature extraction. Inf. Sci. **239**, 142–153 (2013)
11. Fama, E.F.: Efficient capital markets: a review of theory and empirical work. J. Financ. **25**(2), 383–417 (1970)
12. Fama, E.F.: Efficient capital markets: II. J. Financ. **46**(5), 1575–1617 (1991)
13. Fawaz, H.I., et al.: InceptionTime: finding AlexNet for time series classification. Data Min. Knowl. Disc. **34**, 1936–1962 (2020). https://doi.org/10.1007/s10618-020-00710-y

14. Guijo-Rubio, D., Gutiérrez, P.A., Bagnall, A., Hervás-Martínez, C.: Time series ordinal classification via ShapeLets. In: 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2020)
15. Jiang, Y., Nie, H., Ruan, W.: Time-varying long-term memory in bitcoin market. Financ. Res. Lett. **25**, 280–284 (2018)
16. Kaboundan, M.A.: Genetic programming prediction of stock prices. Comput. Econ. **16**, 207–236 (2000)
17. Khuntia, S., Pattanayak, J.: Adaptive market hypothesis and evolving predictability of bitcoin. Econ. Lett. **167**, 26–28 (2018)
18. Latif, S., Mohd, M., Amin, M., Mohamad, A.: Testing the weak form of efficient market in cryptocurrency. J. Eng. Appl. Sci. **12**, 2285–2288 (2017)
19. Nadarajah, S., Chu, J.: On the inefficiency of bitcoin. Econ. Lett. **150**, 6–9 (2017)
20. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system. Decent. Bus. Rev. 21260 (2008)
21. Palamalai, S., Kumar, K.K., Maity, B.: Testing the random walk hypothesis for leading cryptocurrencies. Borsa Istanbul Rev. **21**(3), 256–268 (2021)
22. Powers, D.: Evaluation: from precision, recall and F-factor to ROC, informedness, markedness & correlation. Mach. Learn. Technol. **2** (2008)
23. Tiwari, A.K., Jana, R., Das, D., Roubaud, D.: Informational efficiency of bitcoin-an extension. Econ. Lett. **163**, 106–109 (2018)
24. Urquhart, A.: The inefficiency of bitcoin. Econ. Lett. **148**, 80–82 (2016)
25. Wilcoxon, F.: Individual comparisons by ranking methods. Biomet. Bull. **1**(6), 80–83 (1945)

Blockchain for Supply Chain Traceability with Data Validation

Cristian Valencia-Payan[1], David Griol[2](✉), and Juan Carlos Corrales[1]

[1] Department of Telematics, Universidad del Cauca, Popayán, Colombia
{chpayan,jcorral}@unicauca.edu.co

[2] Department of Software Engineering, University of Granada, CITIC-UGR, Granada, Spain
dgriol@ugr.es

Abstract. Sustainable supply chain management has been proposed to integrate environmental and financial practices into the supply chain to reduce the risk of business and ensure chain's sustainability meeting environmental, economic and social goals. However, the ever-changing environment has made traceability processes in supplies chains should be carried out with special care to avoid vulnerabilities at any level of the supply chain. In this paper, we describe how a Blockchain-based Global Supply Chain Traceability and the use of smart contracts to validate the transactional data can support the sustainable supply chain management through the use of smart contracts with data validation. Our proposal has been evaluated using Hyperledger Caliper, obtaining a combined average throughput of 401 transactions per second and an average latency of 0.24 s.

Keywords: Blockchain · Sustainable supply chain management · Traceability · Industry 4.0

1 Introduction

Sustainable supply chain management (SSCM) has been proposed as a solution to meet environmental, economic, and social goals in the supply chain processes [10,18,21]. However, there are still key challenges in the sustainability of supply chains in a world becoming more of a global economy, such as waste management, employees state, child labor, etc. With the rise of Industry 4.0, smart manufacturing must be also considered a challenge for SSCM due to the supply chain further segmentation [3,11]. In addition, the ever-changing environment has made the supply chains vulnerable at many levels, given that the traceability of a product must be done carefully completed to avoid future problems.

Tracing carefully a product has become of great importance in a global supply chain (GSC) [6,20]. Businesses must be agile, with high resilience and risk mitigation to survive in the current and complex GSC environment [3]. Blockchain technology improves traceability, quality control, safety, and reliability processes [9].

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 156–165, 2023.
https://doi.org/10.1007/978-3-031-18050-7_15

Blockchain technology also improves trust, accessibility, transparency, and reliability of traceability processes using smart contracts deployed on the network that cannot be modified. In addition, Blockchain technology can present new threats to the traceability processes like phishing, routing, and sibyl attacks, as well as data integrity and reliability [8]. For this reason, data integrity and reliability must be addressed before the network block creation. This can also assist in processes such as those proposed by [4,19] in which data from the Blockchain network is used as input into soft computing algorithms for decision making, user segmentation, etc.

The main objective of this paper is to introduce a Blockchain traceability smart contract with data validation. Typically, smart contracts are used to perform transactions that are valid when a set of predefined conditions are met. Our proposal seeks to provide the smart contract with data analysis techniques to determine when the transaction data, such as storage temperature, ingredients, origin, etc., does not correspond with the historical information available on the network. An innovative use of smart contracts to detect anomalies in data would be a first in the use of Blockchain technology for traceability. With this type of analysis, it is possible to identify problems as soon as they arise, improve decision-making processes, and automate management recommendations and price penalties.

The remainder of the paper is as follows. Section 2 presents the main works related to data storage in Blockchain. Section 3 describes the main concepts related to Blockchain-based GSC traceability. Sections 4 and 5 describe our proposal for the definition of smart contracts for GSC traceability, their implementation and evaluation. Finally, Sect. 6 presents the conclusions and future research lines.

2 Related Work

A systematic mapping in scientific research databases based on the methodology proposed in [17] was performed to find relevant contributions focused on data warehousing in the Blockchain. We selected the following researches.

An ontology-based detection framework for anomalous transactions on the Blockchain in business processes is proposed in [16], The proposed framework is evaluated on transaction logs of a simulated Blockchain network. However, the authors do not consider that manipulated data can be included in a typical transaction. Tian et al. highlight the importance of information sharing in a SC and how, with the help of Blockchain technology, it can be used to solve the problems associated with information sharing [2]. A set of models for the development of an information sharing platform is proposed. However, no consideration is given to the quality or reliability of the information being shared.

A subgraph-based anomaly detection method able to run on GPUs was proposed in [15], considering the number of transactions made per second in a Blockchain system. However, the proposed method only works with fraudulent transactions or stolen secret keys. Blockchain has also been proposed to help detect anomalies in IoT devices and in electricity consumption as seen on [12,14]. Also, an abnormal smart contract detection was proposed in [13] using Heterogeneous Graph Transformer Networks focused on financial fraud. Anomaly detection, however, is performed by tools outside the network, which leaves the door open to abuse.

A data-driven SC management is proposed in [2] to develop a framework that improves SC management. The authors conclude that it is required to develop data-driven digital technologies to perform data collection and management, secure storage, and effective data processing for supply chain security, cost-competitive, and sustainability. The main potential contributions of Blockchain to risk management in SC are presented in [7]. However, this survey does not address the possibility of manipulated data, incorrectly sensed, or errors in the sensing devices.

Although related works have dealt with the different sources of information and how they are shared among supply chain members, all of them consider the assumption that the data associated with transactions do not present any problem. However, in order to have full reliability in supply chains supported by Blockchain technology, it is necessary to consider this possibility that the data sent in a transaction could be manipulated or incorrectly perceived such as semantic or syntax errors. The tools to perform said process should be integrated into the Blockchain network to ensure its immutability.

3 Blockchain-Based GSC Traceability

A GSC is a supply chain that extends beyond a single national boundary [10]. Products and services are distributed to maximize profits and minimize waste for all active companies in the supply chain. Most of the members of this chain are transnational companies. Figure 1 shows the general GSC representation. On the left side, suppliers collect, store, transport and manage the raw materials. Companies oversee the management of the materials and the goods and services resulting from their transformation, distribution and storage. Finally, consumers determine which goods and services are produced and/or offered in greater quantities and how they are distributed.

Information flows depend on multiple factors. It can be difficult and costly to identify when a good or service has undergone a modification that could seriously affect the end consumer. In these cases, Blockchain technology can speed up the processes of verifying the conditions of the particular good or service. In the GSC, reputable companies are highly scrutinized for their sustainability performance in the environmental, social and economic dimensions.

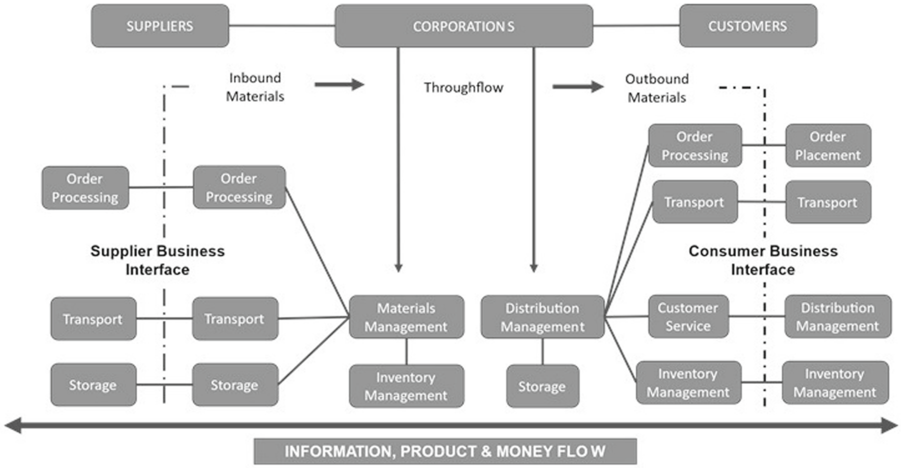


Fig. 1. General global supply chain representation based on [5]

In a traditional traceability scheme, information is stored on private servers with limited access for external members of the server organization. This design delays access to information, which inevitably also affects decision-making processes on production goods and/or services. In the Blockchain-based scheme, each server on the chain can be configured as a node in the network. The most relevant information is available on all nodes simultaneously, which facilitates access to information and allows for streamlined decision-making processes. Similarly, the SSCM benefits from this type of scheme by improving the coordination of information flow in the SC.

4 Smart Contract for GSC Traceability

The smart contract deployed records all information generated by the GSC members from raw material to final goods or services. Initially, the smart contract will issue notifications and/or recommendations based on the terms initially agreed upon by the GSC members. This information and, subsequently, the information sent in each transaction is not susceptible to manipulation.

Once a transaction invokes the smart contract, it will automatically perform data validation looking for errors or anomalies; after the verification, it will generate recommendations based on the control variables; these recommendations will be available to all members of the GSC. To do this, we propose using a structure similar to MAPE-K [1].

The proposed knowledge component consists of all the information generated by the different smart contracts and the transactions associated with them in the Blockchain network (i.e., the knowledge component consists of all the nodes that store the network information). The monitoring module oversees each transaction made on the network, looking for syntax errors or semantic anomalies previously identified to notify those involved promptly about them.

The analysis module is responsible for finding errors or anomalies that have not been previously identified, using the history of data stored in the network, making the comparison with those that have been previously detected, and looking for ways to recall such errors or anomalies quickly and efficiently in the future. The planning module considers the information provided by the analysis module to update the smart contracts deployed in the network so that new transactions related to these contracts can be validated for new errors or anomalies found. Finally, the execution module sends the smart contract update transaction when required or a notification transaction to those involved in the traceability process so that they can make the appropriate decisions to prevent errors from recurring.

With the proposed scheme, it is possible to achieve greater control over the status of products sold, and increase the information available for tracking the processes involved in the path of an altered product, available on a permanent and updated basis. This information could be used to generate management recommendations and/or to set price penalties.

Table 1 shows the description of the variables used to describe how the transaction is processed by the smart contract.

Table 1. States and input variables

Variable	Description
SC_{vark}	State variables used by the smart contract
$SC_{errVark}$	State variables that store the errors found in SCvark
SC_{dVal}	Validation response of the dVal function

The process involved in the smart contract is carried out as follows. First, data is collected and sent to the Blockchain network:

$$T_S = \sum_{k=0}^{n} SC_{vark} \tag{1}$$

where T_S is a transaction related to the traced product. In all cases, n is the total number of invokes.

All transaction variables are set using the capabilities of the SC member to perceive those parameters. After that, the transaction using the variables measured in the coffee bean for each smart contract is sent to the network, and a smart contract is made:

$$InvCall = \sum_{k=0}^{n} Cx(Sx_k, Tx_k) \tag{2}$$

where $InvCall$ is the invoke calling of a smart contract deployed on the network, Cx is the current smart contract, Sx_k is the possible status in the smart contract variables, Tx_k is the transactional data sent to the invoked smart contract, and x represents any of the transactions mentioned before.

The network transactions return the result of the invocation call. The smart contract calls the data validation process if the transaction is valid. If it is not, the state is retained and will be received accordingly:

$$InvResp = \begin{cases} dVal(\sum_{k=0}^{n} Tx_k) \text{ when the information is valid} \\ \sum_{k=0}^{n} (Sx_k, Rx) \quad \text{when the transaction includes invalid data.} \end{cases} \tag{3}$$

where $InvResp$ is the response of the smart contract, $dVal$ is the data validation function, x represents any of the transactions mentioned before.

If the transaction is valid, the smart contract evaluates the transaction data to check for syntactic and/or semantic anomalies. In a Blockchain network, transactions are managed as text entries. A syntax error occurs when the data value can be converted to its expected type of structures. A semantic error occurs when the data is correctly converted to its expected data type, but its value does not match the actual value:

$$SC_{dVal} = \begin{cases} \sum_{q=0}^{n} (Sx_q, Rx) \text{ when the information is valid} \\ \sum_{k=0}^{n} (Sx_k, Rx) \text{ when the transaction includes invalid data.} \end{cases} \tag{4}$$

If the data is valid, a new status Sx_q will be generated, and the corresponding response will be received. In the event that anomalous data is found, the smart contract will place this data in an additional state that can be viewed by the members of the network and will generate an alert regarding the anomalous data found.

5 Smart Contract Implementation and Performance Evaluation

The Blockchain ecosystem to perform traceability processes and the proposed smart contract has been implemented using Hyperledger Fabric and JavaScript. To evaluate the smart contract, we use Hyperledger Caliper. This toolkit evaluates metrics such as throughput, latency, successful and failed transactions in the Blockchain network. Caliper was configured to simulate 10 clients sending transactions to the network, with a transaction load of 10, 20 or 30 using the fixed load rate driver, a minimum duration of 120 s and a maximum duration of 600 s. The driver used in the TransactionLoad parameter specifies a delay in the system by modifying the transaction per second (TPS) driven parameter.

We have performed stress testing on the network in two stages. In the first stage, the Caliper sends workers to create objects on the Blockchain network.

Subsequently, workers send read transactions for each created object. The second stage is the update stage. In it, the transaction is reviewed for errors or anomalies in the submitted data.

Figures 2, 3 and 4 show the results of the described process. Figure 2 shows the total number of transactions made correctly on the test network using the 10 workers and transaction load values of 10, 20 and 30 respectively. As the number of transactions waiting in the network increases, depending on the selected controller, the number of correct transactions increases. In this case, only the process of creating an object and reading the associated data is performed. Thus, it does not represent a significant load on the network and pending transactions can be completed.

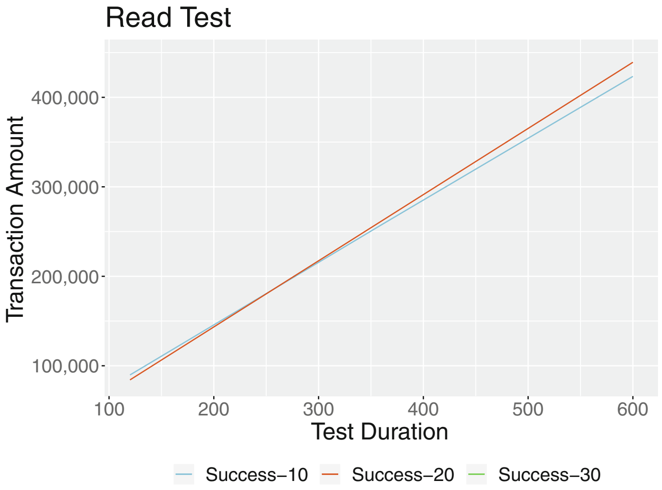


Fig. 2. Successful and failed transactions made in the read test

Figure 3 shows the results of the updated transactions. A drastic decrease in the number of transactions performed on the network is observed. The update transactions require validation of the data sent and generation of alert statuses if necessary. In addition, in this case, there are failed transactions, this transactions are the ones that the network reject, they increase in number as the execution time of each test also increases. On this case failed transactions occur when two or more workers simultaneously attempt to update the same object. However, this problem does not usually occur in real use cases.

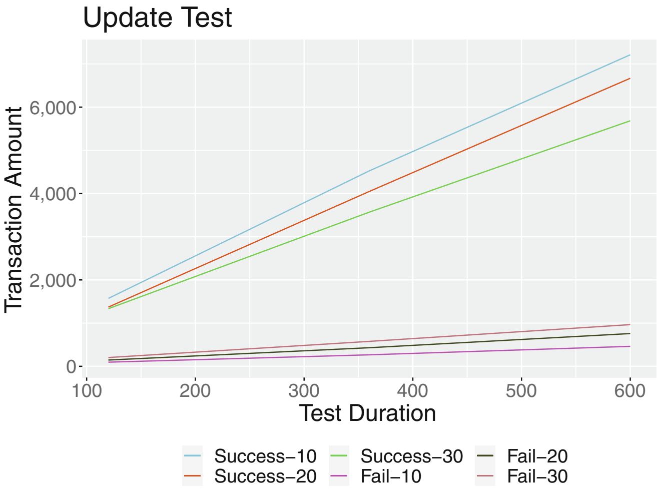
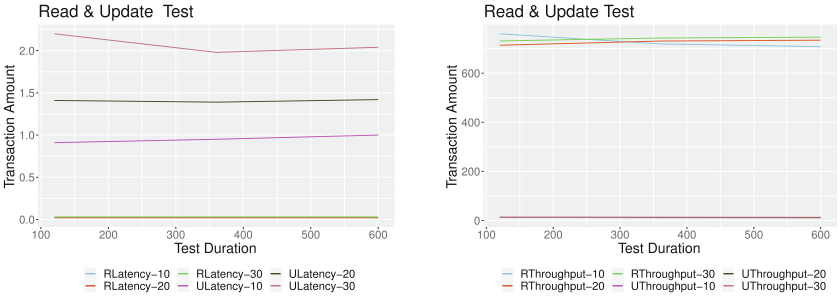


Fig. 3. Successful and failed transactions made in the update test

Figures 4 show the latency and throughput in both cases. The letters R and U are used to differentiate the results of the read test and the update test. Additionally, we include the used transaction load. In the read test, the average latency is 0.023 while the average throughput is 732 TPS. When testing the update, the latency average increases to 1.47, this behavior depends on how many resources are available at the time. The average throughput decreases to 12.5. This is due to the increased use of network resources to perform the operations required by the smart contract as the amount of stored data increases.



(a) Latency results in the read test

(b) Throughput results in the update test

Fig. 4. Latency and throughput results in the update test

6 Conclusions and Future Work

Sustainable supply chain management is a very complex and costly task to identify parameters that can affect the experience of end consumers. To facilitate the identification of this and other problems in a SCM, we have proposed in this work the use of smart contracts with data validation in a traceability scheme in Blockchain.

The proposed smart contract is capable of detecting syntactic and semantic errors in the data and generating alerts about them so that the supply chain operators can detect the problems as soon as possible, correct the causes in time, or prevent the products from reaching the end users. In addition, by adding the necessary expert knowledge, the smart contract is able to generate recommendations regarding the management of the product and other parameters that are not usually considered in other Blockchain-supported traceability proposals.

Considering the results on the evaluation test, the average latency and throughput, we can say that the information uploaded to the Blockchain network, despite the validation processes, will be available to all network members almost immediately.

As future work, additional tests on different configurations of the Blockchain network are required. With this, we pursue the autonomous update of the contracts deployed in the network, so that they can adapt to new types of anomalies that may arise and generate the appropriate recommendations or alerts for them. Additionally, metrics should be developed to measure the performance of the smart contract in correctly detecting anomalies in transaction data. As far as smart contract data analysis is concerned, the development of tools to improve these capabilities over time is important. The use of soft computing tools for the autonomous generation of smart contract anomaly detection rules is also proposed.

Acknowledgements. We want to thank the Telematics Engineering Group (GIT) and the Research Group for Rural Development (Tull) of the University of Cauca and the Ministry of Science Technology and Innovation of Colombia (Minciencias) for the Ph.D. support granted to Cristian Valencia Payan. The research leading to these results has also received funding from "Estrategias para la valorización de los dulces tradicionales de Popayán" (Code: 110380863995) founded by Minciencias - Colombia (Contract 127-2019), the European Union's Horizon 2020 research and innovation program under grant agreement no. 823907 (MENHIR project: https://menhir-project.eu), and the projects supported by the Spanish Ministry of Science and Innovation through the GOMINOLA (PID2020-118112RB-C21 and PID2020-118112RB-C22, funded by MCIN/AEI/10.13039/501100011033), and CAVIAR (TEC2017-84593-C2-1-R, funded by MCIN/ AEI/10.13039/501100011033/FEDER "Una manera de hacer Europa").

References

1. Arcaini, P., Riccobene, E., Scandurra, P.: Modeling and analyzing MAPE-K feedback loops for self-adaptation. In: Proceedings of SEAMS 2015, pp. 13–23 (2015)

2. Bechtsis, D., Tsolakis, N., Iakovou, E., Vlachos, D.: Data-driven secure, resilient and sustainable supply chains: gaps, opportunities, and a new generalised data sharing and data monetisation framework. Int. J. Prod. Res. **207**, 1084–1098 (2021)
3. Ben-Daya, M., Hassini, E., Bahroun, Z.: Internet of things and supply chain management: a literature review. Int. J. Prod. Res. **57**(15–16), 4719–4742 (2019)
4. Erol, I., Ar, I.M., Peker, I.: Scrutinizing blockchain applicability in sustainable supply chains through an integrated fuzzy multi-criteria decision making framework. Appl. Soft Comput. **116**, 108331 (2022)
5. Fastbolt, C.: Program management and supplier of quality fasteners & components (2021). https://fastboltcorp.com/news/the-definitive-guide-to-global-supply-chain-management/
6. Florkowski, W., Shewfelt, R., Brueckner, B., Prussia, S.: Postharvest Handling. Elsevier, Amsterdam (2009)
7. Fridgen, G., Korner, M., Sedlmeir, J., Weibelzahl, M.: (How) can blockchain contribute to the management of systemic risks in global supply networks? In: Proceedings of CEUR Workshop, pp. 89–96 (2019)
8. Grover, J., Sharma, S.: Security issues in wireless sensor network-a review. In: Proceedings of ICRITO 2016, pp. 397–404 (2016)
9. Hong, I.H., et al.: An RFID application in the food supply chain: a case study of convenience stores in Taiwan. J. Food Eng. **106**(2), 119–126 (2011)
10. Koberg, E., Longoni, A.: A systematic review of sustainable supply chain management in global supply chains. J. Clean. Prod. **207**, 1084–1098 (2019)
11. Koberg, E., Longoni, A.: A systematic review of sustainable supply chain management in global supply chains. J. Clean. Prod. **207**, 1084–1098 (2019)
12. Li, M., Zhang, K., Liu, J., Gong, H., Zhang, Z.: Blockchain-based anomaly detection of electricity consumption in smart grids. Pattern Recogn. Lett. **138**, 476–482 (2020)
13. Liu, L., Tsai, W.T., Bhuiyan, M.Z.A., Peng, H., Liu, M.: Blockchain-enabled fraud discovery through abnormal smart contract detection on Ethereum. Futur. Gener. Comput. Syst. **128**, 158–166 (2022)
14. Mirsky, Y., Golomb, T., Elovici, Y.: Lightweight collaborative anomaly detection for the IoT using Blockchain. J. Parallel Distrib. Comput. **145**, 75–97 (2020)
15. Morishima, S.: Scalable anomaly detection in blockchain using graphics processing unit. Comput. Electr. Eng. **92**, 107087 (2021)
16. Musa, T.A., Bouras, A.: Anomaly detection in Blockchain-enabled supply chain: an ontological approach. In: Proceedings of PLM 2021, pp. 253–266 (2022)
17. Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M.: Systematic mapping studies in software engineering. In: Proceedings of Ease 2008, pp. 68–77 (2008)
18. Seuring, S., Muller, M.: From a literature review to a conceptual framework for sustainable supply chain management. J. Clean. Prod. **16**(15), 1699–1710 (2008)
19. Thakur, T., et al.: Smart water conservation through a machine learning and blockchain-enabled decentralized edge computing network. Appl. Soft Comput. **106**, 107274 (2021)
20. Tian, S., Jiang, F., Huang, C.: Global supply chain information compensation model based on free trade port blockchain information platform. Lect. Notes Data Eng. Commun. Technol. **107**, 288–300 (2022)
21. Yakovleva, N., Flynn, A.: Innovation and sustainability in the food system: a case of chicken production and consumption in the UK. J. Environ. Policy Plan. **6**(3–4), 227–250 (2004)

Compression of Clustered Ship Trajectories for Context Learning and Anomaly Detection

David Sánchez Pedroche(✉), Jesús García, and José Manuel Molina

GIAA, University Carlos III of Madrid, Madrid, Spain
{davsanch,jgherrer}@inf.uc3m.es, molina@ia.uc3m.es

Abstract. This paper presents a context information extraction process over Automatic Identification System (AIS) real world ship data, building a system with the capability to extract representative points of a trajectory cluster. With the trajectory cluster, the study proposes the use of trajectory segmentation algorithms to extract representative points of each trajectory and then use the K-means algorithm to obtain a series of centroids over all the representative points. These centroids combined, form a new representative trajectory of the cluster. The results show a suitable approach with several compression algorithms that are compared with a metric based on the Perpendicular Euclidean Distance.

Keywords: AIS data · Context learning · Data mining · Trajectory clustering · Trajectory compression

1 Introduction

The maritime environment is essential for the current world, being necessary to ensure the safety and security on activities such as the shipment of goods, the fishing or the maritime tourism. The different location systems and technologies can provide useful information about ships operating in an environment, and the artificial intelligence and data mining techniques can be applied to obtain new information useful for the maritime safety.

The work presented in this paper follows a line of research that seeks to extract contextual information from maritime trajectories, studies such as [1, 2] approached the perspective of a trajectory classification with the aim of identifying the type of ship, while the [3] proposal involves the use of clustering techniques on trajectories to extract contextual information usable in various problems.

The objective of the line of research is the detection of anomalous trajectories and as a consequence the identification of possible dangers in the environment of operation. Allowing a quickly reaction in maritime surveillance systems.

For this study the proposal is precisely to take advantage of the information provided by a generated cluster of trajectories. The goal is to generate new context from obtaining a trajectory representing the cluster of trajectories. To do this, it is proposed to calculate representative points of all the trajectories that together compose the waypoints of the

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 166–175, 2023.
https://doi.org/10.1007/978-3-031-18050-7_16

representative trajectory. With this new context information, is possible to apply new data mining algorithms, such as anomaly identification.

To obtain the representative points of each cluster, the proposal is the use of trajectory compression and segmentation techniques to obtain the most important points of each trajectory. With all the points, the process applies a K-means algorithm to obtain the centroids of all the trajectories points, thus obtaining the set of representative points of each cluster.

All the investigation is performed using real world AIS data, so it's necessary to prepare the data before the data mining algorithms are applied. This preparation allows for a cleaning of the trajectories to reduce the noise and to detect wrong information that could affect the data mining results.

To select the trajectory compression algorithm with better results for the proposed problem, a series of algorithms are tested over a specific cluster. For each algorithm different configurations are applied to ensure a different number of representative points.

Then to compare all the different results a distance metric based on the Perpendicular Euclidean Distance is applied, comparing the minimum distance of the points of each trajectory to the segments forming the representative trajectory of the cluster.

This paper is organized as follows: In Sect. 2 the state-of-art methods in the subprocess of data mining data preparation, trajectory clustering and trajectory compression are analysed. In Sect. 3 the system implemented to obtain the compression of trajectories and representative points is presented, while in 4 the results of the proposed clustering approaches are analysed. Finally, the conclusions and future work perspectives are presented in the Sect. 5.

2 Background Information

2.1 Data Pre-processing and Data Imbalance

The real-world data used are Automatic Identification System (AIS) measurements [4], which is a resource that has demonstrated problems (mainly human induced errors [5]) despite the fact that it follows an International Maritime Organisation (IMO) standard [6].To reduce the possible problems a cleaning process is necessary to prepare the data for the following data mining processes.

The most common problems are the class imbalance present in the real-world information (usually addressed by resampling the instances [7]), the noise of the measurements (approached by the use of estimation filters like the Interacting Multiple Model (IMM) filter [8]) and the incorrect and erroneous information which must be solved by studying the data and finding the instances that do not bring quality information to the problem to be solved.

2.2 Trajectory Clustering

In the literature there are several approaches for data mining problems applied over trajectory data, being of special interest the way to transform the data coming from a trajectory into data accepted as an input for the data mining technique.

The trajectories are composed of several consecutive positions in time which is not a suitable input for most data mining algorithms, being necessary to transform the data or adapt the used algorithms. A common proposal is the use of a distance metric to compute a similarity metric usable by the clustering algorithms. For example, the Dynamic Time Warping (DTW) [9] is a proven robust technique that has shown results in previous studies, although it presents a high computational complexity since it operates with all the points of the two trajectories to compare.

One of the most common clustering algorithms in the literature of trajectory clustering is the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [10] and its variations, as an example the proposal in [11] makes a DBSCANSD (DBSCAN adapted to consider Speed and Direction) clustering on maritime routes provided by the IMO [12].

2.3 Trajectory Compression

There are several strategies to process a trajectory compression or segmentation, these strategies usually involve the use of more computational resources to achieve higher quality solutions, therefore it is necessary to evaluate whether to use simpler versions that only consider each point of the trajectory once or to consider complex versions that involve multiple passes through all points of the trajectory. The most common approaches for the trajectory compression are[13]:

- Sequential: simplest processing strategy that analyses the trajectory points in order. The simplest algorithm to perform a trajectory compression is the uniform sampling, a sequential approach that takes the trajectory points at a specific distance (for example take a point every 50 measures).
- Window: algorithms are based on the use of processing windows that group several trajectory points, making the decision over a set of points instead of a single point. Keogh et al. [14] proposed a sliding window approach for time series data. It computes an error in distance to the segment formed by the widow, splitting the window when a specific threshold is reached.
- Graph: algorithm that resolve a graph associated to the trajectory. DOTS (Directed acyclic graph based Online Trajectory Simplification[15]) is an approach that uses a Directed Acyclic Graph (DAG) in real detection time. The DAG nodes represent the trajectory points, while the graph vertices represent the possible compression of the two connected trajectory points.
- Split: algorithms based on a segment division strategy. Making segments between points included in the compression and splitting the segments when adding new points. The most known algorithm of trajectory compression, is the algorithm proposed by Douglas and Peucker in [16](DP).
- Merge: opposite algorithms to the split-based. Starting from a set of all the segments (al trajectory points included in the compressed trajectory), and joining them by removing points from the compressed trajectory. Pikaz and Dinstein [17]performed the first appearance of a Bottom-up algorithm.

- Combination: algorithms that combine different strategies. An interesting approach to this solution is the SQUISH [18] and its improved version SQUISH-E [19], both recent algorithms with great impact in the literature. Combining a merge approach with a sliding window to compute the compressed trajectory.

3 Proposed Architecture

To achieve the proposed problem, the following steps are performed. First, it is necessary to perform a data preparation and clustering process as the one presented in previous studies. Secondly the different trajectory compression algorithms are applied over the obtained cluster to obtain each trajectory representative points.

With those trajectory representative points is possible to extract centroids to represent the trajectory, specifically, the proposal consists of applying the algorithm k-means algorithm is applied to each set of trajectory representative points. This allows a comparison of each of the applied trajectory compression techniques for the proposed problem (Fig. 1).

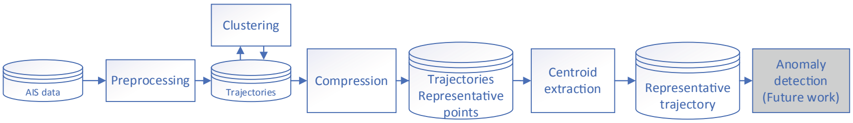

Clustering
AIS data
Preprocessing
Trajectories
Compression
Trajectories Representative points
Centroid extraction
Representative trajectory
Anomaly detection (Future work)

Fig. 1. Representation of the proposed architecture

3.1 Data Preparation and Cluster Generation

The used data consists on raw AIS measures within a day in the Denmark coast without any processing, the proposal needs to use trajectories so the first step of the process is the separation of the data in different trajectories by the use of the MMSI (Maritime Mobile Service Identity) identifier. With the trajectories generation a basic cleaning process is applied, smoothing the noisy measurements with the use of an IMM filter and removing wrong information.

The clustering approach is the one presented in [3] which proposes a DBSCAN algorithm [10] algorithm modified with the DTW algorithm [9] as a distance metric to compare the different trajectories. The DBSCAN algorithm has been chosen because of its simplicity of implementation and its proven effectiveness in other problems with trajectory data, the DTW metric is a technique that allows to find the similarity between two time series and therefore a distance that allow the comparison of two trajectories when needed by the DBSCAN.

This approach has been implemented in Matlab, taking advantage of parallelisation techniques that help in the large consumption of computational resources produced by the selected algorithms.

With this approach the several clusters can be created depending on the configuration for the algorithms. An example is the group of trajectories marked in red for the following figure.

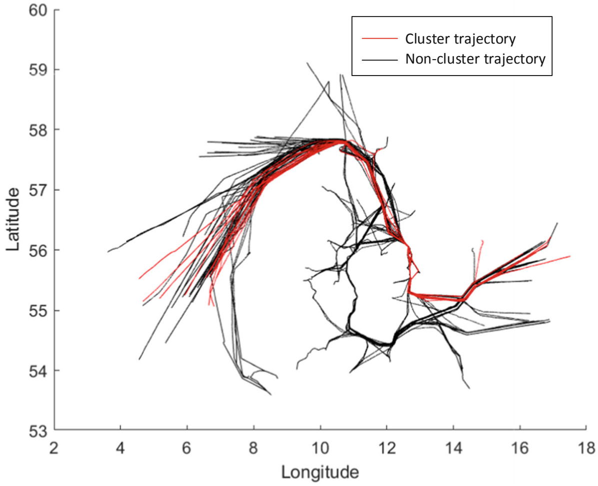


Fig. 2. Cluster of trajectories output

This cluster of trajectories is the one used as input for the following compression of trajectories, obtaining the representative points for all the trajectories marked in red in the previous figure.

3.2 Compression of Trajectories

For the compression of trajectories, several algorithms have been tested. Performing a comparation between the different algorithms to select the best configuration for the problem. Between all the compression algorithms the study considers 3 classic approaches (uniform sampling, opening window, top down) to the compression problem and 2 more advanced approaches (SQUISHE and DOTS).

With the selected approaches the study includes a selection of all the different types of trajectory compression:

- Sequential type through the Uniform sampling, this algorithm has been set up with various cut-off sizes, 50, 500 and 1000 points.
- Window type approach through the opening window and the SQUISHE.

The opening window has been selected using the perpendicular distance as the error function and with several different thresholds: 5, 10, 20, 1000, 2000.

The SQUISH-E variation used for this study, modifies the normal SQUISH algorithm to consider a variable size buffer.

- Graph type through the DOTS algorithm with a threshold value of 50, 100 and 500.
- Split type through the top-down algorithm, using the approach proposed by Douglas and Peucker with the same coniguration as the Openng window algorithm.
- Merge and combination types through the SQUISHE algorithm.

The compression process will generate a series of relevant points of each trajectory, being important to note that depending on the algorithm and its possible configuration the rate of compression could be quite different. Being possible outputs with large set of points along all the trajectory like the results of Fig. 3.(left) or small groups of points in the manoeuvre instants of the trajectory like the results shown in Fig. 3.(right).

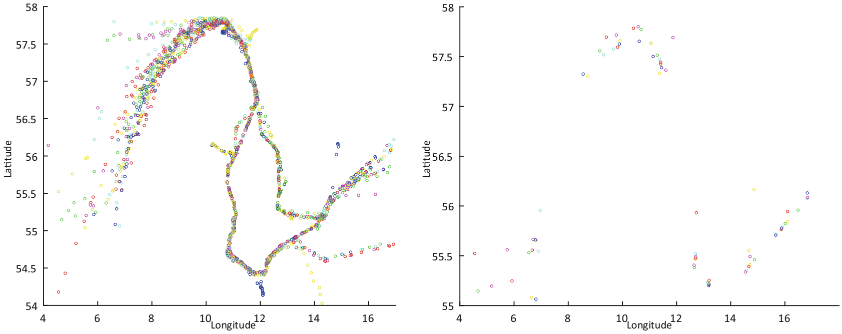


Fig. 3. Compression algorithms example: DOTS with threshold of value 10 (left) and Opening window with threshold of value 10 (right)

3.3 Representative Points Extraction

With the clustering applied and with the compression performed to extract representative points it is necessary to summarise the set of points into a single subset of points that form the representative trajectory of the cluster. In order to achieve this, it is proposed the obtention of centroids that summarizes the different subsets of representative points.

To achieve this, the use of the K-means algorithm is proposed This algorithm allows a simple calculation of centroids in the proposed space, acts quickly and has a proven efficiency.

The centroids of each cluster will be the representative points of the previously performed (see Fig. 2) trajectory clustering. With those representative points a new representative trajectory has been generated.

Once the centroids have been calculated, a metric based on PED distance is proposed to evaluate the different compression algorithms. To do this, the segments of the representative trajectory formed by the centroids are generated. And the minimum distance to these segments is calculated from each point of the different trajectories assigned to the cluster.

With the different distances is possible to calculate an average distance of the trajectories to the representative trajectory. Being possible to make a comparison and result analysis over the different average distances applied in the experimentation.

4 Results Analysis

To test the approximation, different k-means configurations have been implemented to obtain different number of centroids, comparing between the size of 4,5,6,8,10 and 15 centroids. It is noteworthy that, as is natural, the more centroids, the smaller the average distance. However, the goal of the research line makes interesting trajectories that use a smaller number of centroids to summarise the cluster in a smaller number of points.

Comparing the different proposed compression algorithms, the first remarkable property is that the algorithms that use windows such as the opening window or the SQUISHE tend to make segments of a larger size (Fig. 4).

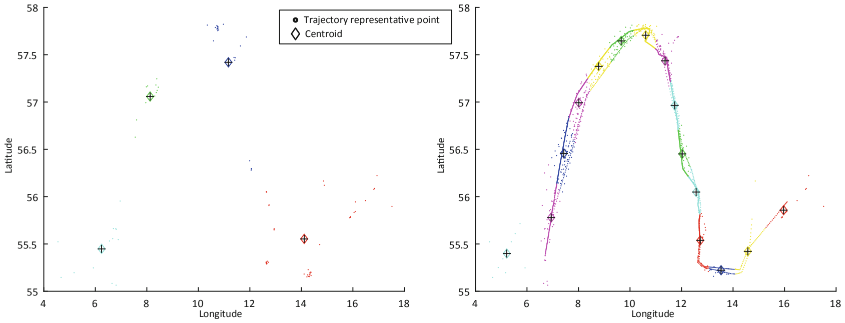


Fig. 4. SQUISHE segmentation with k-means of 4 size (left) and TopDown segmentation with k-means of 15 size (right)

When using the proposed distance metric this difference in segment size is of great interest, since these algorithms show a better result at fewer centroids. While algorithms that generate more segments improve when centroids are added.

As stated above, the more centroids the better the result obtained by all the algorithms, but it is essential to keep the number of centroids small enough to result in an effective summary of trajectories belonging to the cluster.

This implies that selecting the ideal number of centroids is essential to the process, having to adjust it to the chosen segmentation algorithm. If we paint the different results of all the algorithms (Fig. 5.), we can see how algorithms such as SQUISHE or the Top-Down show a better performance in a generalist way (independently of the number of centroids).

If only the sets with a larger number of centroids are evaluated, other algorithms obtain better results. It is therefore important to select an algorithm that fits the desired number of centroids for the representative trajectory.

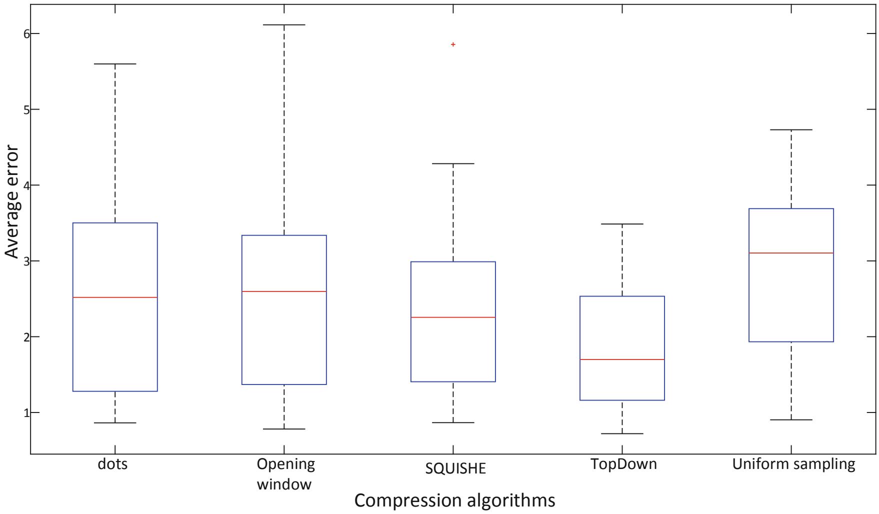


Fig. 5. Box-plotting of the average distance obtained by the different segmentation algorithms

5 Conclusions and Perspectives

During this study, a series of compression algorithms has been tested and compared in the problem of context information extraction of a trajectory clustering. With this method a previous study has been expanded obtaining a series of representative points of a cluster. This provides the capability to implement future data mining processes that take advantage of this context information, an example would be the main proposed future work, which would take advantage of all the information from these representative cluster points to perform anomaly detection, comparing the points with the different cluster trajectories and detecting as anomalies those that are far enough apart.

Regarding the compression processes used, it has been observed by the use of the specific evaluation metric) that the algorithms that creates larger segments (SQUISHE or Top-Down) provide better results in terms of approximation to the trajectories for the solutions with smaller number of centroids. Although this is not necessarily the best, as for greater number of centroids other algorithms obtain improved results.

Therefore, it is necessary to select the compression or segmentation algorithm according to the desired level of compression in the representative trajectory of the cluster.

Acknowledgement. This research was partially funded by public research projects of Spanish Ministry of Science and Innovation, references PID2020-118249RB-C22 and PDC2021-121567-C22 - AEI/https://doi.org/10.13039/501100011033, and by the Madrid Government (Comunidad de Madrid-Spain) under the Multiannual Agreement with UC3M in the line of Excellence of University Professors, reference EPUC3M17.

References

1. Herrero, D.A., Pedroche, D.S., Herrero, J.G., López, J.M.M.: AIS trajectory classification based on IMM data. In: 2019 22th International Conference on Information Fusion (FUSION), pp. 1–8 (2019)
2. Sánchez Pedroche, D., Amigo, D., García, J., Molina, J.M.: Architecture for trajectory-based fishing ship classification with AIS data. Sensors. **20**, 3782 (2020). https://doi.org/10.3390/s20133782
3. Pedroche, D.S., Herrero, D.A., Herrero, J.G., López, J.M.M.: Clustering of maritime trajectories with AIS features for context learning. In: 2021 IEEE 24th International Conference on Information Fusion (FUSION), pp. 1–8 (2021)
4. Enmei, T., Zhang, G., Rachmawati, L., Rajabally, E., Huang, G.-B.: Exploiting AIS data for intelligent maritime navigation: a comprehensive survey from data to methodology. IEEE Trans. Intell. Transp. Syst. **19**(5), 1559–1582 (2018). https://doi.org/10.1109/TITS.2017.2724551
5. Harati-Mokhtari, A., Wall, A., Brooks, P., Wang, J.: Automatic identification system (AIS): data reliability and human error implications. J. Navigation. **60**, 373–389 (2007). https://doi.org/10.1017/S0373463307004298
6. IMO: SOLAS chapter V: Safety of navigation. http://www.imo.org/en/OurWork/facilitation/documents/solas%20v%20on%20safety%20of%20navigation.pdf
7. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002). https://doi.org/10.1613/jair.953
8. Rong Li, X., Jilkov, V.P.: Survey of maneuvering target tracking. part v: multiple-model methods. IEEE Trans. Aerosp. Electron. Syst. **41**(4), 1255–1321 (2005). https://doi.org/10.1109/TAES.2005.1561886
9. Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: AAAIWS'94: Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, pp. 359–370 (1994)
10. Ester, M., Kriegel, H.-P., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD'96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, pp. 226–231 (1996). https://doi.org/10.5555/3001460.3001507
11. Liu, B., de Souza, E.N., Matwin, S., Sydow, M.: Knowledge-based clustering of ship trajectories using density-based approach. In: 2014 IEEE International Conference on Big Data (Big Data), pp. 603–608. IEEE, Washington, DC, USA (2014)
12. International Maritime Organization: Ships' Routeing (2019). ISBN 978-9280100495. https://www.imo.org/en/OurWork/Safety/Pages/ShipsRouteing.aspx
13. Amigo, D., Sánchez, D., García, J., Molina, J.M.: Segmentation optimization in trajectory-based ship classification. In: Herrero, Á., Cambra, C., Urda, D., Sedano, J., Quintián, H., Corchado, E. (eds.) SOCO 2020. AISC, vol. 1268, pp. 540–549. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-57802-2_52
14. Keogh, E., Chu, S., Hart, D., Pazzani, M.: Segmenting time series: a survey and novel approach. In: Last, M., Kandel, A., Bunke, H. (eds.) Data mining in Time Series Databases, pp. 1–21. WORLD SCIENTIFIC (2004). https://doi.org/10.1142/9789812565402_0001
15. Cao, W., Li, Y.: DOTS: An online and near-optimal trajectory simplification algorithm. J. Syst. Softw. **126**, 34–44 (2017). https://doi.org/10.1016/j.jss.2017.01.003
16. Douglas, D.H., Peucker, T.K.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartographica: The International Journal for Geographic Information and Geovisualization **10**(2), 112–122 (1973). https://doi.org/10.3138/FM57-6770-U75U-7727

17. Pikaz, A., Dinstein, I.: An algorithm for polygonal approximation based on iterative point elimination. Pattern Recognit. Lett. **16**, 557–563 (1995)
18. Muckell, J., Hwang, J.-H., Lawson, C.T., Ravi, S.S.: Algorithms for compressing GPS trajectory data: an empirical evaluation. In: GIS '10 (2010)
19. Muckell, J., Olsen, P.W., Hwang, J.-H., Lawson, C.T., Ravi, S.S.: Compression of trajectory data: a comprehensive evaluation and new approach. GeoInformatica **18**(3), 435–460 (2013). https://doi.org/10.1007/s10707-013-0184-0

DR Participants' Actual Response Prediction Using Artificial Neural Networks

Cátia Silva, Pedro Faria(✉), and Zita Vale

GECAD - Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development, LASI - Intelligent Systems Associate Laboratory, Polytechnic of Porto, Porto, Portugal
{cvcds,pnf,zav}@isep.ipp.pt

Abstract. Empowering the consumers will increase the complexity of local communities' management. Enabling bidirectional communication and appliances to become smarter can be a huge step toward implementing demand response. However, a solution capable of providing the right knowledge and tools must be developed. The authors thereby propose a methodology to manage the active consumers on Demand Response (DR) events optimally, considering the context in which it is triggered. The distribution system operator detects a voltage violation and requests a load reduction to the aggregators. In this study, to test a performance rate designed by the authors to deal with response uncertainty, a comparison between requested and actual reduction is done. The proposed methodology was applied to three scenarios where the goal is predicting the response from the consumers using artificial neural networks, by changing the features used in the input.

Keywords: Active consumers · Artificial neural networks · Demand response · Machine learning · Smart grids

1 Introduction

The energy sector is facing changes that will drive toward a more sustainable power and energy use. The growing concern, regarding climate change, introduce distributed generation solutions to deal with the greenhouse effects and air pollution. However, the volatile behavior of these resources requires more flexibility from the demand side. So, the active consumers are empowered and their role in the market is changing. The authors in [1] reinforce the importance of flexibility in the system to enable and promote renewable consumption and reduce the consumers' energy costs while maintaining comfort. Their study presents a Stackelberg game optimization framework for integrated energy systems scheduling by coordinating renewable generations and demand response. However, only the uncertainty from renewable generations is included. The authors in [2] refers those small consumers such as domestic will play a more active role in managing the system and becoming producers of their energy. The simulation results from their study showed that promoting cooperation between the power supplier and the prosumer could lead to significant cost reductions and energy savings. É. Mata et al. [3] also refer

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 176–185, 2023.
https://doi.org/10.1007/978-3-031-18050-7_17

that including flexible behavior can bring several benefits for all the parties: providing service toward energy system stability, security, and cost-effectiveness as well as growing energy awareness for active consumers who participate and provide load reduction.

With this, solutions must be developed to aid the aggregator in the complex task of managing the active communities to be able to send the proper signals to the most trustworthy players. The complexity comes from the different behavior of each resource for each context, increasing the uncertainty of the response. The authors believe that attributing a Contextual Consumer Rate (CCR) will be useful to select the right participants for a DR event since avoiding discomfort from the consumer side and reducing costs from the aggregator perspective. CCR's goal is to characterize the performance of each resource in a DR event, for each context. The present study is a continuation of previous works [4–6], where the goal is to find ways to deal with the DR response uncertainty.

The present paper is organized according to five different sections. The first one is an introduction to the topic with the motivations of the study and innovations from previous works. The following sections present a detailed explanation of the proposed methodology, a case study section, and the results found are analyzed and discussed. Finally, the conclusions withdrawn will be reviewed.

2 Proposed Methodology

To further apply the smart grid concept in the real market, it will be crucial to give active consumers the proper information regarding the market transactions to provide the flexibility to achieve reduction goals. Figure 1 shows the algorithm for the proposed methodology. Considering that Distribution System Operator (DSO), after a power flow analysis where a voltage violation was found, requires a load reduction to all the aggregators associated. With this signal, all of them must trigger a DR event.

With this, it will be possible to identify the proper participants, the ones with higher levels of trust for the context, dealing with the uncertainty. For instance, are all active consumers prepared to participate in the same way and give up their comfort to assure their position in the market? Currently, probably not. Most of them, being new players in the market, have no or insufficient knowledge regarding the actual transactions. Many works in the literature are expecting them to be as always rational and economic players and, in the authors' opinion, this approach may lead to inaccuracies. So, understanding previous behaviors in the same context, and contemplating their availability at the time of the event, can avoid misleading the aggregator perspective when performing the scheduling of the small resources in the community.

CR depends on several factors regarding consumer characteristics: Context Rate (CR), Historic Rate (HR), Last Event Rate (LER), Spatial Rate (SR), and Response Rate (RR). This rate has two phases: the preliminary (PR) - for the selection purposes, and the updated (UR) - for remuneration purposes. The first one is formulated by the sum of CR, HR, SR, and LER, each one with an attributed weight. If a consumer does not have any previous information, for instance, when it is the first time with DR programs, the lowest rate is assigned and must improve the CCR.

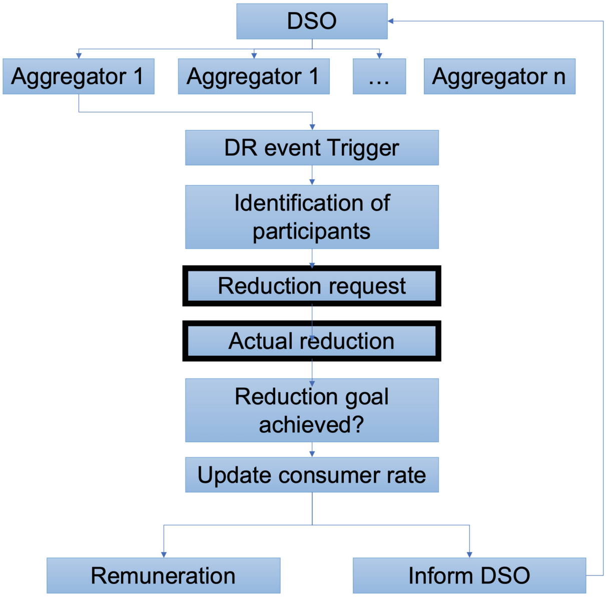


Fig. 1. The proposed methodology, focusing on consumer response.

Understanding each one, CR depends on the availability and the willingness to participate according to the time (ω_{CRP} CRP) and weather (ω_{CRW} CRW) recorded during the period of the event – both have a major influence on consumer response, particularly when distinguishing working days from the weekend or days with extreme temperatures. The formulation can be seen on Eq. 1.

$$\text{CR} = \omega_{\text{CRP}}\text{CRP} + \omega_{\text{CRW}}\text{CRW} \tag{1}$$

Using HR, the Aggregator learns from historical information collected from active consumer and their performance in previous events in similar contexts, according to Eq. 2.

$$\text{HR} = \text{average (previous performances same context)} \tag{2}$$

LER is used to update CCR according to only previous event performance. For instance, a consumer with a higher rate on HR can have a poor LER, which will be important for updating issues.

$$\text{LER} = \text{UR last event in the same context} \tag{3}$$

In the case of SR, for the case where the aggregator has information regarding a voltage violation in a network bus, it will be important to give priority to the ones closer to the local.

The preliminary CCR is formulated according to Eq. 5.

$$\text{PR} = \omega_{\text{HR}} * \text{HR} + \omega_{\text{LER}} * \text{LER} + \omega_{\text{CR}} * \text{CR} + \omega_{\text{SR}} * \text{SR} \tag{5}$$

Moving on to the reduction request phase, a linear optimization of the resources scheduling is done. The objective function aims to minimize operational costs from the Aggregator perspective [7, 8]. Performed the scheduling phase, the reduction is requested to the active consumers. In this step, the innovation from the present paper is presented. Here, the authors intend to predict the actual response from the selected participants as an evaluation of the CCR methodology. The authors opted for the Artificial Neural Networks method. The training database will be composed of the members of the community that already participated in the event, whereas the test database will gather contextual information regarding the current event and will try to predict the response from the active consumers for the event. Should be highlighted that the load requested to be reduced is shifted to another period according to the consumers' preferences.

After, as soon as the reduction request and the actual reduction is compared, the CCR is updated, as mentioned earlier. This updated version is formulated by the sum of CR, HR, SR, LER, and RR, each one with an attributed weight, according to Eq. 6.

$$\mathrm{UR} = \omega_{\mathrm{HR}}\mathrm{HR} + \omega_{\mathrm{LER}}\mathrm{LER} + \omega_{\mathrm{SR}}\mathrm{SR} + \omega_{\mathrm{CR}}\mathrm{CR} + \omega_{\mathrm{RR}}\mathrm{RR} \tag{6}$$

RR represents the performance according to the actual response of the consumer in the current event: if the active consumer responded as requested, the rate would be high, the opposite will also apply, and the active consumer will be penalized with a reduced value. Once the CCR is updated, the remuneration is attributed to the participants, and the DSO is informed of the DR reduction obtained.

3 Case Study

For the present section, the authors intend to test the proposed innovation in the case study, creating different scenarios. For this case, Fig. 2 represents the low voltage distribution network used for the case study, based on a real distributed grid with 236 buses.

The authors believe that context is important, so, it should be necessary for consumers to participate several times in the same context, for different contexts. In this way, to define this dataset, the authors use information from several consumers and the different contexts they participate.

The active consumer availability must be also provided in the input since it is essential to predict the actual response from the participants – several schedules are agreed upon in the DR contract between both parties. As mentioned in the previous section, the DR program applied is load shifting. The selected consumers allow shifting the appliances' schedule – both players agreed on a schedule to control this load, to avoid causing major inconvenience to the participants. Still, an uncertainty factor exists since the consumer can switch on the appliance without further notice – penalties should be applied.

To the ANN algorithm training step, a total of 406 participants in the DR events were used, considering several contexts – both temperature and time factors. Also, to evaluate the importance of the personal data, the authors added two new features and simulated according to the percentages presented in the study done in [9], performing the extrapolation for this case.

The goal of the ANN is to try to predict the actual response from the active consumer, as their expected availability for the context in which the DR event was triggered. The authors wanted to distinguish between both perspectives: with (Scenario 2 and 3) and without (Scenario 1) any information that could probably identify the active consumer. With this, the authors intent to understand if the personal data can lead to better results, since can better identify each consumer.

4 Results and Discussion

Table 1 defines several scenarios where the features are DR period, temperature, day of the week, day of the month, the CCR, age and gender of the participants. The independent rates such as CR, HR, SR and LER were not included as input feature since are already represented by the CCR. To express the age interval and gender with integers, the authors used a label, and their distribution can be seen in Tables 2 and 3. Also, reference [9] was used to extrapolate the percentage of participants within the age interval and gender features.

Table 1. Scenarios defined for the proposed study.

Scenario		Type	1	2	3
Features	Period	Integer	x	x	x
	Temperature	Decimal	x	x	x
	Day of Week	Integer	x	x	x
	Day of Month	Integer	x	x	x
	CCR	Integer		x	x
	Age	Integer			x
	Gender	Integer			x

Table 2. Age interval input definition

Age Interval	[9]	Participants	Label
[20,29]	3.00%	11	1
[30,39]	25.80%	105	2
[40,49]	38.70%	158	3
[50,59]	23.8%	97	4
[60,69]	8.60%	35	5
[70,79]	0.10%	0	6

Table 3. Gender input definition

Gender	[9]	Participants	Label
Female	49.80%	202	0
Male	50.2%	203	1

The three defined scenarios are increasing the level of information since more features are added. The first one does not include any knowledge that could lead to the identification of the active consumer, since privacy problems can be raised – namely since CCR includes the SR. A data preparation step was performed, where all the missing values and categorical data were dealt with. In the "Day of the week" feature, the first day is Sunday, classified as 1. The "Period" feature regards the data gathered on a 15-min basis, where the first period was at 12 PM, and the input dataset has information regarding one month of events.

The implementation of ANN for the present case study was performed using python language and resorting to Google Colab. The libraries for this purpose were pandas, numpy, tensorflow and scikit-learn. As input, the dataset with more features has 6 dimensions and a total of 1.169.274 records per dimension and was withdrawn from previous works by the authors [7], diving the test and train datasets in a 20 to 80 percentage. The authors main goal is to create an ANN capable of predict the active consumer availability to respond to a DR event in a certain context. There was a total of two hidden layers, the batch size was 32 and the number of epochs was 100. For the target value, it was considered that 0 represents a non-response and 1 represents a response.

When importing the dataset, and since the categorical variable were already dealt with, it was time to split into training and testing dataset. Firstly, the authors used the train_test_split function from the scikit-learn library, using a configuration such that 80 percent of data will be there in the training phase and 20 percent of data will be in the testing phase. Also, the feature scaling was performed.

Initializing the ANN by creating an object by using a Sequential class – the input layer. After, the first test is initialized, modifying the number of hidden layers comparing one and two. It is believed that one hidden layer might be enough by many authors in the literature in the Dense class: units and activation. Units stands for the number of neurons that will be presented in the layer and activation specifies. For the present study, rectified linear unit was used as an activation function. Finally, the output layer is created. Should be highlighted that, since this is a binary classification problem – 0 represents a non-response and 1 represents a response, only one neuron it is allocated to the output. So, unit is equal to one. For this final layer, the activation function was sigmoid. With this, it is possible to compile the ANN, where the optimizer used was adam, the loss function was binary_crossentropy, and the performance metrics used was accuracy. As the last step, the fitting process introduces the number of epochs. For this case, one hundred epochs were studied.

So, the results from the case study one, as a training dataset with 80 percent, number of hidden layers equal to one and number of epochs equal to 100 and can be seen in Fig. 2.

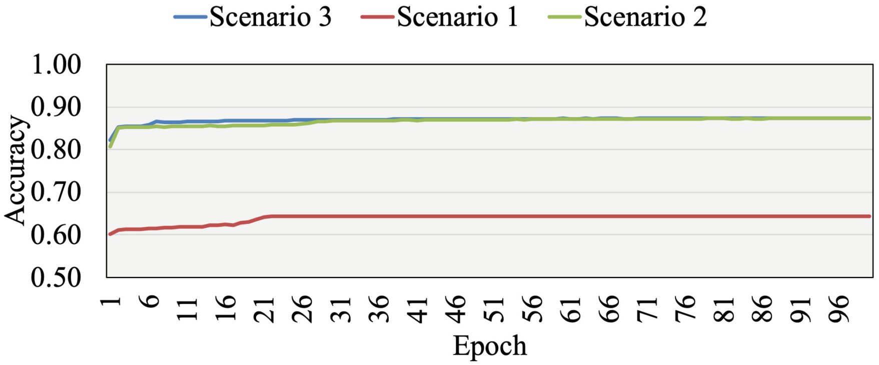


Fig. 2. Case study 1 – scenario accuracy comparison

The time per epoch was rather small for all the data considered, around 20 s, taking a maximum of 34 min per scenario. Regarding the values of accuracy, scenario 1 was the one with a lower value. The remaining had similar behaviour, achieving 87.32% and 87.41%, respectively. Regarding the prediction results, Table 4, presents the comparison for this case study.

Table 4. Case study 1 – prediction vs actual results from ANN

Scenario	Prediction		Actual	
	Responses	Non-responses	Responses	Non-responses
1	115199	0	74.065	41.134
2	78939	36260		
3	71080	44119		

The ANN in scenario 1 predicted that all the participants responded for the contexts in the test. Although only 74.065 actually participated. Regarding scenario 2, the error on the prediction was a total of 4874 records while scenario 3 deviation was around 2985 records.

Moving for the case study 2, where the dataset was divided into 80 percent for training, the number of hidden layers equals to two and number of epochs equals to 100. So, the ANN accuracy for the three different scenarios can be seen in Fig. 3. Again, the time per epoch was rather small for all the data considered, around 20s, taking a maximum of 36 min per scenario.

The scenario 1 was the one with the lowest accuracy. Considering only the outside contexts do not lead to a high value of accuracy since the maximum value (64.35%) was found on epoch 16 and maintained this value until epoch 100. Regarding scenario 2, remembering that the different between the last is the CCR, the accuracy increased around 24.95% for the maximum value. However, this value was only achieved in the

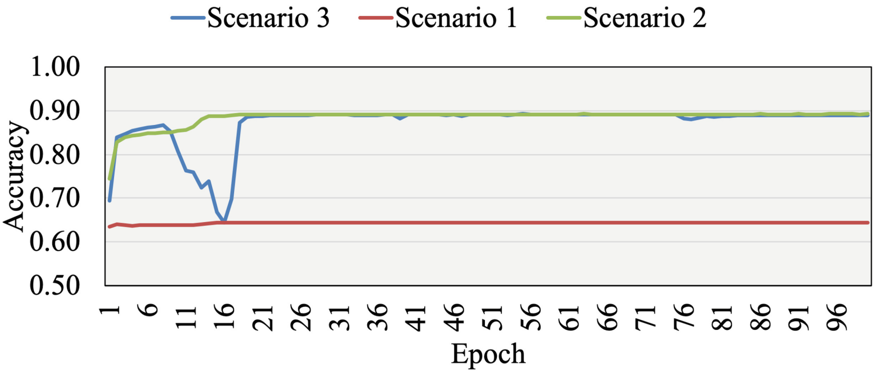


Fig. 3. Case study 2 – scenario accuracy comparison

epoch 86. For the last scenario, the maximum accuracy value was close to scenario 2 (89.29%), yet, it was reached sooner, on epoch 55.

So, in terms of practical numbers, Table 5 presents the comparison between the predicted and the actual values for the trained ANNs.

Table 5. Case study 2 – prediction vs actual results from ANN

Scenario	Prediction		Actual	
	Responses	Non-responses	Responses	Non-responses
1	115.199	0	74.065	41.134
2	80.217	34.982		
3	78.944	36.255		

The test dataset had a total of 115.199 records and, that was the number of responses that scenario 1 predicted – although the number of actual responses was 74.065. For scenario 2, the number of predicted responses was 6.152 above the actual value. While for scenario 3, the predicted responses were below, 4.879 records closer to the actual value. Indeed, having more information regarding the active consumers helped reduce the number of responses that were non-responses. Regarding the number of hidden layers, for scenario 1 there was no difference. For both scenario 2 and scenario 3, two layers had a higher value of accuracy but in the comparison of predicted records, one layer had lower errors.

5 Conclusion

Nowadays, to deal with the impacts of the non-renewable resources regarding climate change, the active consumers must provide flexibility to achieve system balance when the renewable resources do not. Yet, it will take time, education, and resource to make rational decisions as economic players. In this way, the authors propose a Contextual Consumer Rate as a tool to aid the aggregator in choosing the proper participants for a DR event. This rate classifies the participants according to their performance at DR events.

In the case study presented, the idea was to design an ANN capable of predicting the actual response of a participant in a DR event using features such as DR period, temperature, day of the week, day of the month, the CCR, age, and gender of the participants. However, privacy concerns were raised so, three different scenarios were created to understand the need for personal information for the proposed methodology. Also, a comparison between the number of hidden layers in ANN was performed. The scenario, where besides outside context information, the CCR was included had high values of accuracy. Although CCR can provide personal information, regarding to the location of the active consumer, can also give to the aggregator proper knowledge to predict the actual response from the active consumers in the local community with only one feature. Also, according to the results section, after epoch 30 nothing significant happens but, to take this conclusion was needed to test with a higher number of epochs.

Acknowledgments. This article is a result of the project RETINA (NORTE-01–0145-FEDER-000062), supported by Norte Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, through the European Regional Development Fund (ERDF). Cátia Silva is supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with PhD grant reference SFRH/BD/144200/2019. Pedro Faria is supported by FCT with grant CEECIND/01423/2021. The authors acknowledge the work facilities and equipment provided by GECAD research center (UIDB/00760/2020) to the project team.

References

1. Li, Y., Wang, C., Li, G., Chen, C.: Optimal scheduling of integrated demand response-enabled integrated energy systems with uncertain renewable generations: a stackelberg game approach. Energy Convers. Manage. **235**, 113996 (2021). https://doi.org/10.1016/j.enconman.2021.113996
2. Stavrakas, V., Flamos, A.: A modular high-resolution demand-side management model to quantify benefits of demand-flexibility in the residential sector. Energy Convers. Manage. **205**, 112339 (2020). https://doi.org/10.1016/j.enconman.2019.112339
3. Mata, É., et al.: Non-technological and behavioral options for decarbonizing buildings – a review of global topics, trends, gaps, and potentials. Sustain. Prod. Consum. **29**, 529–545 (2022). https://doi.org/10.1016/j.spc.2021.10.013
4. Silva, C., Faria, P., Vale, Z.: Finding the trustworthy consumers for demand response events by dealing with uncertainty. In: 21st IEEE International Conference on Environment and Electrical Engineering and 2021 5th IEEE Industrial and Commercial Power System Europe, EEEIC/I and CPS Europe 2021 – Proceedings (2021). https://doi.org/10.1109/EEEIC/ICPSEUROPE51590.2021.9584667
5. Silva, C., Faria, P., Vale, Z.: Rating the participation of electricity consumers in demand response events. In: International Conference on the European Energy Market, EEM, vol. 2020-September (Sep 2020). https://doi.org/10.1109/EEM49802.2020.9221907
6. Silva, C., Faria, P., Vale, Z.: A Consumer trustworthiness rate for participation in demand response programs. IFAC-PapersOnLine **53**(2), 12596–12601 (2020). https://doi.org/10.1016/J.IFACOL.2020.12.1825

7. Silva, C., Faria, P., Vale, Z.: Rating the participation in demand response programs for a more accurate aggregated schedule of consumers after enrolment period. Electronics (Basel) **9**(2), 349 (2020). https://doi.org/10.3390/electronics9020349
8. Silva, C., Faria, P., Vale, Z., Terras, J.M., Albuquerque, S.: Rating the participation in demand Response events with a contextual approach to improve accuracy of aggregated schedule. Energy Rep. **8**, 8282–8300 (2022). https://doi.org/10.1016/j.egyr.2022.06.060
9. Yoo, S., Eom, J., Han, I.: Factors driving consumer involvement in energy consumption and energy-efficient purchasing behavior: evidence from Korean residential buildings. Sustainability (Switzerland) **12**(14), 1–20 (2020). https://doi.org/10.3390/su12145573

Non-linear Neural Models to Predict HRC Steel Price in Spain

Roberto Alcalde[1](✉), Daniel Urda[2], Carlos Alonso de Armiño[2], Santiago García[2], Manuel Manzanedo[2], and Álvaro Herrero[2]

[1] Universidad de Burgos, Pza. de la Infanta Dña. Elena S/N, 09001 Burgos, Spain
radelgado@ubu.es

[2] Universidad de Burgos, Av. Cantabria s/n, 09006 Burgos, Spain
{durda,caap,lgpineda,ahcosio}@ubu.es, mms0133@alu.ubu.es

Abstract. Steel is a raw material widely used in industry due to its advantages over other alternatives, such as cost, fast and environmentally friendly recycling, ease of use, high strength, different finishes and qualities. Forecasting steel prices has been an important and challenging task that has traditionally been tackled with econometric, stochastic-Gaussian and time series techniques. Advancing from previous work on this open challenge, in the present paper some Artificial Neural Networks are applied for the first time to forecast the price of hot rolled steel in Spain. More precisely, some non-linear neural networks are applied to several different input time series. The target of this research is twofold; on the one hand, identify which of the neural models outperforms the other ones when predicting steel prices and, on the other hand to validate different data series for such prediction. The main outcomes of this research, after validating the neural models on real data from last 7 years, greatly contribute to this field as novel and relevant conclusions are obtained.

Keywords: Neural networks · NIO · NAR · NARX · Forecasting · Steel price

1 Introduction and Previous Work

Steel is an alloy of iron ore with other elements, mainly carbon, but also other minerals to modify its properties. Besides using these minerals, steel can also be produced using steel scrap, making it a highly recyclable material that represents a true circular economy today. The discovery of steel and its subsequent improvement in steel production technology was part of the first and second industrial revolutions, so steel has been essential to the modernisation of the world and continues to be so today.

Global steel production exceeded 1875 million tonnes in 2019, according to the International Steel Association[1], and it is the most energy-intensive industrial sector in the world [1].

Therefore, steel is of great importance for world industry, as it is used in different formats for a large number of industrial sectors (construction, household appliances,

[1] https://worldsteel.org/

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 186–194, 2023.
https://doi.org/10.1007/978-3-031-18050-7_18

vehicles, etc.), and it is a material with an excellent relationship between its physical properties (elasticity, strength, resistance) for its cost. It is also a versatile material that is used in new and varied ways everyday thanks to the development of new alloys and coatings with improved properties [2].

Initially, steel was obtained in a furnace, to which iron ore and coking coat were added to melt these minerals with fire. Air blowing was used to drive the furnace fire to force the combustion gases through and mix the raw materials to obtain pig iron, which is an alloy that melts at a lower temperature than steel. The pig iron is then minced and ground, and mixed with coal in a crucible, which is sealed to prevent oxidation and slowly cooled to obtain the piece of steel, which is then used for forging. Later, various technological advances were made in the process (Bessemer process, Gilchrist-Thomas process, Siemens-Martin), to produce large quantities at low cost. Today, electric mini-furnaces are also used to produce steel from scrap (ferrous scrap and iron ore fines Fe 62%). There are two different routes in the steelmaking process, the primary or BOS (Basic Oxygen Furnace), which mainly uses iron ore and coal, and the secondary route or EAF (Electric Arc Furnace), which uses steel scrap [3].

A distinction can be made between different types of steel depending on their chemical composition (carbon, stainless, electrical and special steel), their physical form (coil, flat and long steel), their processing stage (hot-rolled, cold-rolled, galvanised steel), as well as their finish, which means that steel is not a homogeneous product [3].

The price of different types of steel depends on many factors, such as raw materials (iron ore, coal), production process, production capacity, investments, energy costs, demand, supply, political factors [4]. In addition, it appears that there may be some correlation between the price of steel and the share price of the steelmaking companies [5]. On the other hand, there are studies that suggest a one-way correlation between some variables, such as the price of oil influencing the price of steel, but not vice versa [6]. And, metal prices have been considered as relevant indicators for markets due to their ability to adjust to macroeconomic and speculative conditions [7].

Due to the widespread use of steel in many sectors, the price of steel is a major factor influencing the competitiveness of many companies. Therefore, predicting price trends can help companies to make decisions on buying and selling steel in order to increase their profitability.

In general, the prices of steel products all follow the same pattern, but Hot-rolled Sheet Coil (HRC) is usually considered as the benchmark because it accounts for the largest volume of steel exports [8].

Although the steel market is global, the price of steel can differ from country to country, due to different reasons, such as tariffs, market demand, taxes, among others. Nevertheless, the evolution of steel prices is parallel in each country [9–11]. In addition, the price of commodities, including steel, seems to experience relatively different correlated fluctuations for each country, which may be due to similar international terms of trade [12].

For the prediction of steel price, multiple models have been used, such as the Auto-Regressive Integrated Moving Average (ARIMA) model [13, 14], as well as Long Short-Term Memory LSTM [15]. On the other hand, the applied non-linear neural networks have been previously applied to some other problems such as logistics [16] or precision agriculture [17] among others. Differentiating from this previous work in the present paper some non-linear neural networks are applied to this problem for the first time. Additionally, novel data series are used as the input data for such predicting models.

This study focuses on the prediction of the HRC price for Spain, and it includes a comparison of several neural network models (Described in Sect. 2), namely: Non-linear Input-Output (NIO), Non-linear Auto-Regressive (NAR), and Non-linear Auto-Regressive with Exogenous Input (NARX)). Such supervised-learning models are applied for steel price time series forecasting. Furthermore, the impact of the parameter tunning for these models is also assessed. Additionally, some different data series (described in Sect. 3) are studied and compared in order to identify which one contributes to the most precise prediction.

2 Materials and Methods

In the present section both the used data (Subsect. 2.1) and the applied models (Subsect. 2.2) are introduced.

2.1 Dataset

As previously mentioned, the present research focuses on predicting the price of hot-rolled sheet coil (HRC) for Spanish market. The source for this data is CRU-Commodities Research Unit[2] which is a leading independent supplier of steel market information. In order to predict the price, three groups of data are studied:

- Series 1: Stock market prices of three major global steel producing companies, which are Posco, ArcelorMittal and NipponSteel, whose data source is NYSE – Nasdaq Real Time Price in USD[3].
- Series 2: Economic situation in Spain, including data related to three issues: a) Price (Index 2010 = 100): Consumer Price Index Harmonized, Consumer Price Index All items, Producer Price Index All Commodities Index; b) Economic Activity- Industrial Production (Index 2010 = 100): Total Index, Manufacturing Index, Mining Index; c) External Trade Goods (US Dollars): Value of Imports (Cost, Insurance, Freight CIF), Value of Exports (Free on Board FOB). (Data source from International Monetary fund[4]).
- Series 3: Price of other commodities: price of Aluminum; Brent Crude; Cobalt; Copper; Dubai Crude; Gold; Lead; Molybdenum; Silver; Zinc (Data source from International Monetary fund).

The data series for all features corresponds to a monthly periodicity during the period 2013–2019. A monthly periodicity is used for the data source as this is the time unit generally established in steel price indexation and is sufficient to reflect the volatility of the steel price [18]. As a sample of the target data to be forecasted, the maximum price of HRC in the series under analysis was identified in.

[2] https://www.crugroup.com/

[3] https://finance.yahoo.com/

[4] https://www.imf.org

2.2 Non-lineal Neural Models

In order to predict the price of HRC steel, the data described in previous subsection has been used as the input data for several neural models for non-linear time-series forecast [19], namely: Non-linear Input-Output (NIO), Non-linear Autoregressive (NAR) and Non-linear Autoregressive with Exogenous Input (NARX). All these three can be defined as feedforward Neural Networks (NN) that includes a tap delay associated to the input weight. The models have a finite dynamic response to time series input data thanks to such delay.

Although these three models have similar characteristics, there are also some differences, being mainly the data the model is provided with. Firstly, there is the NIO models that tries to predict a data series only taking into account previous values of other data series but not the target one. That is, the data series to be predicted is not used itself as an input. Oppositely, the NAR model, as stated in its name just models a relation between previous values of the data series whose future values is trying to predict. Hence, only the series to be predicted is used by this model. These two approaches are combined in the NARX model, as it predicts the values of a data series by using the previous values of it (endogenous input) while considering some other data series (exogenous inputs).

In a more formal, way, these models can be mathematically defined as follows, considering y(t) the feature to be predicted in time instant t and $f()$ the function to be approximated by the time-series forecasting model. Mainly consisting of a model that predicts a series (y) only using n previous values of it, the NAR model can be expressed as:

$$y(t) = f(y(t-1), \ldots y(t-n_y)) \tag{1}$$

From an opposed perspective, the NIO model tries to predict a series (y) using n previous values of a different one (x):

$$y(t) = f(x(t-1), \ldots, x(t-n_x)) \tag{2}$$

Finally, the NARX is a model that combines previous values of both the series to be predicted (y) and those from a different one (x) that is considered the exogenous one:

$$y(t) = f\left(y(t-1), \ldots, y\left(t-n_y\right), x(t-1), \ldots, x(t-n_x)\right) \tag{3}$$

3 Experiments and Results

This section presents information about the performed experiments, as well as the obtained results. When defining the experiments and based on previous results, the parameters of these models has been tuned with combinations of the following values:

- Training algorithm: {1 – Levenberg-Marquardt, 2 – Batch Gradient Descent, 3 – Gradient Descent with Momentum, 4 – Adaptive Learning Rate Backpropagation, 5 – Gradient Descent with Momentum and Adaptive Learning Rate, 6 – Scaled Conjugate Gradient, 7 – Broyden–Fletcher–Goldfarb–Shanno Backpropagation}

- Number of hidden neurons: {1, 5, 10, 15, 20}
- Number of input delays: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
- Number of output delays: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

In order to obtain more statistically significant results, 10 executions have been performed for the same configuration of the parameters. Due to the combination of all these values, 350 executions have been carried out for the NIO and NAR models per each one of the data series. In the case of NARX, as it combines a varying number of both input and output delays, 3.500 executions have been carried out per each one of the data series.

The averaged Mean Squared Error (MSE) for the 10 executions is provided in this section. In each one of the tables, the lowest error value per column is in bold.

Initially, the averaged results obtained by the time-series models are presented per the number of input delays in Table 1.

Table 1. MSE of the results obtained by the NAR, NIO, and NARX neural models. Averaged results for the 3 data series are shown per the number of input delays.

N input delays	NAR	NIO	NARX
1	44777.62	34768.56	**30731.95**
2	36811.43	33635.41	1259633.09
3	35092.46	36092.17	31739.07
4	37317.69	**30805.89**	31725.67
5	35941.02	34324.44	34231.69
6	32765.64	34953.35	34211.34
7	29586.15	35951.07	34099.02
8	35067.41	42498.11	36024.10
9	35714.85	41546.65	37190.99
10	**27439.20**	39195.92	39736.90

From the results in Table 1, it can be said that opposed results have been obtained for some of the models: in the case of NAR, the lowest error is obtained with the highest number of input delays (10) while in the case of NARX it has been obtained with the lowest one (1). In the case of NIO, an intermediate value (4) for input delays is the one with the lowest error.

Results obtained by the neural models are also presented per the number of hidden neurons in Table 2.

When considering the number of neurons in the hidden layer of the neural models, opposing results are found: when only the price data series is used (NAR model), the lowest error is obtained with the biggest hidden layer (20 neurons). However, when the endogenous data series are used (NIO and NARX models), the lowest error is obtained with the smallest hidden layer (1 neuron).

Table 2. MSE of the results obtained by the NAR, NIO, and NARX neural models. Averaged results for the 3 data series are shown per the number of hidden neurons.

Neurons	NAR	NIO	NARX
1	35092.46	**10662.66**	**10453.78**
5	36205.07	21073.80	20342.18
10	34353.33	33933.62	646951.91
15	32326.78	52494.22	47046.96
20	**31577.02**	63721.49	59867.09

In Table 3, the obtained results are presented per the training algorithm.

Table 3. MSE of the results obtained by the NAR, NIO, and NARX neural models. Averaged results for the 3 data series are shown per the training algorithm.

Training algorithm	NAR	NIO	NARX
1	2029.26	12615.28	11009.04
2	115329.59	94843.10	947446.69
3	117558.44	95041.44	89026.89
4	3321.75	15365.87	12867.00
5	3845.27	13214.13	16612.29
6	**1448.84**	**11720.95**	10919.50
7	1826.28	11839.34	**10645.26**

As it can be seen in Table 3, there are strong differences regarding the errors obtained when training the models with different algorithms, being the best and worst results clearly identified. The two algorithms associated to best results (smallest error) are “Scaled Conjugate Gradient” (6) and “Broyden–Fletcher–Goldfarb–Shanno Backpropagation” (7) for all the three models. On the other hand, the worst results (highest error rates) are associated to the “Batch Gradient Descent” (2) and Gradient Descent with Momentum (3) for all the three models.

As one of the targets of the present research is identifying the best time series for predicting the price, averaged results are also presented in Table 4 per the used data series (only NIO and NARX models as the NAR model is only provided with the price itself).

When only using the endogenous data series to make the prediction (NIO), it is the data series 3 (Price of other commodities) the one associated to best results and the data series 1 (Stock market prices) the second best. When combining the endogenous data series with the price itself (NARX), it is the other way round: the data series 1 is the one associated to best results and the data series 3 is the second best. For both models,

Table 4. MSE of the results obtained by the NIO, and NARX neural models. Averaged results are shown per the data series.

Data series	NIO	NARX
1	29775.95	**26434.71**
2	52250.05	417830.09
3	**27105.46**	26532.35

the data series 2 (Economic situation in Spain) is the one associated to worst results. Regarding the neural model that is applies, it can be said that each one of the data series, the best result is always obtained by the NARX model, being NIO in all cases the second best.

Finally, in order to clearly identify the best single results, Table 5 presents the lowest error rates for each parameter combination. In this case, the error has been averaged only for the 10 executions run with exactly the same values for the different parameters.

Table 5. MSE of the results obtained by the NAR, NIO, and NARX neural models. Best single results (averaged only for the 10 executions) are shown.

Input data series	NAR	NIO	NARX
Price itself	**566,63**	**-**	**-**
1	**-**	789,75	**460,41**
2	**-**	1150,34	796,86
3	-	**664,81**	582,63

For this individual results (error is not averaged for all the parameter combinations), similar results to those in Table 4 can be seen: the best result is obtained with data series 3 in the case of the NIO model while it is the data series 3 in the case of the NARX models. In both cases, the worst results are obtained with data series 2. Regarding the neural models, for all the endogenous data series, NARX clearly outperforms NIO in all cases.

All in all, the best single result (10 executions) is obtained when combining the NARX model and the data series 1. The second best is obtained by the NAR model, hence using only the price data series.

4 Conclusions and Future Work

This study has presented empirical results obtained with three datasets (stock market, economic and other commodity) on the performance of several non-linear models (NIO, NAR, NARX) for steel price forecasting. The performance of the different neural models and their parameters has been compared. The used models have not been previously applied for steel price forecasting.

After analyzing the individual results, it can be concluded without any doubt that best predictions of the HRC steel price are obtained with the NARX model and data series 1, showing superior performance to the other models and data series. So, it can be concluded there is a strong relationship between the share price of steel companies and the price of steel.

From the perspective of parameter tuning, it can be concluded that for some of the parameters there is not a consensus affecting all the models. The best values for the number of input delays and hidden neurons varies from one model to the other one so it must be adjusted case by case. However, in the case of the algorithm used to train the model, it can be concluded that the "Scaled Conjugate Gradient" and "Broyden–Fletcher–Goldfarb–Shanno Back-propagation" are the most advisable ones for all the three models.

When considering the input data series in general terms, the "Price of other commodities" and "Stock market prices" are those associated to best results. Worst results have always been obtained when using the "Economic situation in Spain" data series.

From a business perspective, the result obtained shows that the stock market performance of the shares of steel producing companies is correlated with the evolution of the price of steel. This connection is consistent with the effect that increases in the selling price of a company's material, with equal demand, have a positive influence on the profitability of the company, and therefore the value of the share is increased by investors.

In future studies, new soft-computing techniques and other data series can be used to improve the existing predictive models.

References

1. Conejo, A.N., Birat, J.P., Dutta, A.: A review of the current environmental challenges of the steel industry and its value chain. J. Environ. Manag. **259** (2020). https://doi.org/10.1016/j.jenvman.2019.109782
2. Gutierrez, J.P., Vianna, A.C.: Price effects of steel commodities on worldwide stock market returns. North Amer. J. Econ. Fin. **51**(April) (2020). https://doi.org/10.1016/j.najef.2018.09.007
3. European Commission: Towards competitive and clean European steel (2021)
4. Malanichev, A.G., Vorobyev, P.V.: Forecast of global steel prices. Stud. Russ. Econ. Dev. **22**(3), 304–311 (2011). https://doi.org/10.1134/S1075700711030105
5. Manu, M.V., Brătășanu, V., Vasile, I.: Challenging the status quo: steel producer case study on the enterprise value for M&A. Manag. Dyn. Knowl. Econ. **7**(2), 207–228 (2019). https://doi.org/10.25019/mdke/7.2.05
6. Ming-Tao Chou, S.-C.C., Yang, Y.L.: Review of Economics & Finance A Study of the Dynamic Relationship between Crude Oil Price. Better Advances Press, Canada in its Journal Review of Economics & Finance., vol. 2, no. May (2012)
7. Omura, A., Todorova, N., Li, B., Chung, R.: Steel scrap and equity market in Japan. Resour. Policy **47**, 115–124 (2016). https://doi.org/10.1016/j.resourpol.2016.01.001
8. Rodríguez Liboreiro, P.: Competencia, rendimientos crecientes y exceso de capacidad: la industria siderúrgica mundial (2000–2014). Cuadernos de Economía **38**(76) (2019). https://doi.org/10.15446/cuad.econ.v38n76.61257

9. Gutierrez, J.P., Vianna, A.C.: Price effects of steel commodities on worldwide stock market returns. North Am. J. Econ. Fin. **51**(Oct) (2020). https://doi.org/10.1016/j.najef.2018.09.007
10. Pauliuk, S., Wang, T., Müller, D.B.: Steel all over the world: Estimating in-use stocks of iron for 200 countries. Resour. Conserv. Recycl. **71** (2013). https://doi.org/10.1016/j.resconrec.2012.11.008
11. Panasiyk, D., Laratte, B., Remy, S.: Steel stock analysis in Europe from 1945 to 2013. Proc. CIRP **48**, 348–351 (2016). https://doi.org/10.1016/j.procir.2016.04.084
12. Adewuyi, A.O., Wahab, B.A., Adeboye, O.S.: Stationarity of prices of precious and industrial metals using recent unit root methods: implications for markets' efficiency. Resour. Policy **65**(Jul), 2020 (2019). https://doi.org/10.1016/j.resourpol.2019.101560
13. Zola, P., Carpita, M.: Forecasting the steel product prices with the arima model. Stat. e Appl. **14**(1) (2016)
14. Adli, K.A.: Forecasting steel prices using ARIMAX model: a case study of Turkey. Int. J. Bus. Manag. Technol. **4**(5), 62–68 (2020)
15. Cetin, K., Aksoy, S., Iseri, I.: Steel price forcasting using long short-term memory network model (2019). https://doi.org/10.1109/UBMK.2019.8907015
16. Alonso de Armiño, C., Manzanedo, M.Á., Herrero, Á.: Analysing the intermeshed patterns of road transportation and macroeconomic indicators through neural and clustering techniques. Pattern Anal. Appl. **23**(3), 1059–1070 (2020). https://doi.org/10.1007/s10044-020-00872-x
17. Yartu, M., Cambra, C., Navarro, M., Rad, C., Arroyo, Á., Herrero, Á.: Humidity forecasting in a potato plantation using time-series neural models. J. Comput. Sci. **59** (2022). https://doi.org/10.1016/j.jocs.2021.101547
18. Guo, S., Li, H., An, H., Sun, Q., Hao, X., Liu, Y.: Steel product prices transmission activities in the midstream industrial chain and global markets. Resour. Pol. **60**(Nov), 56–71 (2019). https://doi.org/10.1016/j.resourpol.2018.11.014
19. Leontaritis, I.J., Billings, S.A.: Input-output parametric models for non-linear systems Part I: Deterministic non-linear systems. Int. J. Control **41**(2) (1985). https://doi.org/10.1080/00207178508961129

Soft Computing Applications

First Steps Predicting Execution of Civil Works from Georeferenced Infrastructure Data

Baterdene Batmunkh[1], José Antonio Chica Paez[1], Sergio Gil Lopez[1], Maider Arana Bollar[1], Oihana Jauregi Zorzano[1], Andoni Aranguren Ubierna[1], Manuel Graña[2], and J. David Nuñez-Gonzalez[2](✉)

[1] TECNALIA, Basque Research and Technology Alliance (BRTA), San Sebastian, Spain

[2] Computational Intelligent Group, University of Basque Country (UPV/EHU), Leioa, Spain

{manuel.grana,josedavid.nunez}@ehu.eus

Abstract. Geospatial data treatment is an important task since it is a big part of big data. Nowadays, geospatial data exploitation is lacking in terms of artificial intelligence. In this work, we focus on the usage of an machine learning models to exploit a geospatial data. We will follow a complete workflow from the collection and first descriptive analysis of the data to the preprocess and evaluation of the different machine learning algorithms. From unload dataset we will predict if the unload will lead to civil work, in other words, it is a classification problem. We conclude that combining machine learning and geospatial data we can get a lot out of it.

1 Introduction

In recent years, the amount of geospatial data has grown and will grow exponentially according to the U.S. National Geospatial Intelligence Agency. That is why the traditional treatment of this information has become completely obsolete, both in terms of computing capacity and exploitation of knowledge. Thus, we must move towards an information analysis methodology that delegates tasks to computational intelligence.

The objective of this project is the combination of an artificial intelligence models and geospatial data to enrich and exploit this data. Therefore, a complete workflow will be followed from the collection and first descriptive analysis of the data to the preprocess and evaluation of the machine learning models.

To be precise, we will have a dataset of all the unloads carried out since 2010 from two cities. First we are going to clean this dataset and apply some methods such as outlier detection, feature selection, etc. to get a certain dataset adapted to our project, machine learning algorithms.

Furthermore, we will have another dataset of construction works done in those two cities. However, due to some problems there is still no data to validate

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 197–207, 2023.
https://doi.org/10.1007/978-3-031-18050-7_19

our proposal, so we built some models to generate a synthetic dataset with some degree of arbitrary complexity, so that if our approach succeeds on this dataset, it can be useful when we acquire the dataset.

Then, we will associate the construction work dataset to the unload dataset and select possible unloads that led to construction work. Therefore, it is a classification problem which identifies the unloads that led to construction work.

Finally, we will use this prediction to take actions on some issues, such as financial aid, take advantage of the area to do similar construction works, etc.

1.1 State of the Art

Over recent years, the exploitation of geospatial data has been of great importance, since a significant part of big data is actually geospatial data, and the size of such data is rapidly growing by at least 20% every year as it says in [1]. This exploitation benefits in fuel and time savings, increased income, urban planning, medical care, etc. On the other hand, geospatial data is important for Earth observation, geographic information system/building information modeling (GIS/BIM) integration and 3D/4D urban planning [2]. The general concept to analyze GIS and BIM data structures and spatial relationship will be of great importance in emergin applications such as smart cities and digital twins [3].

In last seven years there have been several projects related to geospatial data and artificial intelligence.

In 2015 [4] support vector machine (SVM) and coactive neuro-fuzzy inference system (CANFIS) algorithms were tested to predict crash severity in a regional highway corridor and discover spatial and non-spatial factors that are systematically related to crash severity. Also, a sensitivity analysis is carried out to determine the relative influence of the crash. In 2017 [5] a GIS based flood modeling for Damansara river basin in Malaysia. The frequency ratio method was combined with SVM to estimate the probability of flooding. The flood hazard map was produced by combining the flood probability map with flood triggers such as daily rainfall and flood depth. The approach of this project would be effective for flood risk management in the study. In 2018 [6] used machine learning to address the challenge of layers in geospatial data. The fundamental hurdle in geospatial data is identifying what number of feature levels is necessary to represent user's multidimensional preferences by considering semantics, such as spatial similarity and metadata attributes of static data sets. In 2019 [7] a modeling combining the LogitBoost classifier and decision tree and geospatial data from multiple sources were used for the spatial prediction of susceptibility to tropical forest fires. This project is necessary for disaster management and a primary reference source in territorial planning. SVM, random forest (FR) and kernel logistic regression (KLR) were used as benchmarks. En 2020 [8] a project to facilitate planning efforts using multitude of tightly interlocked component measured by new sensors, data collection and spatio-temporal analysis methods. With geospatial data and urban analysis understand urban dynamics and human behavior for planning to improve livability. Moreover, [9] in India random forest model to produce exposure maps of the areas and populations potentially exposed to high arsenic concentrations in groundwater. Finally [10] using AI-based techniques for 3D point clouds and geospatial digital twins as generic component of geospatial AI. 3D point clouds can be seen as a corpus with similar properties as natural language corpora and formulate a ‘naturalness hypothesis’ for 3D points clouds.

1.2 Research Proposal

The proposal of this project is the combination of the geospatial data and machine learning models to get the most out of this concrete geospatial data, for example, show the construction work predictions to the town hall, thus they can take action sooner, etc. Therefore, the main focus of this project is to classify the unload dataset if they will led to construction work.

Since, the steps taken to do construction works and the features of this datasets are quite similar in different areas, this work can be applied in different cities for same purpose.

2 Methodology

In this section, we will describe the database and the steps taken to develop this project .

To begin with, we start with a dataset of construction unloads done in a city since 2010. This dataset has 2859 unloads and 14 features (Table 1).

Table 1. Features: on the left feature names and on the right feature description.

REQUEST_ID	Unload identification, unique integer values
NAME	Name of the enterprise that unloads the materials
USER	User that records the information of the unloads on a database
DATE	Date when the unload is done: YY-MM-DD HH:MM:SS.MMM
CENTROID	Geographic centroid of the unload: 123456,1234567
MUNICIPALITY	Municipality where the unload is done
TYPE	Unload type: ’cultive’ (3); ’work’ (1932); ’minor work’ (18); ’project’ (906)
PROMOTER	Name of the company that has requested the unload
CIF	CIF of the enterprise that unloads
CLIENT_ID	Identification in the database of the enterprise that unloads the materials
ACTIVITY	The activity carried out by the enterprise that unloads the materials
AREA	The area it covers from the centroid in hectares
COST	Unload cost in euros: two decimal places for cents
SERVICE_CLASS	The purpose of the unload: ‘canalization’ (1946); ‘edification’ (146); ‘civil work’ (423); ‘urbanistic planning’ (5); ‘urbanization’ (160); ‘others’ (175)

2.1 Preprocess

After analyzing the dataset, we provide all preprocessing done to have certain dataset for machine learning algorithms (Table 2). First of all, feature selection, some features of the unload dataset are meaningless:

Table 2. Meaningless features: on the left feature names and on the right the reason why it is meaningless.

REQUEST_ID	We can identify the unloads by integer sequence beginning by 1
NAME	We can identify the enterprises by their CIF, how they name it is meaningless for machine learning algorithms
USER	Users that records the information do not affect the result
MUNICIPALITY	Municipality where the unload is done do not affect the result. Moreover, it has static value, therefore the standar deviation is 0
CLIENT_ID	We will use CIF to identify the enterprises that unloads the materials

On the other hand, some features are better represented by other ways (Table 3).

Table 3. Features that can be represented better.

DATE	We represented the date yearly and monthly. In future we may represent daily too. Thus the dataset has two more features: YEAR; MONTH
CENTROID	Geographic points are better represented by X and Y.

After previous changes, the new dataset has 11 features: ‘TYPE’; ‘PROMOTER’; ‘CIF’; ‘ACTIVITY’; ‘AREA’; ‘COST’; ‘SERVICE_CLASS’; ‘X’; ‘Y’; ‘YEAR’; ‘MONTH’.

Over this dataset, we changed the values of the feature ’PROMOTER’, since there were values that meant the same but it was written differently. For example, ‘*city hall*’ and ‘*city hal*’.

Furthermore, atypical or abnormal observations potentially affect the estimation of parameters. There are different ways to deal with atypical observations, but we decided to remove them. For this task, after analyzing the dataset we perceived that some points were out of the range of the city, thus this points would be classified as atypical observations, then to be removed.

2.2 Data Analysis

Once the cleaning was done, we analyzed the dataset associating the features between them and drew some conclusions.

To begin with, we did temporal analysis using the features ‘YEAR’ and ‘MONTH’ to see the temporal unload behavior. Analyzing it yearly we realized that this city had uptrend since 2010 (100 unloads) until 2021 (400 unloads). Hence, it can be expected that in 2022 there will also be quite a few unloads. Likewise, analyzing it monthly we can notice the deterioration in August and December, since these months are holidays, and between these two months there is a rise and a fall.

Then, we did temporal analysis of the feature ‘TYPE’ and realized that the uptrend is due to the unload type *project*. Doing it monthly, follows the trend that we have said before, deterioration in August and December, and between them a rise and a fall (Fig. 1).

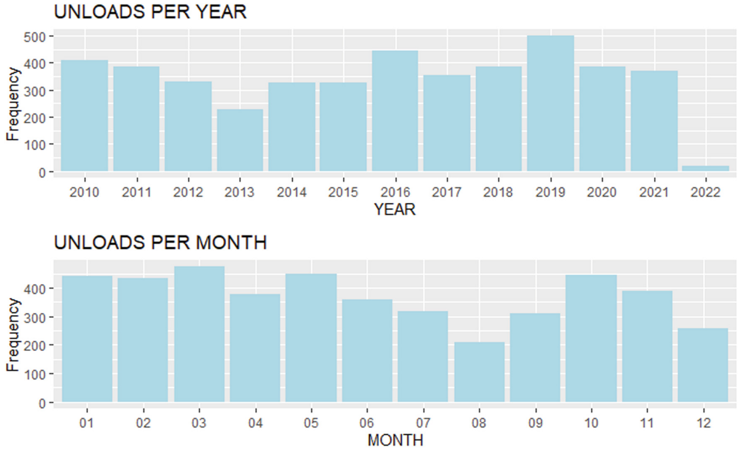


Fig. 1. Data analysis: unloads per year and unloads per month.

Also, when we did temporal analysis of the promoter, the city hall promoted lot more in years of covid compared to the other years, since in 2020 and 2021 promoted 120 unloads each year and the maximum of the other years do not surpass 60.

Finally, unloads that leads to urbanization and civil work has uptrend temporarily, while canalization and the others had different distributions.

Furthermore, we did associate different features and drew some conclusions as the city hall focus more on canalization and civil work, the city hall contacts more with enterprises that are contractor, the area it covers depending on unload type, etc.

After analyzing the dataset we decided to cluster by geographical points. This way we could analyze the changes depending on clusters.

We used two different methods for the clustering, therefore, we have two more features, one for each cluster:

- Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [11]: Unsupervised learning algorithm for clustering. This algorithm uses density to cluster the data points. Initially this algorithm classifies the points into three categories: Core points, Border points and Noise points. Core points must have equal or greater than minimum neighbors. Border points has less than minimum neighbors and the point should be in the neighborhood of a core point. Lastly, noise points are points that are neither a core point nor a boundary point. Once classified the points, if two core points are neighbors they are linked by a density edge and are called density connected points. Finally, it discards noise, assign cluster to a core point, color all the density connected points of a core point and color boundary points according to the nearest core point.
- Model-Based Clustering [12]:
  Statistical approach to data clustering. The fit between the given data and some mathematical model, and is based on the assumption that data are created by combination of a basic probability distribution. Initially assigns k cluster centers randomly and iteratively refines the cluster based on two steps: Expectation step and maximization step.

Then we analyzed the cost depending on clusters and saw that the cost do not change depending on where it unloads.

2.3 Dataset Generation

Once the unload dataset is cleaned, we have to associate with the construction works dataset to add another feature to the unload dataset that represents the unload has led to a construction work. However, as there is still no data of construction works done, we have had to generate a synthetic feature following some reasoning. So, if our approach succeeds, it can be successful in real life experiments.

First, we added a new binary variable 'WORK' to the unload dataset which represents that the unload has led to construction work or not (Fig. 2. Then, depending on the features we set 1 value on some unloads:

- TYPE: Set 1 on 90% of the minority classes, that is, those that appear less than 40 times.
- PROMOTER: From the 'city hall' (553) and 'company name' (544) to %60 we have set 1, since these values have risen in recent years. Therefore, we have reasoned that their licenses were being accepted. Moreover, we have given more importance to 'city hall', since we thought that 'city hall' projects has more possibility to be accepted.
- ACTIVITY: From the 'city hall' and 'engineering' to %60 we have set 1. Here we have followed same reasoning: uptrend and city hall.
- SERVICE_CLASS: From 'urbanization' and 'civil work' to %70 we have set 1.

- From other features we could not get reasonable things, so from the rest of the 0 values to the 15% we have set 1 randomly.

The unload dataset need a general review, since we just added one variable value, that is, for one construction work there can be more than one unload, so if one of those unloads has 1 other unloads have to have 1. This review was carried out by checking if some features are equals and the centroid point is in a certain radius of the unload centroid that led to construction work. After this review we got 1245 instances for class 0 and 1487 instances for class 1.

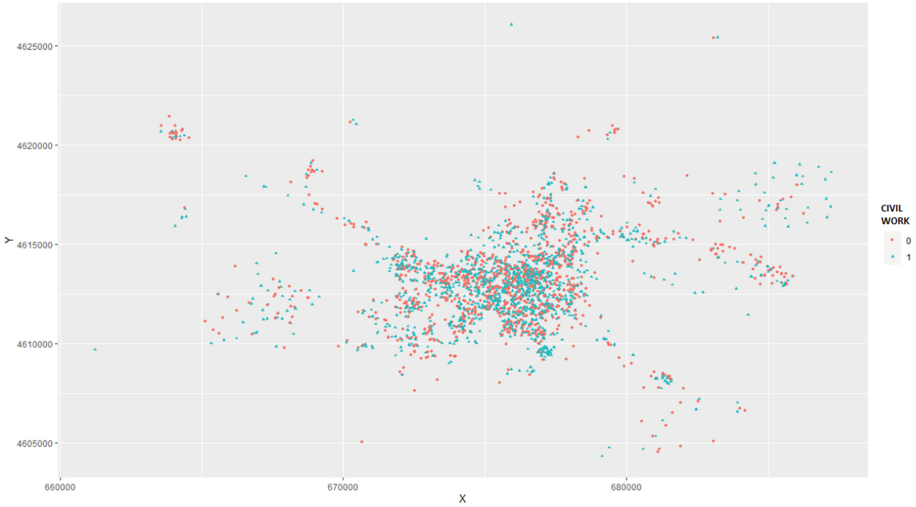


Fig. 2. Data map after the ‘WORK’ variable generation. 0 are the unloads that did not lead to construction work and 1 are the unloads that led to construction work.

2.4 Supervised Classification

In this phase we will fed the unload dataset to classification models using generated variable as class variable. We used 100 different seeds for every model, which was trained using 10 fold cross-validation [15]. This method partition the training dataset depending on a variable ‘k’ which in this case is 10. It partition the training dataset into 10 subset, then uses each subset as test data and the rest as training data. Consequently, avoids overfitting [16], overfitting occurs when a statistical model fits exactly against its training data. So the model cannot perform accurately against unseen data. Selected classifier for this experiments are: Support Vector Machine [17], Decision Tree [19], Random Forest [20], Gaussian Naive Bayes [22] and K-Nearest Neighbor [23].

2.5 Evaluation

Once the models are trained and tested it has to be evaluated to see the performance. For this task we used different evaluation metrics. This evaluation metrics are calculated using a confusion matrix. Confusion matrix is an $N \times N$ matrix, where N is the number of target classes. This matrix compares the actual target values with those predicted by the machine learning model. In our case, the matrix is 2×2 and it has 4 different values: TP (True Positive), TN (True Negative), FP (False Positive) and FN (False Negative). We select as evaluation measures these well known measures: accuracy, precision, recall, f-score, Matthews Correlation Coefficient and Modified Confusion Entropy (MCEN) [25].

After testing the models and measuring the performance practically, we have compared the models statistically. First we applied Kolmogorov Smirnov [26] to see if there is normality in our evaluation metrics. In our case, the p-value is less than .05, thus we reject the null hypothesis. We have sufficient evidence to say that the data does not follow normal distribution.

As our data does not come from a normal distribution we applied Kruskal Wallis [27]. After applying Kruskal Wallis we got the p-value higher than .05, in consequence we can conclude that there are not significant differences between machine learning models.

2.6 Results

In this section we will show the results achieved by the machine learning algorithms. The first table shows the results achieved of mean value and standard deviation for different evaluation metrics in each classification model. We achieved the best results for our dataset using random forest as we can see the numbers in boldface. On the other hand, the second table shows the confidence interval for previous evaluation metrics in each classification (Table 4 and 5).

Table 4. Mean and standard deviation of evaluation metrics for different machine learning algorithms.

Mean/ Standard deviation	SVM	Random Forest	Decision Tree	Gaussian Naive Bayes	k Nearest neighbor
Accuracy	0.68/ ± 0.02	**0.79**/ ± 0.02	0.75/ ± 0.02	0.65/ ± 0.03	0.53/ ± 0.02
Precision	0.71/ ± 0.02	**0.78**/ ± 0.02	0.71/ ± 0.07	0.58/ ± 0.03	0.57/ ± 0.02
Recall	0.74/ ± 0.03	**0.84**/ ± 0.03	0.82/ ± 0.10	0.77/ ± 0.03	0.62/ ± 0.04
f-Score	0.72/ ± 0.02	**0.81**/ ± 0.02	0.75/ ± 0.04	0.66/ ± 0.02	0.59/ ± 0.02
MCC	0.36/ ± 0.05	**0.55**/ ± 0.04	0.37/ ± 0.17	0.33/ ± 0.06	0.03/ ± 0.05
MCEN	0.77/ ± 0.02	0.65/ ± 0.03	0.64/ ± 0.16	0.74/ ± 0.03	**0.87**/ ± 0.01

Table 5. Confidence interval of evaluation metrics for different machine learning algorithms.

Confidence Interval	SVM	Random Forest	Decision Tree	Gaussian Naive Bayes	k Nearest Neighbor
Accuracy	0.685 – 0.688	0.787 – 0.790	0.749 – 0.752	0.650 – 0.654	0.528 – 0.532
Precision	0.710 – 0.713	0.781 – 0.784	0.705 – 0.713	0.578 – 0.582	0.568 – 0.571
Recall	0.736 – 0.741	0.844 – 0.847	0.817 – 0.830	0.775 – 0.779	0.623 – 0.628
F-score	0.723 – 0.726	0.811 – 0.813	0.751 – 0.756	0.662 – 0.666	0.594 – 0.597
MCC	0.359 – 0.366	0.554 – 0.560	0.362 – 0.384	0.331 – 0.338	0.026 – 0.034
MCEN	0.772 – 0.775	0.652 – 0.657	0.635 – 0.655	0.746 – 0.750	0.877 – 0.879

3 Conclusion

We have introduced a classification problem based on machine learning algorithms. The different algorithms has been demonstrated over this concrete geospatial dataset, which also can be used in similar datasets of different areas. Therefore, we can apply the previous steps to take advantage of the prediction, such as making most of the construction work area, etc. In future works, first, we will carry out the previous steps using the real dataset. Then, crossing our data with the data of the use of credit cards, we will analyze the economic impact of the construction work on local commerce. Thus, it can help on planning financial aid to the commerces of that area.

Acknowledgements. This work has been partially funded by Ministerio de Ciencia e Innovacion from Spain under the project PID2020-116346GB-I00

References

1. Lee, J.-G., Kang, M.: Geospatial big data: challenges and opportunities. Big Data Res. (2015). **2**(2), 74–81. Visions on Big Data (2015)
2. Breunig, M., et al.: Geospatial data management research: Progress and future directions. ISPRS Int. J. Geo-Inf. **9**(2) (2020)
3. Dembski, F., Wössner, U., Letzgus, M., Ruddat, M., Yamu, C.: UUrban digital twins for smart cities and citizens: the case study of Herrenberg, Germany. Sustainability **12**(6), 2307 (2020)
4. Effati, M., Thill, J.-C., Shabani, S.: Geospatial and machine learning techniques for wicked social science problems: analysis of crash severity on a regional highway corridor. J. Geogr. Syst. **17**(2), 107–135 (2015). https://doi.org/10.1007/s10109-015-0210-x
5. Mojaddadi, H., Pradhan, B., Nampak, H., Ahmad, N., bin Ghazali, A. H.: Ensemble machine-learning-based geospatial approach for flood risk assessment using multi-sensor remote-sensing data and GIS. Geom. Nat. Hazards Risk **8**(2), 1080–1102 (2017)
6. Jiang, Y., et al.: Towards intelligent geospatial data discovery: a machine learning framework for search ranking. Int. J. Digit. Earth **11**(9), 956–971 (2018)

7. Tehrany, M.S., Jones, S., Shabani, F., Martínez-Álvarez, F., Tien Bui, D.: A novel ensemble modeling approach for the spatial prediction of tropical forest fire susceptibility using LogitBoost machine learning classifier and multi-source geospatial data. Theoret. Appl. Climatol. **137**(1), 637–653 (2019)
8. Kovacs-Györi, A., et al.: Opportunities and challenges of geospatial analysis for promoting urban livability in the era of big data and machine learning. ISPRS Int. J. Geo-Inf. **9**(12) (2020)
9. Podgorski, J., Wu, R., Chakravorty, B., Polya, D.A.: Groundwater arsenic distribution in India by machine learning geospatial modeling. Int. J. Environ. Res. Public Health **17**(19) (2020)
10. Dollner, J.: Geospatial artificial intelligence: Potentials of machine learning for 3D point clouds and geospatial digital twins. PFG. Photogram. Remote Sens. Geoinf. Sci. **88**(1), 15–24 (2020)
11. Schubert, E., Sander, J., Ester, M., Kriegel, H.P., Xu, X.: DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. ACM Trans. Database Syst. **42**(3) (2017)
12. Fraley, C., Raftery, A.E.: Model-based clustering, discriminant analysis, and density estimation. J. Am. Stat. Assoc. **97**(458), 611–631 (2002)
13. Murdoch, J., Barnes, J.A.: Normal distribution. In: Statistics: Problems and Solutions, pp. 80–108. Palgrave Macmillan UK, London (1973). https://doi.org/10.1007/978-1-349-01063-9_4
14. Murphy, E.A.: One cause? many causes?: the argument from the bimodal distribution. J. Chronic Dis. **17**(4), 301–324 (1964)
15. Refaeilzadeh, P., Tang, L., Liu, H.: Cross-validation. Encyclop. Database Syst. **5**, 532–538 (2009)
16. Dietterich, T.: Overfitting and undercomputing in machine learning. ACM Comput. Surv. (CSUR) **27**(3), 326–327 (1995)
17. Noble, W.S.: What is a support vector machine? Nat. Biotechnol. **24**(12), 1565–1567 (2006)
18. Hofmann, M.: Support vector machines-kernels and the kernel trick. Notes **26**(3), 1–16 (2006)
19. Quinlan, J.R.: Learning decision tree classifiers. ACM Comput. Surv. (CSUR) **28**(1), 71–72 (1996)
20. Cutler, A., Cutler, D.R., Stevens, J.R.: Random forests. In Ensemble Machine Learning, pp. 157–175. Springer, New York (2012). https://doi.org/10.1007/978-1-4419-9326-7
21. Hip, T.K.: The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, Issue 8, 832–844 (1988)
22. Jahromi, A.H., Taheri, M.: A non-parametric mixture of Gaussian Naive Bayes classifiers based on local independent features. In: 2017 Artificial Intelligence and Signal Processing Conference (AISP), pp. 209-212. IEEE (2017)
23. Guo, G., Wang, H., Bell, D., Bi, Y., Greer, K.: KNN model-based approach in classification. In: Meersman, R., Tari, Z., Schmidt, D.C. (eds.) OTM 2003. LNCS, vol. 2888, pp. 986–996. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39964-3_62
24. Goldberg, Y.: A primer on neural network models for natural language processing. J. Artif. Intell. Res. **57**, 345–420 (2016)
25. Delgado, R., Núñez-González, J.D.: Enhancing confusion entropy (CEN) for binary and multiclass classification. PLoS ONE **14**(1), 1–30 (2019)

26. Massey, F.J., Jr.: The kolmogorov-smirnov test for goodness of fit. J. Am. Stat. Assoc. **46**(253), 68–78 (1951)
27. McKight, P.E., Najab, J.: Kruskal-Wallis Test. In: The Corsini Encyclopedia of Psychology, pp. 1–1. Wiley, New York (2020)

Virtual Sensor to Estimate Air Pollution Heavy Metals Using Bioindicators

María Inmaculada Rodríguez-García[1](✉), Nawel Kouadria[2],
Arantxa M. Ortega León[1], Javier González-Enrique[1], and Ignacio J. Turias[1]

[1] Higher Technical Engineering School, University of Cádiz, Avd. Ramón Puyol s/n,
11202 Algeciras (Cádiz), Spain
{inma.rodriguezgarcia,arantxa.ortega,javier.gonzalezenrique,
ignacio.turias}@uca.es

[2] Département le Vivant et L'Environnement, Université des Sciences et de la Technologie
Mohamed Boudiaf, El M'Naouar, Algérie
nawel.kouadria@univ-usto.dz

Abstract. The main objective of this work is to demonstrate that a set of bioindicators linked to the lichen *Lobaria Pulmonaria* and the bryophyte called *Leucodon Sciuroides* are adequate predictors of air pollution heavy metals (HM). A study case was performed in Oran, a port and coastal city in northwestern Algeria, located on the coast of the Mediterranean Sea. Each of the HM has been modelled using a machine learning procedure and in the experiments, the artificial neural networks (ANN) produces always better and more accurate results than multiple linear regression (MLR). Furthermore, good obtained results (R correlation coefficient greater than 0.9) demonstrate the main hypotheses and could be used as a virtual sensor.

Keywords: Virtual sensor · Air pollution estimation · Bioindicators · Machine learning

1 Introduction

Air pollution produces a reduction in the air quality in cities due to anthropogenic activities, this is why a wide variety of pollution sources coexist in densely populated cities. This is the case in the coastal area of Oran in northwestern Algeria where this study is developed. Oran has a huge commercial centre and an important port reminding us of other coastal areas where pollution studies have been recently developed [1, 2].

Amongst air pollutants released into the urban environment are traced metals elements (TMEs) due to the urban dust of rolling traffic. Heavy metals in the air affect the health of both human beings and ecosystems with dose-dependent toxicity levels [3]. This has prompted authorities worldwide to establish means of monitoring air quality, including air pollutant monitoring systems, however, air quality monitoring stations provide information about regulatory air pollutants such as gaseous pollutants or particle matter (PM), but rarely about heavy metals and organic pollutants [4, 5].

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 208–216, 2023.
https://doi.org/10.1007/978-3-031-18050-7_20

In this study, the use of bioindicators has been tested to estimate heavy metals (HM) air pollutants. Bioindicators are organisms that have the ability to bioaccumulate pollutants in their tissues or on their surfaces [6] and can detect the degradation of air quality before it severely affects the biotope or human beings.

Plant biomonitoring of air quality using lichens has acquired a considerable scientific maturity over the last years and has become a valuable complement to instrumental measurements. The use of lichens and bryophytes is based on the fact that they have no roots, making them in a direct and unique contact with the air in their environment [7]. Continuing studies in the field of lichen (*Lobaria Pulmonaria)* and bryophyte (*Leucodon Sciuroides*) monitoring around air pollution [8, 9], the data of biomonitoring air quality in the City of Oran (Algeria) have been used in this study. This study is a continuation research in the same field of biological pollution bioindicators as shown by studies such as [3–9].

In this work, the aim was to test bioindicators to design a virtual sensor, continuing with studies in port areas such as [10–12], in order to avoid carrying out more certain costly laboratory tests. For this purpose, multiple linear regression (MLR) was tested as a benchmark method and, on the other hand, the efficiency of models based on artificial neural networks (ANNs) was tested, comparing both results. The ANNs have non-linear modelling capabilities that usually allow better behaviour to be obtained, and also usually superior generalization results are provided for unseen data. A pioneer study in this field is [7] where they realised that pure instrumental pollution monitoring presented weak points due to calibration or inaccuracy. The use of bioindicators such as lichens presented more sensitivity and they were more affordable. Machine Learning approach has been used in actual studies related to lichens facing air pollution forecasting. In the study [13] lichens were used in building surfaces in smart cities as an indicator of air pollution. Advancements in technology permit to apply machine learning to bioindicators proving that this technique is effective in monitoring air quality. Additionally, the main objective is to design a virtual sensor to emulate the performance of an air pollution bioindicator and therefore for saving costs of laboratory and/or human resources.

The advantages of the approach presented in this work is to develop a simple and cost-effective alternative for instrumental measures enabling a much higher spatial resolution of measurements. In fact, studies obtained air pollution synthetic data from virtual sensors providing complementary information [14].

This paper is organized as follows. The database is given in Sect. 2. The methodology is presented in Sect. 3. The results are conducted in Sect. 4. Finally, some conclusions are shown in Sect. 5.

2 Database

This study is located in the city of Oran, Algeria. Oran is a city in northwestern Algeria, located on the coast of the Mediterranean Sea.

The database was collected from the period of December 2018 to June 2019 in one control site in the Mountains of Annaba and twelve sample points spread by the city of Oran (see Fig. 1) following a biomonitoring technique that consists of the transplantation of bioindicators samples using the bryophyte called *Leucodon Sciuroides* and the lichen *Lobaria Pulmonaria* resulting from the symbiosis between a fungus (*Ascomycetes*) which ensures protection against dehydration, and a green alga (*Dictyochloropsis Reticulata*) which produces the necessary carbohydrates for the organism [15]. Bryophytes (B) are the small green plants commonly known as mosses, liverworts and, hornworts. Lichens (L) are dual organisms consisting of a fungus and an alga or a cyanobacterium. Both are cryptogam, but the bryophyte (B) acts as like bio accumulators (because bryophyte can accumulate very important amounts of heavy metals) and the lichen (L) acts as much more as a biomonitor since its more sensitive to low concentrations, so it reacts to very low concentrations of xenobiotics [9].

The first twelve sample points were set in urban areas where pollution is supposed to be higher, and the last sample point, the control site, was set in an unspoilt mountain zone where there is no existence of human activity.

Monthly, samples were carried out in order to conduct the analyses and the dosages (ascorbate peroxidase activity (APX), guaiacol peroxidase activity (GPX), catalase activity (CAT), chlorophyll *ab* content, chlorophyll *a* content, chlorophyll *b* content, as well as the fresh weight/dry weight ratio (FW/DW)) to quantifying the content of biological substance. All these dosages data form our database. The complex and expensive laboratory techniques for measuring HM could be improved by applying the virtual sensor.

Table 1 shows the names of heavy metals, bioindicators, and their units.

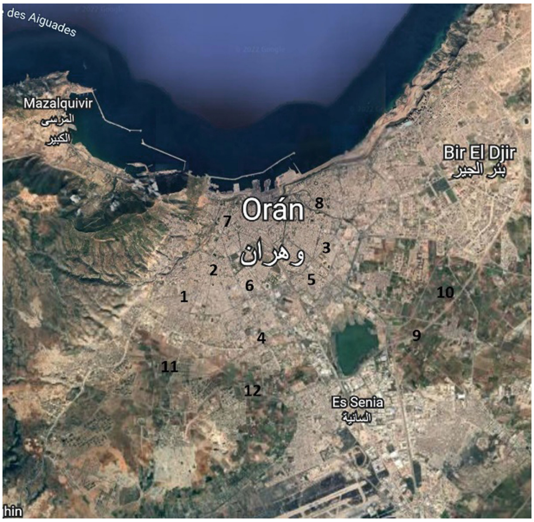


Fig. 1. Location of the sample points in the city of Oran.

Table 1. Heavy metals and bioindicators used as variables of the study.

Name initials	Description		Units
B	The bryophyte called *Leucodon Sciuroides*	Cryptogam (inferior plant)	—
L	The lichen is a Foliaceous epiphyte called *Lobaria Pulmonaria.* A symbiotic association between algae *Dictyochloropsis Reticulata* and fungus *Ascomycetes*		
Pb	Heavy metal Lead	Metallic trace elements	ppm
Cr	Heavy metal Chrome		
Cd	Heavy metal Cadmium		
APX	Ascorbate peroxidase	Detoxification enzyme	µmol/min/mg of proteins (nano mole/minutes/ milligrammes of proteins)
GPX	Guaiacol peroxidase		
CAT	Catalase		
Chl ab	Chlorophyll pigments differing in their concentrations in plants and their role		mg/g
Chl a			
Chl b			
FW/DW	Fresh weight/dry weight ratio		Adimesional (mg/mg)

3 Methodology

In this approach, a soft sensor for HM concentrations estimation has been developed. HM air pollutants are a function of the bioindicators (HM = f (bioindicators)), i.e., a soft sensor of them, therefore, we can use each model (the model which we will select for each HM output) as a digital twin of each bioindicator.

Two different methods were compared in order to get knowledge about their behaviour in this problem: Multiple Linear Regression (MLR) and Artificial Neural Networks (ANNs). Heavy metals such as lead (Pb), cadmium (Cd), and chromium (Cr) are the desired outputs to be estimated as a function of the measured bioindicators (see Table 2). Two different experimental tests have been developed: test-1 using the bryophyte together with the rest of heavy metals and bioindicators, and test-2 using the lichen instead of bryophyte. Our approach used the heavy metals as outputs and bioindicators as inputs.

In this study, we have used multilayer perceptrons (MLP) with feedforward connections, fully connected by layers, and with backpropagation training algorithm using the Levenberg Marquardt optimisation algorithm (which is a quasi-Newton algorithm). Besides, algorithm uses early-stopping, when it detects that it does not improve with the internal validation data part [16, 17]. An MLP architecture based on three layers has

Table 2. The two tests and their variables of study: Outputs and Inputs. B (Bryophyte) and L (Lichen).

Test-1		Test-2	
Name initials	Variable	Name initials	Variable
B-Pb	Output	L-Pb	Output
B-Cr		L-Cr	
B-Cd		L-Cd	
B-APX	Inputs	L-APX	Inputs
B-GPX		L-GPX	
B-CAT		L-CAT	
B-Chl ab		L-Chl ab	
B-Chl a		L-Chl a	
B-Chl b		L-Chl b	
B-FW/DW		L-FW/DW	

been applied in this study. Sigmoidal functions have been used for neurons in the hidden layer and linear function for the elements in the output layer.

A randomised procedure has been designed in order to compare the different models. Statistical methods analyse the behaviour on average. When this resampling procedure strategy is adequately applied to the obtained error rates (or correlation coefficient) in a well-designed experiment, it is possible to determine the optimal complexity of a neural network model. Several neural networks topologies have been tested using different hidden units (in a three-layer MLP feedforward network).

Once the best model has been selected, we propose to use it to determine the relative importance of the bioindicator's input variables for each experimental test (see Table 2). The connection weights of each selected neural network model are used according to the procedure developed by [18]. This procedure has been widely successfully applied in environmental sciences [19, 20].

$$IM\left(X_b\right) = \frac{\sum_{j=1}^{N}\left[\left(|W|b_j \sum_{k=1}^{N_b}|W|b_{j,k}|O|_j\right]}{\sum_{i=1}^{N_b}\left\{\sum_{j=1}^{N}[(|W|b_{i,j}\sum_{k=1}^{N_b}|W|b_{i,j,k})|O|_j]\right\}} \tag{1}$$

Equation 1 was proposed by [17]. *IM(X_b)* is the importance for the b^{th} input variable X_b (here, bioindicators). N_b is the number of inputs. $|W|b_j$ term is the absolute value corresponding to the b^{th} input variable and the j^{th} hidden layer. $|O|_j$ is the absolute value of the output layer weight corresponding to the j^{th} hidden layer.

4 Results

An experimental procedure for the computation of MSE and R quality indexes has been used in this work in order to compare the quality of the tested models in the proposed estimation approach. Table 3 shows the obtained results in a 20-times resampling procedure for each HM bioindicator´s soft sensor.

Table 3. Comparison between methods: MLR and ANNs for 20 repetitions in the two tests. B (Bryophyte) and L (Lichen).

Test	B/L-Heavy metals	Quality index	MLR	ANNs					
				Number of hidden units (input-hidden-output)					
				1	2	5	10	20	50
Test 1	B-Pb	**MSE**	1,673	**1,537**	1,579	5,721	3,959	2,965	3,088
		R	0.911	**0.935**	0.922	0.912	0.920	0.899	0.851
	B-Cr	**MSE**	20.481	24.985	18.556	19.657	**16.797**	26.257	70.562
		R	0.936	0.926	0.9423	0.943	**0.944**	0.914	0.793
	B-Cd	**MSE**	55.775	**57.604**	73.711	88.402	85.941	162.038	257.701
		R	0.8932	**0.8979**	0.8760	0.8438	0.8521	0.8012	0.684
Test 2	L-Pb	**MSE**	1,128	1,617	1,191	**968.1**	1,949	2,474	4,664
		R	0.853	0.802	0.881	**0.877**	0.775	0.753	0.634
	L-Cr	**MSE**	53.878	56.418	122.062	47.291	**46.816**	122.188	137.874
		R	0.846	0.820	0.851	**0.878**	0.863	0.753	0.796
	L-Cd	**MSE**	1.717	1.961	**1.7166**	1.887	2.776	3.383	6.057
		R	0.883	0.854	0.885	0.872	**0.893**	0.816	0.613

Table 3 shows the best configurations (best models) for each of the HM analysed bioindicators. The best configurations are marked in bold and they are results from a *post-hoc* Bonferroni-test [21] applied in a randomised experimental procedure. The results are collected for test sets and they are the mean of the 20-times repetition procedure. In this work, in each repetition, an internal validation has been applied using 15% of the data, while to test and calculate the results another 15% of the data has been used. In the case of B-HM estimators, the results reached more than 0.94 in the correlation coefficient, and around 0.89 in the case of L-HM estimators. Both generalization results (for test sets) were obtained for a ANN using 10 hidden neurons. It is not necessary to use a more complex model (using more neurons). These results show the suitability of the proposed approach. Therefore, a virtual sensor could be used as an alternative of bioindicators of air pollution heavy metals.

Once the best neural model is selected, which means, the one that obtains the best generalisation for test patterns, an analysis of the relevance of the input variables in terms of the neural weights has been performed. The results are shown in Table 4.

As can be seen in bold, the variables with a higher relative contribution are APX for B-Pb (0.1942/0.0564) or Chl ab for L-Pb (0.1758/0.0760) for the importance of the bioindicators.

The computed importance of the bioindicators could help to ecotoxicologists or experts to use the most relevant ones.

Table 4. Relative importance of the input variables to their best ANN model. Mean and standard deviation (SD). B (Bryophyte) and L (Lichen).

Test	Heavy Metals		HM-APX	HM-GPX	HM-CAT	HM-Chl ab	HM-Chl a	HM-Chl b	HM-FW/DW
Test 1	B-Pb	**Mean**	**0.1942**	**0.1630**	0.1447	0.0899	0.0826	**0.1864**	0.1392
		SD	**0.0564**	**0.0864**	0.0839	0.0563	0.0523	**0.0780**	0.0782
	B-Cr	**Mean**	**0.1641**	0.1323	**0.1517**	**0.1797**	0.1305	0.1233	0.1184
		SD	**0.0453**	0.0342	**0.0326**	**0.0616**	0.0445	0.0409	0.0443
	B-Cd	**Mean**	**0.4620**	0.1475	0.0759	0.0735	0.0362	0.1408	0.0642
		SD	**0.1160**	0.1023	0.0586	0.0426	0.0343	0.0658	0.0282
Test 2	L-Pb	**Mean**	0.1202	**0.1509**	0.1283	**0.1758**	0.1422	0.1358	0.1469
		SD	0.0491	**0.0492**	0.0599	**0.0760**	0.0562	0.0518	0.0408
	L-Cr	**Mean**	**0.2180**	0.1204	0.1104	0.1334	0.1214	0.1299	**0.1665**
		SD	**0.0847**	0.0536	0.0568	0.0396	0.0372	0.0445	**0.0553**
	L-Cd	**Mean**	0.1369	**0.1539**	**0.1722**	0.1391	0.1408	0.1306	0.1264
		SD	0.0313	**0.0360**	**0.0263**	0.0329	0.0298	0.0307	0.0286

5 Conclusions

This study describes a suitable methodology for the estimation of air pollution heavy metals levels. It is shown how virtual sensor models can be developed including the exogenous information of other different bioindicators.

The random procedure applied in the study presented here leads to robust results, minimising the uncertainty associated with the natural data randomness. This procedure allows us to determine the optimal complexity of the models. Besides, we analysed the relative importance of the different bioindicators in order to get more physical information about the process. Errors are less than 5% for the best models based on ANNs. Therefore, Pb, Cr, and Cd values can be successfully inferred as a function of more easily measured and cheaper biomarkers.

Acknowledgements. This work is part of the research project RTI2018-098160-B-I00 supported by ‘MICINN’ Programa Estatal de I+D+i Orientada a ‘Los Retos de la Sociedad’.

References

1. González-Enrique, J., Turias, I.J., Ruiz-Aguilar, J.J., Moscoso-López, J.A., Franco, L.: Spatial and meteorological relevance in NO_2 estimations: a case study in the Bay of Algeciras (Spain). Stochast. Environ. Res. Risk Assess. **33**(3), 801–815 (2019)
2. Rodríguez-García, M.I., González-Enrique, J., Moscoso-López, J.A., Ruiz-Aguilar, J.J., Rodríguez-López, J., Turias, I.J.: Comparison of maritime transport influence of SO_2 levels in Algeciras and Alcornocales Park (Spain). XIV Conf. Transp. Eng. CIT2021, **58**, 2352–1465 (2021)
3. Lanier, C., et al.: Combined toxic effects and DNA damage to two plant species exposed to binary metal mixtures (Cd/Pb). Ecotoxicol. Environ. Saf. **167**, 278–287 (2019)
4. Uzhinskiy, A., Aničić Urošević, M., Frontasyeva, M-V.: Prediction of air pollution by potentially toxic elements over urban area by combining satellite imagery, moss biomonitoring data and machine learning. Ciência e Técnica Vitivinícola J. **35**(12), 34–46 (2020)
5. Di Fiore, C., et al.: Honeybees as bioindicators of heavy metal pollution in urban and rural areas in the South of Italy. Atmosphere **13**, 4 (2022)
6. Asta, J., Garrec, J.P.: Etude de l'accumulation du fluor dans les lichens d'une vallee alpine polluee. Environ. Poll. Ser. Ecolog. Biol. **4**, 267–286 (1980)
7. Bargagli, R., Nimis, P.L., Monaci, F.: Lichen biomonitoring of trace element deposition in urban, industrial and reference areas of Italy. J. Trace Elem. Med Biol. **11**, 173–175 (1997)
8. Van Haluwyn, C., Semadi, A., Deruelle, S., Letrouit, M.A.: The corticolous lichen vegetation of the Annaba, eastern Algeria; [La vegetation lichenique corticole de la region d'Annaba (Algerie orientale)]. Cryptogamie: Bryologie et Lichenologie, **15**, 1–21 (1994)
9. Kouadria, N., Belhoucine, F., Bouredja, N., Ait Kaci, M., Abismail, Y., Alioua Berrebba, A.: Bioaccumulation of lead by Xanthoria Parietina and Hylocomium Splendens, and its effect on some physiological parameters. J. Mater. Environ. Sci. **11**, 247–254 (2020)
10. Turias, I.J., González, F.J., Martín, M.L., Galindo, P.L.: Prediction models of CO, SPM and SO_2 concentrations in the Campo de Gibraltar region, Spain: a multiple comparison strategy. Environ. Monit. Assess. **143**(1–3), 131–146 (2008)
11. González-Enrique, J., Ruiz-Aguilar, J.J., Moscoso-López, J.A., Urda, D., Turias, I.J.: A comparison of ranking filter methods applied to the estimation of NO_2 concentrations in the Bay of Algeciras (Spain). Stoch. Env. Res. Risk Assess. **35**, 1999–2019 (2021)
12. Moscoso-López, J.A., Urda, D., González-Enrique, J., Ruiz-Aguilar, J.J., Turias, I.J.: Hourly air quality index (AQI) forecasting using machine learning methods. In: Herrero, Á., Cambra, C., Urda, D., Sedano, J., Quintián, H., Corchado, E. (eds.) SOCO 2020. AISC, vol. 1268, pp. 123–132. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-57802-2_12
13. Sri Preethaa, K.R., Yuvaraj, N., Jenifa, G., Indhu, R., Kanmani, P.: Lichen element based autonomous air pollution monitoring around smart cities – a deep learning approach. Turkish J. Comput. Math. Educ. **12**(10), 151–161 (2021)
14. Campos, G.O., Aparecido Villas, L., Da Cunha, F.D.: Analysis of air pollution utilizing virtual sensor models. In: Proceedings - 2021 IEEE Latin-American Conference on Communications, LATINCOM, pp. 1–6 (2021)
15. Carlsson, R., Nilsson, K.: Status of the red-listed lichen Lobaria pulmonaria on the Åland Islands. SW Finland. Ann. Botanici Fennici. **46**(6), 549–554 (2009)
16. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. In: Parallel Distributed Processing: Explorations in the Microstructure of Cognition. MIT Press, Cambridge (1986)
17. Hornik, K., Stinchcombe, M., Halbert, W.: Multilayer feedforward networks are universal approximators. Neural Netw. **2**, 359–366 (1989)
18. Garson, G.D.: Interpreting neural connection weights. Artif. Intell. Expert **6**, 47–51 (1991)

19. Elkamel, A., Abdul-Wahab, S., Bouhamra, W., Alper, E.: Measurement and prediction of ozone levels around a heavily industrialized area: a neural network approach. Adv. Environ. Res. **5**, 47–59 (2001)
20. Martín, M.L., et al.: Prediction of CO maximum ground level concentrations in the Bay of Algeciras. Spain Artif. Neural Netw. Chemosphere **70**(7), 1190–1195 (2008)
21. Jobson, J.D.: Applied Multivariate Data Analysis. Springer Texts in Statistics, vol. 1, Springer-Verlag, New York (1991). https://doi.org/10.1007/978-1-4612-0955-3_3

Regression Techniques to Predict the Growth of Potato Tubers

Ángel Arroyo[1](✉), Carlos Cambra[1](✉), Nuño Basurto[1], Carlos Rad[2], Milagros Navarro[2], and Álvaro Herrero[1]

[1] Grupo de Inteligencia Computacional Aplicada (GICAP), Escuela Politécnica Superior, Universidad de Burgos, Av. Cantabria s/n, 09006 Burgos, Spain
{aarroyop,ccbaseca,nbasurto,ahcosio}@ubu.es

[2] Grupo de Investigación en Compostaje (UBUCOMP), Escuela Politécnica Superior La Milanera, Universidad de Burgos, c/ Villadiego s/n, 09001 Burgos, Spain
{crad,minago}@ubu.es

Abstract. Potatoes (*Solanum tuberosum*) is one of the highest produced commodities for human consumption, with a global production of 359.07 mega metric ton per year and covering a huge cultivation area in the world. As a result, being able to predict the yield of this crop is an interesting topic, with a high impact on agriculture production. Accordingly, to predict some potato-tuber production and quality indicators, this work innovates by using phenotypical plant characteristics to which soft-computing techniques are applied. More precisely, the Linear Multiple-Regression, the Radial Basis Function Network, the Multilayer Perceptron, and the Support Vector Machine are benchmarked in the present work. Promising results have been obtained, validating the application of soft-computing techniques to predict the growth of potato tubers.

Keywords: Precision agriculture · Potato crop · Yield prediction · Artificial neural networks · Regression

1 Introduction

Smart or Precision Agriculture is a challenging field that is attracting increasing efforts from researchers in the Soft Computing community. Among this field, it is particularly interesting the topic of yield prediction. Predicting crop yield and product quality is a real challenge due to the important variations induced by climate, soil properties, fertilizers, crop management, and plant species and varieties [1].

Plant phenotyping usually implies labour intensive activities such as physical sampling of plant tissues, counting flowers or fruits at various phases of growth, and lab determination of fruit quality based on physicochemical analyses. In this way, great efforts are being devoted to it by using image analysis in general and deep learning models in particular, trying to get precise field and yield maps currently [2].

With a global production of 359.07 mega metric ton in 2020, and covering an area higher than 17 million ha in the world [3], potatoes (*Solanum tuberosum*) rank as the fifth

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 217–225, 2023.
https://doi.org/10.1007/978-3-031-18050-7_21

highest produced commodity used for human consumption, with a constantly growing market. International potato trade has doubled from 1986 (<10 mega metric ton) to 2006 with a quadrupled value in the same time range reaching $9,600 million in 2007 [4]. These statistics and trends support the need for efficient, accurate, rapid, reproducible, cost effective and easy-to-use systems to assure that final production meets the required quantity and quality [5]. Specially, an accurate estimation of optimal harvest time is critical for potato tubers as it strongly affects quality. In the case of potato, image processing is particularly challenging as tuber is formed into the soil, its shape is irregular, its colour is like the colour of the soil, and this colour varies according to the change in water content [6].

Hence, some other alternatives must be explored for yield prediction regarding potato crops, other than image processing. In keeping with this idea, to predict some of crop production and quality indicators using alternative methods, this work aims to use phenotypical plant characteristics to predict tuber quality properties by means of supervised soft-computing techniques. More precisely, the following regression techniques are applied in the present paper: linear multiple-regression [7] and soft-computing models, namely the Radial Basis Function Network (RBFN) [8], the Multilayer Perceptron (MLP) [9], and the Support Vector Machine (SVM) [10]. These techniques are applied in a novel way to predict crop growth.

The remaining sections of this study are structured as follows: previous work is discussed in Sect. 2 while the applied techniques are described in Sect. 3. Section 4 introduces the real-life problem that is addressed as well as the associated dataset, while the obtained results are presented in Sect. 5. Finally, both conclusions and future work proposals are discussed in Sect. 6.

2 Previous Work

Smart agriculture and the yield prediction subfield require a set of technologies that usually combines sensors, information systems, machinery with automation skills, and data processing capabilities. Among all of them, satellite imagery has been widely used to predict yield indicators. Many studies have demonstrated the validity of using soft-computing techniques such as regression trees, random forest, multivariate regression, association rule mining, and ANN to predict yield indicators, enabling farmers to take appropriate measures for fertilization variability and plan best harvest period [11].

This kind of techniques have been previously applied to a wide variety of crops, as yield prediction significantly varies from one to another. Some researchers [12] have evaluated different strategies for monitoring within-field soybean yield using Sentinel-2 visible, near-infrared and shortwave infrared (Vis/NIR/SWIR) spectral bands. In order to process the images, partial least squares regression (PLSR) and Support Vector Regression (SVR) methods have been applied. Normalized difference vegetation index (NDVI) is a useful vegetation index, that has been used in to detect seasonal and inter-annual changes in vegetation growth and activity and it is sufficiently stable to permit meaningful comparisons [13].

The growing demand for potato, coupled with climate change and hottest and driest summer periods in the Mediterranean area, implies the need for better crop protection and management practices in order to improve crop yields and reduce the waste of water resources [14]. Baker and Kirk [15] compare four types of models for incorporating extended range forecast data into a potato late blight risk system A method was developed to derive hourly microclimate variables associated with potato late blight risk. Data was analyzed by determinacy analysis, logistic regression, discriminant analysis and ANN models. Some authors developed a tailored algorithm for harvest prediction [16], that was designed for early prediction of potato yield (i.e. prior to the harvest period) using multispectral satellite remote sensing on a field scale. More recently, ANNs have also been applied to potato crop prediction, obtaining successful results as forecasting models of tuber yield [17]. Authors compared the prediction accuracy of a linear regression model and an ANN for three very early potato cultivars. Some positive results have been reported using multispectral images obtained from Unmanned Aerial Vehicles (UAVs). That is the case of Li *et al.* [18], which proposed Random Forest Regression (RFR) and Partial Least Square Regression (PLSR) to process such images.

Differentiating from this previous research, in the present paper the prediction accuracy focused on tuber size prediction, an important quality production factor, in contrast with previous works using ANNs to predict an estimation yield without size determination of tubers. Number and weight values from acquisition data of the tubers were determined according to the caliber of a random collection of several samples taken with a certain spatial periodicity. This type of variable is an input data for Radial Basis RBFN. Results can help in advance to the farmer to commercial market strategy to the harvests date.

3 Regression Techniques

In order to predict the growth indicators on the data described in Sect. 3, four regression techniques (both statistic and soft-computing techniques) are applied. They are described in the following subsections.

3.1 Multiple Linear Regression

The Multiple Linear Regression (MLR) models the relationship between a target variable and at less two variables by generating a linear equation to observed data [7]. A value of the target variable (y) is linked to a value of the independent variable (x). The regression line for p explanatory variables $(x_1, x_2,\ldots, x_p)$, is defined (1):

$$u_{\mathrm{y}} = \beta_0 + \beta_1 \mathrm{x}_1 + \beta_2 \mathrm{x}_2 + \ldots + \beta_{\mathrm{p}} \mathrm{x}_{\mathrm{p}} \tag{1}$$

The fitted values b_0, b_1,…,b_{p} estimate the parameters β_0, β_1,…,β_{p} of the regression line. Equation 1 describes how the mean response u_{y} varies according to the explanatory variables. The observed values for y change about their means u_{y} and computes the same standard deviation (σ).

The model can be expressed as DATA = FIT + RESIDUAL, where the “RESIDUAL” term refers to the deviations of the observed values y from their means u_{y} and “FIT” is represented by (1).

3.2 Multilayer Perceptron

The well-known Multilayer Perceptron (MLP) is an Artificial Neural Network (ANN) [19] consisting of several layers of nodes. The weights associated to the connected nodes and output signals are generated by calculating the activation to the sum of the inputs. Its architecture consists of an input layer that pass the input vector to the other layers of the network. The terms "input vectors" and "output vectors" refer to the inputs and outputs of the MLP and are represented as single vectors [20]. MLPs are fully connected: every node is connected to each one of the nodes in the previous and next layer.

During training, the update of weights is performed according to the backpropagation [21] algorithm. In the present paper, the Levenberg-Marquardt (LM) [22] training method has been applied.

3.3 Radial-Basis Function Network

The Radial-Basis Function Network (RBFN) [23] also is an ANN where a centroid is associated to each node in the hidden layer, and for each input vector $x = (x_l, x_2, \ldots, x_n)$, it computes the distance between the node centroid and x and its centroid. The output of the unit is then calculated as a non-linear function of this distance. Finally, the output of hidden nodes are weighted and combined in the nodes of the output layer.

In the case of r input nodes and m output nodes, the response function of each one of the output nodes can be calculated as:

$$\sum\nolimits_{i=1}^{M} W_i * K(\frac{x - z_i}{\delta_i}) = \sum\nolimits_{i=1}^{M} W_i * g\left(\frac{||x - z_i||}{\delta_i}\right) \tag{2}$$

where x is an input vector , M ЄN is the number of hidden units in the hidden layer, Wi Є $\mathbb{R}m$ are the weights linking the ith hidden-layer unit to the output nodes, K is a kernel function that is radially symmetric, σ_i is the smoothing factor of the ith kernel node, z_i is the centroid factor of the ith kernel node, and g: $[0, \infty)$ -> $\mathbb{R}$ is the activation function.

3.4 Support Vector Machine

A Support Vector Machine (SVM) [24] is a machine learning tool for classification and regression [25]. SVM regression, based on statistical learning, is considered a nonparametric technique because it relies on kernel functions. In SVM regression, the set of training data includes predictor variables and observed response values. The goal is to find a function $f(x)$ that deviates from y_n by a value no greater than ε for each training point x, and at the same time is as flat as possible.

4 Materials and Methods

Field experiments were conducted from April 16th to October 10th 2019, in a potato field crop of 5 ha, located in Cabia (Spain), 42°16′57" N and 3°51′25" W. Soil was classified as *Calcic Luvisol* (*LVk*) according to Food and Agriculture Organization FAO, with loam texture, bulk density 1.26 kg L^{-1}, field capacity 0.31 (w/w), pH (1:5 w/v) 7.6, Electrical

Conductivity (1:5 w/v, 25 °C) 0.65 dS m^{-1}, Organic Mater 3.33%, Total N 0.16% and lime 16.7%. Climate in this area is Attenuated Mesomediterranean, according to FAO.

Potatoes (*Solanum tuberosum* L. Var. Agria) were planted in April 16th and from mid-June, phenological development was assessed according to BBCH-scale and four plants from the centre of the plot (20 × 20 m) were removed for laboratory analysis every 15 days. As a result, the following morphological parameters related to the plant were collected regarding this potato plantation:

1. *Plant height1: a Carpenters meter (±1 mm) was used.*
2. *Plant weight1: a weight scale (±1 mg) was used.*
3. *% Humidity1: weight losses after 38 h at 70 ºC (±1 ºC).*
4. *Aerial part length1: a Ruler lab (±1 mm) was used.*
5. *Roots length1: a Ruler lab (±1 mm) was used.*
6. *Plant Nitrogen content1: aerial part of plants was dried at 70 ºC and thereafter, ground in a mill. Samples of 0.2 g were analysed by Dumas method in a TruSpec CN (LECO, USA) with IRD (Infra-Red Detector) and TCD (Thermal Conductivity Detector) for CO_2 and N_2, respectively.*

Additionally, before harvesting, four sampling locations of 3 m^2 were chosen at random for yield estimation; tubers were classified by considering their diameter in different commercial classes: >80 mm, between 40–80 mm and <40 mm (as shown in Fig. 1).

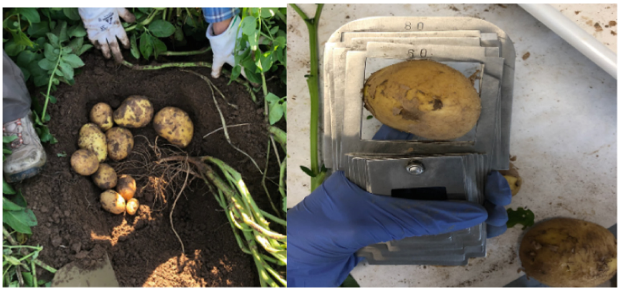


Fig. 1. Tuber samples measurement for data collection.

The following features about the tubers are available for a given time frame:

1. Tubers weight per plant1: a weight scale (±1 mg) was used.
2. Number of tubers per plant1: tubers were visually counted.
3. Tubers humidity1: weight losses after 38 h at 70 ºC (±1 ºC).
4. Percentage of tubers in the 0–40 cm diameter range1: a squared measurement frame of 40 cm was used.
5. Percentage of tubers in the 40–80 cm diameter range1: squared measurement frames of 40 and 80 cm were used.
6. Percentage of tubers in the > 80 cm diameter range1: a squared measurement frame of 80 cm was used.

7. Tubers Nitrogen content1: crushed fresh tubers were dried at 70 ºC and there-after, ground in a mill. Samples of 0.2 g were taken and analysed as previously described.

As only some measurements were taken, in order to get daily data, the widely-used cubic interpolation method has been applied. This is a shape-preserving method for interpolation that, based on the shape of the known data, generates new values taking into account the values at neighboring grid points. This method has been selected based on previous results on a potato plantation dataset [26].

5 Results and Discussion

The techniques described in Sect. 3 have been applied to the case study presented in Sect. 4 and the results are discussed in the present section. To obtain more accurate results, they are validated by the n-fold cross-validation [27] scheme (data partitions has been set to the value of 10 for all the experiments).

The regression has been carried out on the seven indicators related to the tuber, based on the six indicators related to the potato plant, all of which are described in Sect. 4. The indicator on which the regression is performed is set as the predictor attribute and the plant six features are the criterion ones.

Table 1 shows the Mean Squared Error (MSE) [28] value obtained after applying the MLR and SVM techniques. Both mean values and standard deviation for the cross-validated executions are shown. Lowest mean MSE values for each applied model are highlighted in bold.

Table 1. MLR and SVM prediction results (in terms of MSE) for the target indicators.

Indicator	MLR		SVM	
	Mean	Std Dev	Mean	Std Dev
Weight	3.76E−03	1.90E−04	4.30E−05	1.01E−05
Tubers	4.25E−08	3.88E−09	1.81E−08	1.55E−08
Humidity	1.58E−07	1.67E−08	1.70E−08	1.20E−08
Caliber 0–40	4.50E−07	2.90E−08	5.27E−08	3.36E−08
Caliber 40–80	5.09E−07	2.88E−08	2.17E−07	1.05E−07
Caliber 80	1.22E−07	7.33E−09	3.91E−08	2.58E−08
Nitrogen	4.40E−07	2.16E−08	3.69E−08	2.10E−08

From the MLR results, it can be seen that the best results in terms of error are obtained for the indicator “number of tubers”, while the highest error corresponds to the indicator “Weight”, which is associated to worst results (highest error) than the rest of the indicators analysed. In the case of SVM, in general terms better results (lower error) than in the case of MLR have been obtained. The indicator with the best SVM results is “Humidity”, although similar error rates have been obtained for the “Tubers” indicator.

Once again, the worst prediction (highest error) has been obtained for the "Weight" indicator. However, the error is significantly reduced by SVM when compared to that obtained by MLR.

Table 2 shows the MSE values obtained after applying the neural models (MLP and RBFN).

Table 2. MLP and RBFN prediction results (in terms of MSE) for the target indicators.

Table. Indicator	MLP		RBFN	
	Mean	Std Dev	Mean	Std Dev
Weight	8.88E−02	6.73E−02	7.07E−07	4.73E−08
Tubers	7.19E−02	8.08E−03	1.73E−10	1.25E−11
Humidity	7.41E−02	4.30E−03	2.27E−10	2.28E−11
Caliber 0–40	8.98E−02	1.16E−02	9.03E−10	6.28E−10
Caliber 40–80	7.90E−02	5.73E−03	4.71E−09	1.97E−10
Caliber 80	7.71E−02	4.94E−03	6.33E−10	2.89E−11
Nitrogen	6.45E−02	4.16E−03	3.99E−10	1.38E−10

In the case of MLP, the results are slightly different from those previously introduced (Table 1). Quite similar error rates have been obtained for all the indicators, being the worst ones (highest error) in all the experiments that have been carried out. Error rates for the "Weight" indicator are similar to those of the other ones. The indicator associated to the best MSE results is "Nitrogen" and the worst is "Caliber 40–80", although there are small differences among the indicators.

Finally, an analysis of the RBFN results reveals that this model attains the best error rates (both in terms of mean and standard deviation). More precisely, the lowest mean MSE value in all the performed experiments has been obtained when applying RBFN to predict the "Tubers" indicator. Furthermore, RBFN is the technique that obtains the best results when predicting the tuber indicators, in terms of the mean MSE, being SVM the second best.

6 Conclusions and Future Work

In this research work, four regression techniques have been applied to predict seven indicators of potato tubers based on another indicators obtained from the potato aerial part.

Tables 1 and 2 show the prediction results in terms of MSE (both mean values and standard deviation). From these results, it can be said that soft-computing techniques can be successfully applied to the addressed problem, being reliable techniques to estimate the target indicators. More precisely, for all the analysed indicators, the RBFN model is the one obtaining the lowest error rates and hence it can be successfully applied to predict them. As a result, yield prediction could be successfully carried out, reducing the cost of estimating it.

When considering the indicators individually, it can be concluded that all of them could be successfully estimated by RBFN, with error rates below 1E–06. However, there are differences from some indicators to the other ones regarding the error. For most indicators (“Tubers”, “Humidity”, “Caliber 0–40”, “Caliber 80”, and “Nitrogen”) the error rates are extremely low (below 1E–09) while it is higher for the “Weight” indicator.

Since the study validates the proposal of doing regression to predict the tuber indicators, authors consider that it could also be applied for the imputation of missing values on this fields. As a result, validating these methods to obtain more complete and valuable real datasets would be explored in the near future.

Acknowledgements. Part of this work was funded by a grant agreement between Lab-Ferrer and UBUCOMP. Authors are grateful to the farmer Mr. José María Izquierdo for providing the experimental field and the monitoring of irrigation. Special thanks to Ms. Mercedes Yartu, whose master thesis contributed to the data gathering.

References

1. van Klompenburg, T., Kassahun, A., Catal, C.: Crop yield prediction using machine learning: a systematic literature review. Comput. Electron. Agric. **177**, 105709 (2020). https://doi.org/10.1016/j.compag.2020.105709
2. Darwin, B., Dharmaraj, P., Prince, S., Popescu, D.E., Hemanth, D.J.: Recognition of bloom/yield in crop images using deep learning models for smart agriculture: a review. Agronomy **11**(4), 646 (2021). https://doi.org/10.3390/agronomy11040646
3. Potato production worldwide. Statista. https://www.statista.com/statistics/382174/global-potato-production/
4. Cromme, N., Prakash, A.B., Lutaladio, N., Ezeta, F.: Strengthening potato value chains: technical and policy options for developing countries. In: Workshop. Rome (Italy). Nov. 2008. Rome (Italy). Food and Agriculture Organization of the United Nations (FAO). ISBN 978-92-5-106627-0, p. 147 (2010). https://www.fao.org/docrep/013/i1710e/i1710e.pdf
5. Rady, A.M., Guyer, D.E.: Rapid and/or nondestructive quality evaluation methods for potatoes: a review. Comput. Electr. Agric. **117**, 31–48, (2015). https://doi.org/10.1016/j.compag.2015.07.002. ISSN 0168–1699
6. Lee, H.-S., Shin, B.-S.: Potato detection and segmentation based on mask R-CNN. J. Biosyst. Eng. **45**(4), 233–238 (2020). https://doi.org/10.1007/s42853-020-00063-w
7. Yale, U.O.: Multiple Linear Regression (2017). http://www.stat.yale.edu/Courses/1997-98/101/linmult.htm
8. Lippmann, R.P.: Pattern classification using neural networks. IEEE Commun. Mag. **27**(11), 47–50 (1989)
9. Pal, S.K.S.: Mitra, Multilayer Perceptron, Fuzzy Sets, Classifiaction (1992)
10. Pantazi, X.E., Moshou, D., Alexandridis, T., Whetton, R.L., Mouazen, A.M.: Wheat yield prediction using machine learning and advanced sensing techniques. Comput. Electron. Agric. **121**, 57–65 (2016)
11. Hara, P., Piekutowska, M., Niedbała, G.: Selection of independent variables for crop yield prediction using artificial neural network models with remote sensing data. Land **10**(6), 609 (2021)

12. Crusiol, L.G., Sun, L., Sibaldelli, R.N., et al.: Strategies for monitoring within-field soybean yield using Sentinel-2 Vis-NIR-SWIR spectral bands and machine learning regression methods. Precis. Agric. **23**(3), 1093–1123 (2022)
13. Huete, A., Didan, D., Miura, T., Rodriguez, E.P., Gao, X., Ferreira, L.G.: Overview of the radiometric and biophysical performance of the MODIS veg-etation indices. Rem. Sens. Environ. **83**(1–2), 195–213 (2002). ISSN 0034-4257
14. Fleisher, D.H., et al.: A potato model intercomparison across varying climates and productivity levels. Glob. Chang. Biol. **23**, 1258–1281 (2017)
15. Baker, K.M., Kirk, W.W.: Comparative analysis of models integrating synoptic forecast data into potato late blight risk estimate systems. Comput. Electr. Agric. **57**, 23–32 (2007)
16. Al-Gaadi, K.A., Hassaballa, A.A., Tola, E., Kayad, A.G., Madugundu, R., Alblewi, B., et al.: Prediction of potato crop yield using precision agriculture techniques. PLoS One **11**(9), e0162219 (2016)
17. Piekutowska, M., et al.: The application of multiple linear regression and artificial neural network models for yield prediction of very early potato cultivars before harvest. Agronomy **11**, 885 (2021)
18. Li, D., et al.: Improving potato yield prediction by combining cultivar information and uav remote sensing data using machine learning. Rem. Sens. **13**, 3322 (2021). https://doi.org/10.3390/rs13163322
19. Bishop, C.M.: Neural Networks for Pattern Recognition (1995)
20. Pal, S.K., Mitra, S.: Multilayer Perceptron, Fuzzy Sets, and Classification (2011)
21. Hecht, N.: Theory of the backpropagation neural network. In: IEEE Conference Publication, vol. 1, pp. 583–605 (1989)
22. MacKay, D.J.C.: Bayesian interpolation. Neural Comput. **4**(3), 415–447 (1992)
23. Park, J., Sandberg, I.W.: Universal approximation using radial-basis-function networks. Neural Comput. **3**(2), 246–257 (2008). https://doi.org/10.1162/neco.1991.3.2.246
24. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Statist. Comput. **14**(3), 199–222 (2004)
25. Vapnik, V.: The Nature of Statistical Learning Theory, Springer Science & Business Media (1999)
26. Yartu, M., Cambra, C., Navarro, M., Rad, C., Arroyo, Á., Herrero, Á.: Neural models to predict irrigation needs of a potato plantation. In: Herrero, Á., Cambra, C., Urda, D., Sedano, J., Quintián, H., Corchado, E. (eds.) SOCO 2020. AISC, vol. 1268, pp. 600–613. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-57802-2_58
27. Arlot, S., Celisse, A.: A survey of cross-validation procedures for model selection. Statist. Surv. **4**, 40–79 (2010)
28. Das, K., Jiang, J., Rao, J.: Mean squared error of empirical predictor. Ann. Stat. **32**(2), 818–840 (2004)

Reliability-Sensitive Optimization for Provision of Ancillary Services by Tempo-Spatial Correlated Distributed Energy Resources

Payam Teimourzadeh Baboli(✉), Amin Raeiszadeh, Michael Brand, and Sebastian Lehnhoff

Resilient Monitoring and Control R&D Group, Energy Division, OFFIS – Institute for Information Technology, Oldenburg, Germany
{payam.teimourzadehbaboli,amin.raeiszadeh,michael.brand,
sebastian.lehnhoff}@offis.de

Abstract. In this paper, a reliability-sensitive optimization approach for the provision of ancillary services from distributed energy resources is proposed. The main focus here is on small-scale renewable energy sources such as wind turbines and PV arrays, which are highly distributed, and the generated power is weather dependent and intermittent. Therefore, providing a reliable ancillary service through these unreliable resources is a complicated problem addressed here. In this paper, the tempo-spatial correlation between these renewable resources is mathematically modeled by pair-copula functions and included in the reliability evaluation procedure using the non-sequential Monte Carlo simulation method. An optimization problem is formulated using proposed joint reliability model to find the minimum number of renewable resources to provide ancillary services within a desired reliability level. The proposed optimization model is applied to real wind farms in Lower Saxony in Germany to provide the spinning reserve service. The results prove that including the correlation concept in the reliability evaluation procedure leads to the more realistic scenarios that play an essential role in maintaining the service's reliability level.

1 Introduction

Wind power forecasting plays an important role for power and ancillary services market players and operators. Most of the current probabilistic forecasting techniques consider the uncertainties of each wind turbine separately and eliminate the spatial and inter-temporal dependency of forecast errors [1]. However, wind power forecast errors tend to propagate in space and time, and considering such tempo-spatial dependence is crucial for a wide range of decision-making problems in energy systems. To make optimal decision making, it is necessary to have forecast uncertainty information in the form of scenarios [2].

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 226–235, 2023.
https://doi.org/10.1007/978-3-031-18050-7_22

Tempo-spatial dependence features should be included in a proper scenario generation method, which is not achievable through individual probabilistic forecasting. Wind energy scenario generation methods are classified into the following main groups: *sampling-based* [3], *forecast-based*, [4] *optimization-based*, [5] and *correlation-based* [6]. A comparative survey on existing methods of various scenario generation methods are carried out in [2], and concluded that *copula-based* correlation modeling is more suitable for the power system operation problems.

The copula theory is frequently used to model the joint distributions by modeling the dependence structure from its marginal distributions [7]. Therefore, *copula-based* scenario generation has the benefit of including spatial or temporal dependence in existing probabilistic forecasting methods, such as quantile regression. One of the first studies to consider the time-dependent concept is [8]. This method employs the evolving covariance matrix of the time blocks to generate temporal scenarios and has been known as a benchmark. In [9], pair-copula constructions are proposed to model the time dependence and the spatial correlation is ignored. In [9], the performance of C-Vine and D-Vine models are compared, and it is shown that D-Vine structure performs better in most of the use-cases. In the regular Vine (R-Vine) structure the number of possible trees is high compared to the D-Vine and C-Vine. However, the sequential estimation algorithm [10] could somehow solve this issue and makes the computational burden similar to the C-Vine and D-Vine, but the accuracy of the results are not so much better. Few studies have examined the use of Vine structures in power system applications [9]. The spatial dependency between multiple wind farms is modeled using R-vine in [11], and a conditional sampling algorithm is proposed for generating spatial scenarios. In this paper, the D-Vine structure is used to model both the spatial and temporal dependence between wind turbines.

In this paper, the reliability-sensitive optimization is proposed is proposed based on the multidimensional correlations between wind turbines. Since there are a significant number of them in a distribution system, the scale of the problem would be enormous. The pair-copula function is a mathematical way to decompose the high-dimensional stochastic correlation problem into solvable equations. The first step in the proposed method is to create proper D-Vine structures. The second step is to create the parametric pair-copula decomposition formulation based on the number of dependent variables, evaluate different possible copula families and choose the best one. Then, the correlation model is used in the non-sequential Monte Carlo reliability assessment, and the joint reliability indices are calculated for each wind turbine. The optimal coalition of wind turbines is optimally selected, taking into account the calculated joint-reliability indices and the desired reliability value of the provision of the spinning reserve service. The details of multivariate correlation modeling are covered in Sect. 2, and Sect. 3 explains the Procedure of reliability-sensitive optimization. In Sect. 4, a simulation study is performed to evaluate effectiveness of the proposed model and Sect. 5 concludes the paper.

2 Multivariate Correlation Modeling

Let's consider p_t as the forecasted power generated by a wind turbine at time t. There is always a forecast error, which could be defined as $e_t = p_t - \hat{p}_t$, where $\hat{p}_t$ is the actual power generated at time t. Considering the correlation between the generated power of some wind turbines in the same area, e_t could be modeled to include joint correlations instead of the individual forecast. The problem in this section is to model the multivariate correlation on the forecast error. From Sklar's theorem, the joint cumulative distribution function (CDF) of a joint distribution function $F(x_1, \ldots, x_n)$ can be expressed using the correlation coefficient matrix ρ and the marginal distribution functions $F(x_1), ..., F(x_n)$ [12]:

$$F(x_1, \ldots, x_n) = C\big(F(x_1), \ldots, F(x_n), \rho\big) \tag{1}$$

where $C(\cdot)$ is the copula function. Using chain rule, the joint probability density function (PDF) f can be decoupled into the n-variate copula density function $c(\cdot)$ and marginal densities $f_1(x_1), \ldots, f_n(x_n)$.

$$\begin{aligned} f(x_1, x_s, \ldots, x_n) = c\big(F_1(x_1), F_2(x_2), \ldots, F_n(x_n)\big) \\ \times f_1(x_1) \times f_2(x_2) \times \cdots \times f_n(x_n) \end{aligned} \tag{2}$$

If $u_n = F_n(x_n)$, then the copula density function is:

$$c(u_1, ..., u_n) = \frac{\partial C(u_1, u_2, \ldots, u_n)}{\partial u_1 \times \partial u_2 \times \cdots \times \partial u_n} \tag{3}$$

2.1 Pair-Copula Construction

Although a multivariate copula function could be decoupled using Sklar's theorem, it would be complicated for high-dimensional cases. So a decoupling technique is needed to decompose the multidimensional copulas into two-dimensional copula functions. If $X = (X_1, X_2, ..., X_n)$ defines wind power forecast errors, the joint PDF $f(x_1, x_2, ..., x_n)$ can be decoupled using the conditional distribution functions:

$$\begin{aligned} f(x_1, x_2, ..., x_n) =& f_n(x_n) \times f(x_{n-1}|x_n) \\ & \times f(x_{n-2}|x_{n-1}, x_n) \times \cdots \times f(x_1|x_2, \ldots, x_n) \end{aligned} \tag{4}$$

By Sklar's theorem, all conditional densities can be replaced by multiplying conditional bivariate copula density by marginal densities, *e.g.* $f(x_1|x_2)$ is equal to:

$$f(x_1|x_2) = c_{12}\big(F_1(x_1), F_2(x_2)\big) \times f_1(x_1) \tag{5}$$

where $c_{12}(\cdot, \cdot)$ defined as *pair-copula function* which could decouple a bivariate density function to its marginals. The above equation can be generalized for n-dimensional case as:

$$f(x|\nu) = c_{x\nu_j|\nu_{-j}}\big(F(x|\nu_{-j}), F(\nu_j|\nu_{-j})\big) \times f(x|\nu_{-j}) \tag{6}$$

where ν_j represents the j-th component in vector ν, and ν_{-j} is the vector ν, without ν_j. So each conditional density can be decomposed into the bivariate copula and marginal densities. Bivariate copula density arguments can also be obtained from the bivariate copula distribution:

$$F(x|\nu) = \frac{\partial C_{x,\nu_j|nu_{-j}}\big(F(x|\nu_{-j}), F(\nu_j|\nu_{-j})\big)}{\partial F(\nu_j|\nu_{-j})} \tag{7}$$

where $C_{a,b|c}$ is a bivariate copula function. So it is possible to calculate the pair-copula functions for high-dimensional correlation modeling problems and implemented them by commercial Python and R libraries.

2.2 D-Vine Copula Structure

While the conditional CDF arguments of the copula in (6) are expressed by (7), a ptoper structure of bivariate copula must be made at each step. Indeed, all possible permutations decomposing $f(\cdot)$ are not a valid pair-copula construction. There are many possible ways to construct pair-copula for multivariate distributions, *e.g.*, 240 various constructions for 5-variate density [13]. For better organization, *regular vines* are introduced as a graphical model for the construction. Regular vines are generally classified into two clusters, *i.e.*, *C-vine* and *D-vine*. Each offers a different decomposition technique using a nested set of trees. In this paper, D-vine construction is used. For details on the basics and definitions of vine trees, see [10]. For a D-vine structure there are $d-1$ trees, each edge represents a conditional bivariate copula $C_{j_e,k_e|D(e)}$ between the variables X_{j_e} and X_{k_e} conditional on the variables with subscripts in $D(e)$, where j and k are the nodes corresponding to the edge e. D-vine copula with d nodes is expressed as:

$$c(\bar{u}) = \prod_{i=1}^{d-1} \prod_{e \in E_i} C_{j_e,k_e|D(e)}\big(u_{j_e|D(e)}, u_{k_e|D(e)}\big) \tag{8}$$

where E_i is the set of edges in i-th tree. Vine copula modeling of a sample dataset involves three steps: (1) selecting the best Vine tree structure, (2) selecting the best fitting bivariate copula family, and (3) estimating the parameters of each copula. The sequential estimation method is proposed to construct the vine tree structure by sorting the pairs of variables based on the highest dependency in the first tree and then proceeding with copula selection and estimation [14]. Then, two additional steps should be done to optimize the vine copula model: (1) Select suitable pair copula families using *Akaike* information criterion [10]; and (2) Estimate the parameters of all copula families by *maximum likelihood* estimation method [15]. In this paper, the same procedure as [6] has been used for selecting the proper copula families and estimating its parameters.

3 Reliability-Sensitive Optimization

In this section, the correlation model of Sect. 2 is used to generate cross-correlated scenarios, then it has been explained how these samples are included in joint reliability evaluation and optimization problem. The general procedure of the problem is shown in Fig. 1, which is explained in the following subsections.

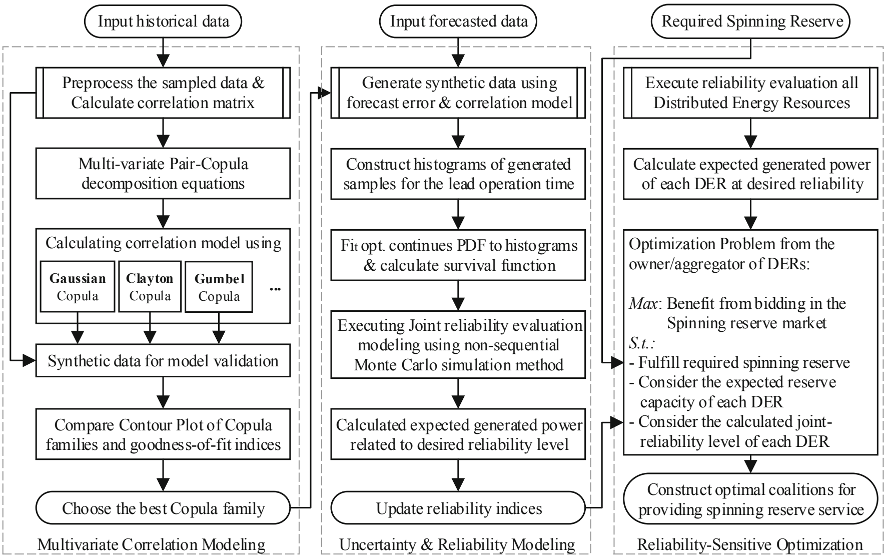

Input historical data
Preprocess the sampled data & Calculate correlation matrix
Multi-variate Pair-Copula decomposition equations
Calculating correlation model using
Gaussian Copula
Clayton Copula
Gumbel Copula
...
Synthetic data for model validation
Compare Contour Plot of Copula families and goodness-of-fit indices
Choose the best Copula family
Multivariate Correlation Modeling
Input forecasted data
Generate synthetic data using forecast error & correlation model
Construct histograms of generated samples for the lead operation time
Fit opt. continues PDF to histograms & calculate survival function
Executing Joint reliability evaluation modeling using non-sequential Monte Carlo simulation method
Calculated expected generated power related to desired reliability level
Update reliability indices
Uncertainty & Reliability Modeling
Required Spinning Reserve
Execute reliability evaluation all Distributed Energy Resources
Calculate expected generated power of each DER at desired reliability
Optimization Problem from the owner/aggregator of DERs:
Max: Benefit from bidding in the Spinning reserve market
S.t.:
- Fulfill required spinning reserve
- Consider the expected reserve capacity of each DER
- Consider the calculated joint-reliability level of each DER
Construct optimal coalitions for providing spinning reserve service
Reliability-Sensitive Optimization

Fig. 1. Proposed algorithm for multivariate correlation, joint-reliability modeling and reliability-sensitive optimization.

3.1 Multivariate Correlation Modeling

The main steps for multivariate correlation modeling are: (1) historical and forecast data preparation and acquisition, (2) performing uncertainty modeling and providing an appropriate error model for the forecast module of each individual wind turbine, (3) constructing the Vine structure and decompose the multivariate correlation problem to solvable bivariate copula equations using pair-copula functions, (4) selection of most suitable copula families. These steps are explained in Sect. 2 in detailed.

3.2 Joint Reliability Evaluation Methodology

In this paper, the simulation-based method is used for reliability evaluation and the states are modeled using the non-sequential Monte Carlo simulation method. The main steps are (1) generation of synthetic data to provide correlated samples for the simulation of the upcoming generation scenarios, (2) creating histograms based on scenarios and fitting them to optimal PDF, (3) calculate the joint-reliability model. The reliability level of a wind turbine or the probability of each generation level can be calculated based on its forecast error model. This probability value indicates the minimum deviation of x from the forecast value. The $P(X \geq x)$ shows the probability that at least x kW would be generated by one specific wind turbine. It can be computed directly from the PDF by calculating the area under the PDF curve for $(X \geq x)$. Or using survival function, which directly gives the reliability level.

4 Simulation Study

Historical data of wind power generation in five cities in the Lower Saxony state, Germany, *i.e.*, Cloppenburg, Varel, Oldenburg, Butjadingen, and Walsrode are used in this simulation study from Jan. 2017 to Jan. 2020. The wind turbines are Vestas V150-5.6 with capacities between 4 and 5 MW and are installed at 125 to 155 m. Figure 2 shows the locations, some of the distances from each other, and their pair-wise correlation graph, in which correlation coefficients are also addressed. The correlation coefficients provide a visualization view for their spatial dependency and help assess the system's behavior before executing the copula-based correlation modeling. The larger correlation coefficient indicates a more positive dependency. As can be seen, the distance varies between 30 and 124 km, which is not so far to be independent and not so close as to be too dependent.

The fitted 5-variate pair-copula function models the uncertainty scenarios and generates cross-correlated synthetic data around the short-term forecast values. Therefore, the forecast data is available, and shown in Fig. 3. The dash line with red cross shows the forecast values of the wind generation in Oldenburg for the next 10 h of the system operational horizon in a test day. A copula-based scenario generation is employed to incorporate the dependency characteristics between wind turbines into the error models of their individual forecast models by generating synthetic data. The hourly generated samples for each site are then converted into histograms. Histograms are fitted to optimal continuous density functions to facilitate their integration into the scenario generation stage. The fitted density functions are shown in Fig. 3 with error boxes. As can be seen, this is not symmetrical, and normal distribution cannot be a suitable PDF and an optimal distribution function should be fitted in each hour.

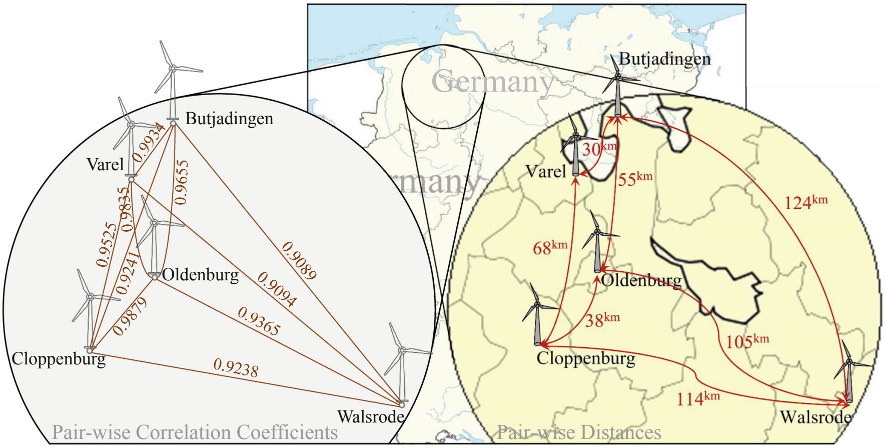


Fig. 2. The location of the five considered wind turbines on the map, including the pair-wise distances and correlation coefficients.

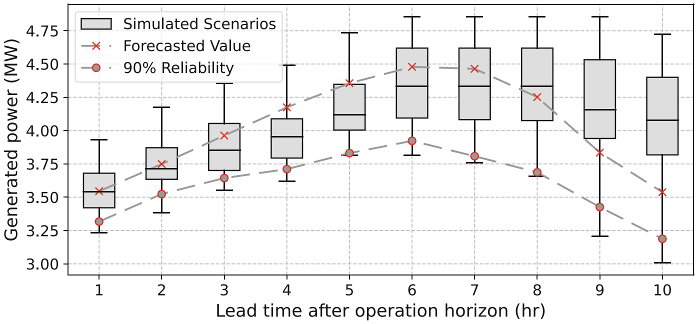


Fig. 3. Fitted density functions, wind generation forecast, and expected generation at 90% reliability level.

The most significant contribution of this paper is introduction of reliability-sensitive optimization using the tempo-spatial correlation models. The proposed reliability model is applied to the fitted density functions, and two indices are calculated. The first index is the joint-reliability level of the predicted values. The range of this index is from 0 to 1, and a higher value indicates a higher level of reliability. The forecast values are shown in Fig. 3, together with the density functions for upcoming hours. As can be seen, the reliability of the power generation at the forecast value is different in different hours, *e.g.*, the reliability levels at hours 3 and 9 are 0.41 and 0.62, respectively. It is even possible to define a predefined reliability level as a constrain and calculate its corresponding generated power. Figure 3 shows also the generated power of the wind turbine in Oldenburg with a reliability level of 90% or 0.90. This value indicates the wind generation at least x MW in more than 90% of scenarios.

This study considered that these wind turbines, located in 5 different cities, are operated by the same aggregator participating in the ancillary services market to provide spinning reserve. In order to focus more on the main contribution of the paper, which is the consideration of the temporal-spatial correlation between wind turbines and their impacts on the reliability modeling, the amount of spinning reserve to be provided is assumed as an input value and considered to be 1.2 MW for the next 10 h of the operating point. Furthermore, it is considered that this aggregator assigns 10% of the expected generated power of each wind turbine to the auxiliary market and the remaining 90% to the energy market. The generation costs of wind turbines are also assumed differently and in ascending order could be listed as Walsrode, Cloppenburg, Varel, Oldenburg, and Butjadingen.

Figure 4 shows each wind turbine's expected spinning reserve capacity for the next 10 h, sorted by generation cost from low to high. As seen in this figure, in some operating times, such as 1, 6, 7, and 8, the three cheapest wind turbines are able to provide the desired spinning reserve. For the rest of the time, four wind turbines should be involved. In order to assess the influence of the tempo-spatial correlations, two other scenarios, i.e., high and low correlations, are considered. In the high correlation scenario, which is addressed in Fig. 5, the fluctuation of the expected wind generations in different locations is similar. However, in the low-correlation scenario in Fig. 6, wind generation changes independently. As seen in a high-correlated scenario, at least four wind turbines are needed to meet the given spinning reserve, which decreases the aggregator's benefit. However, in the low-correlated scenario, the spinning reserve could be provided at a lower cost and consequently at a higher benefit.

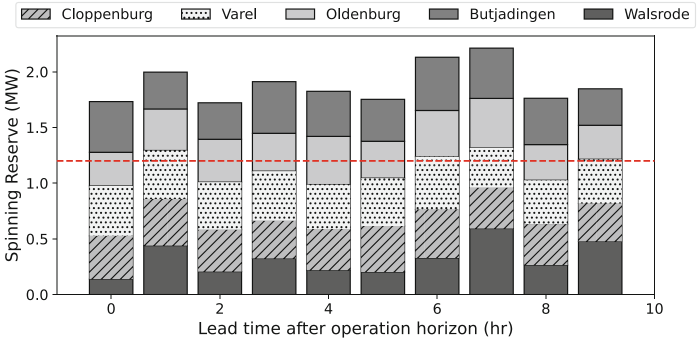


Fig. 4. Optimal coalitions for providing 1.2 MW spinning reserve using the real data.

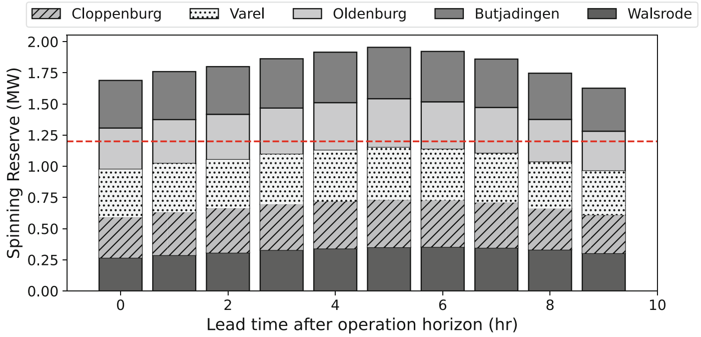


Fig. 5. Optimal coalitions for providing 1.2 MW spinning reserve using the high-correlated synthetic data.

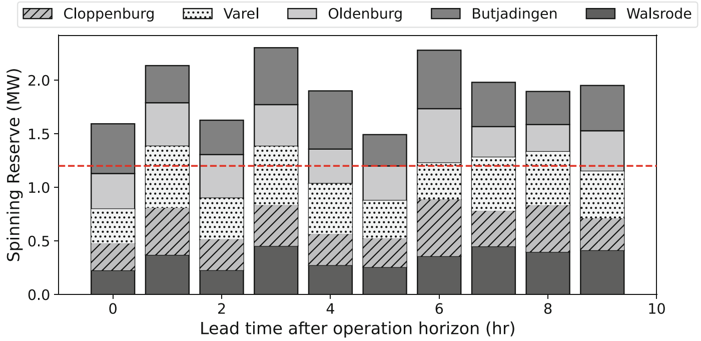


Fig. 6. Optimal coalitions for providing 1.2 MW spinning reserve using the low-correlated synthetic data.

5 Conclusion

This paper addressed the problem of providing reliable ancillary services from unreliable DERs. An algorithm is developed to optimally compute the parameters of the multivariate copula functions in the D-Vine structure. It was considered that each wind turbine is equipped with a forecasting module, and the short-term forecast values are available. Cross-correlated scenarios around the forecast values are generated by including the spatial-temporal correlation in the forecast module's error model. These scenarios are then integrated into the joint-reliability assessment to run the reliability-sensitive optimization problem. As shown in the simulation study, it is possible to assign a reliability level to the forecast value, calculate the expected wind generation with a certain reliability level, and construct a coalition to provide reliable ancillary service. The results showed that including the correlation concept in reliability-sensitive optimization plays an important role in the ancillary modeling of the system. It showed that if a proper correlation is not considered in the problem, the results would not be so accurate and may lead to failure in providing the desired ancillary service.

Acknowledgements. This research has been funded by the Lower Saxony Ministry of Science and Culture through the 'Niedersächsisches Vorab' grant programme (grant ZN3563) and of the Energy Research Centre of Lower Saxony through the research project SiNED - Systemdienstleistungen für sichere Stromnetze in Zeiten fortschreitender Energiewende und digitaler Transformation.

References

1. Ni, L., et al.: Vine copula selection using mutual information for hydrological dependence modeling. Environ. Res. **186**, 109604 (2020)
2. Li, J., Zhou, J., Chen, B.: Review of wind power scenario generation methods for optimal operation of renewable energy systems. Appl. Energy **280**, 115992 (2020)
3. Zhang, N., Kang, C., Xia, Q., Liang, J.: Modeling conditional forecast error for wind power in generation scheduling. IEEE Trans. Power Syst. **29**(3), 1316–1324 (2013)
4. Chen, Y., Wang, Y., Kirschen, D., Zhang, B.: Model-free renewable scenario generation using generative adversarial networks. IEEE Trans. Power Syst. **33**(3), 3265–3275 (2018)
5. Li, J., Lan, F., Wei, H.: A scenario optimal reduction method for wind power time series. IEEE Trans. Power Syst. **31**(2), 1657–1658 (2015)
6. Teimourzadeh Baboli, P., Brand, M., Lehnhoff, S.: Stochastic correlation modelling of renewable energy sources for provision of ancillary services using multi-dimensional copula functions. In: 2021 11th Smart Grid Conference (SGC), IEEE, 2021, pp. 1–6 (2021)
7. Li, B., et al.: Probabilistic analysis to analyze uncertainty incorporating copula theory. J. Electr. Eng. Technol. **17**, 1–11 (2021)
8. Pinson, P., Madsen, H., Nielsen, H.A., Papaefthymiou, G., Klöckl, B.: From probabilistic forecasts to statistical scenarios of short-term wind power production. Wind Energy **12**(1), 51–62 (2009)
9. Becker, R.: Generation of time-coupled wind power infeed scenarios using pair-copula construction. IEEE Trans. Sustain. Energy **9**(3), 1298–1306 (2017)
10. Dissmann, J., Brechmann, E.C., Czado, C., Kurowicka, D.: Selecting and estimating regular vine copulae and application to financial returns. Comput. Stat. Data Anal. **59**, 52–69 (2013)
11. Wang, Z., Wang, W., Liu, C., Wang, Z., Hou, Y.: Probabilistic forecast for multiple wind farms based on regular vine copulas. IEEE Trans. Power Syst. **33**(1), 578–589 (2017)
12. Sun, M., Feng, C., Zhang, J.: Conditional aggregated probabilistic wind power forecasting based on spatio-temporal correlation. Appl. Energy **256**, 113842 (2019)
13. Vatter, T., Nagler, T.: Generalized additive models for pair-copula constructions. J. Comput. Graph. Stat. **27**(4), 715–727 (2018)
14. Kraus, D., Czado, C.: D-vine copula based quantile regression. Comput. Stat. Data Anal. **110**, 1–18 (2017)
15. Zhang, J., Gao, K., Li, Y., Zhang, O.: Maximum likelihood estimation methods for copula modelsa. Computat. Econ. **60**, 1–26 (2021)

Special Session on Machine Learning and Computer Vision in Industry 4.0

Predictive Maintenance of ATM Machines by Modelling Remaining Useful Life with Machine Learning Techniques

Riccardo Rosati[1(✉)], Luca Romeo[1,4], Víctor Manuel Vargas[2], Pedro Antonio Gutiérrez[2], César Hervás-Martínez[2], Lorenzo Bianchini[3], Alessandra Capriotti[3], Rosario Capparuccia[3], and Emanuele Frontoni[1,4]

[1] VRAI Lab - Department of Information Engineering, Università Politecnica delle Marche, Via Brecce Bianche, 60131 Ancona, Italy
r.rosati@pm.univpm.it

[2] Department of Computer Science and Numerical Analysis, University of Córdoba, Campus Universitario de Rabanales, 14014 Córdoba, Spain
{vvargas,pagutierrez,chervas}@uco.es

[3] Sigma S.p.A., 63824 Altidona, Italy

[4] Università degli Studi di Macerata, 62100 Macerata, Italy
{luca.romeo,emanuele.frontoni}@unimc.it

Abstract. One of the main relevant topics of Industry 4.0 is related to the prediction of Remaining Useful Life (RUL) of machines. In this context, the "Smart Manufacturing Machine with Predictive Lifetime Electronic maintenance" (SIMPLE) project aims to promote collaborations among different companies in the scenario of predictive maintenance. One of the topics of the SIMPLE project is related to the prediction of RUL of automated teller machines (ATMs). This represents a key task as these machines are subject to different types of failure. However the main challenges in this field lie in: i) collecting a representative dataset, ii) correctly annotating the observations and iii) handling the imbalanced nature of the dataset. To overcome this problem, in this work we present a feature extraction strategy and a machine learning (ML) based solution for solving RUL estimation for ATM devices. We prove the effectiveness of our approach with respect to other state-of-the-art ML approaches widely employed for solving the RUL task. In addition, we propose the design of a predictive maintenance platform to integrate our ML model for the SIMPLE project.

1 Introduction

In recent years, in the context of Industry 4.0 and intelligent manufacturing, there has been increasing emphasis on the predictive maintenance (PdM) task [1,7,14], which aims to estimate when a machine might fail in order to schedule corrective maintenance operations before the point of failure. Data-driven algorithms for PdM imitate the normal data behaviour of a machine and use it as a baseline to identify and report deviations in real-time. A machine

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 239–249, 2023.
https://doi.org/10.1007/978-3-031-18050-7_23

monitoring system includes input data (time-series) on a range of factors, e.g. from temperature to pressure. The output is the desired target, i.e. a warning of a future failure or the remaining useful life (RUL) of the tool or machinery [12]. The algorithm will then be able to predict when a failure is likely to occur or estimate the life time of the machine. In the literature, the two main machine learning (ML) strategies to solve these tasks are supervised and unsupervised methodologies. The two categories of approaches may be relevant for a different scenario and depend on the availability of sufficient historical training data and the frequency of equipment failure [5].

In this context, the "Smart Manufacturing Machine with Predictive Lifetime Electronic maintenance" (SIMPLE) project was promoted, involving various companies and universities in the Marche region (Italy). The aim of the project is to create innovative products that can be monitored and controlled both locally and remotely, able to implement PdM logics, connected to a new flexible platform. One of the topic of SIMPLE project is related to the prediction of RUL of automated teller machines (ATMs) manufactured by Sigma company. This represents a key task as these machines are subject to different types of failures, which are difficult to predict by maintenance staff. From a practical perspective, identifying when the next failure might occur is a relevant aspect for significantly reducing maintenance costs and avoiding lack of service delivery for long periods of time. However the main challenges in this field lie in: i) collecting a representative dataset, ii) correctly annotating the observations and iii) handling the imbalanced nature of the dataset.

Starting from this gap in the current state-of-the-art, our work aims to propose a scalable decision support system for solving the RUL task powered by a feature engineering stage and a ML algorithm. The main contributions of the present work can be summarized as follows:

- to design a feature engineering stage, performed in collaboration with domain expert maintainers, that ensures to build off a representative dataset;
- to design an efficient ML strategy to predict the RUL of multiple ATMs;
- to integrate the algorithm in a scalable cloud-based architecture as the main core of a decision support system.

2 Materials

ATM devices generate a very large amount of logs associated to each individual machine. These files can contain all the information about the operation of individual devices and are normally written in verbose mode to allow the analysis of any abnormal behaviour. However, they are generally not required to be structured according to strict rules. In the case of the SIMPLE project, the logging process was rationalised by implementing an original log management solution with aggregation and storage features, secure transfer, automatic parsing and transformation of logs into data. In addition, the data of the technical interventions was extracted, both as a result of calls for malfunctions and linked to preventive maintenance activities, from Sigma's ticket management system. The information coming from the logs and the maintenance server was then reconciled by means of automatic operations (batches specially developed to correlate the data by machines/dates) and manual operations (for labeling the types of faults found after interpreting the unstructured notes of the technician at the time of the intervention).

Table 1. Characteristics of the ATMs dataset.

	Machines	# Observations
Total	89	15254
	Failures × machine	# Observations × cycle of fault
Max	8	533
Min	0	1
Mean	2	82
Std	1.8	91

The collected dataset represents a total of 89 different machines, and it is built on raw ATM logs which contain information about several sub-devices. In particular, attention was focused on three different devices installed in the ATM: Cash Recycling Module (CRM), badge reader and receipt printer. The CRM is the most complex device within the ATM as it consists of many mechanical parts that must be driven and maintained in the best possible way to avoid problems when handling money. The dataset resulting from the data extracted through the parsing of the logs is composed of 236 features and a target variable corresponding to the information on the failure event. A failure can be caused by a wide variety of factors, making the prediction task very difficult: it may be the result of a component degradation or a sudden and unpredictable event (such as a jam of a crumpled banknote). Input features are data about opening and closing intervals of the device shutter, number of processed banknotes, number and types of movements in each zone of the device and associated time to complete, number and type of low level error code occurred, information about standard maintenance execution (cleaning of parts or their substitution), labels about abnormal status of single parts observed during maintenance interventions, etc. The high number of features is a consequence of the complexity of the CRM device, which must be able to dispense banknotes, both single and bundled, from the ATMs, and therefore it requires extremely sophisticated mechanics, control/implementation sensors and management/monitoring software. The extracted features are aggregations on a daily basis of this data, while the labels are manually interpreted by the tickets of technical interventions. Statistics about dataset distribution are collected in Table 1.

The final dataset presents the following challenges:

- the data corresponds to time-series belonging to different machines with an extremely not homogeneous number of failures;
- there are a large number of variables with unknown correlation within the various types of faults;
- there are a limited number of failure cases;
- there is a possible data leakage issue in dataset preparation, since features are pre-aggregated and some information may be lost on generalization.

3 Methods

3.1 Task Definition

The problem of RUL estimation can be associated both with regression and classification tasks. In a regression approach, RUL is maintained as a continuous value in order to predict the exact remaining time before failure. In the context of ATMs maintenance, estimating the exact moment of failure is a very complex task according to company expertise; for this reason, it was considered more appropriate to treat the problem as a classification task. For the classification approach, the RUL value is converted into a discrete value to predict if the machine will fail within a certain time frame. Therefore two different supervised methodologies were evaluated for solving the task:

(a) Binary classification: we convert the RUL in a binary value (class 1 high-risk of failure, class 0 low-risk of failure). Label is considered as class 1 according to a time window of 6 days before failure;
(b) Multi-class classification: RUL is converted into 5 classes to predict machine failures in different time windows. In our experiments, we consider the following label encoding:
    - very high risk (class 5): number of days before failure between 1 and 3;
    - high risk (class 4): number of days before failure between 4 and 6;
    - medium risk (class 3): number of days before failure between 7 and 9;
    - low risk (class 2): number of days before failure between 10 and 14;
    - very low risk (class 1): number of days before failure between 15 and 19.

One of the main concerns approaching the problem as a classification task is related to the huge imbalance distribution of classes. For example, as regards the binary approach, considering the fault class as only the day 0 of RUL implies a small number of failures (and, from a practical perspective, no time to model and to prevent failures); on the other side, considering the fault class as several n days before the day 0 of RUL implies a high risk of training ML model on data which are not enough discriminative of failures (i.e. data that are too far from failures).

3.2 Feature Extraction and Selection

The domain experts suggested the most discriminative features that can be monitored in order to discriminate the RUL. These features represent daily statistical measures (i.e. max, min, std, mean, median, 1-st quartile, 3-rd quartile) derived from the original raw data. To delete redundant features, a standard correlation analysis was applied using a threshold of Pearson correlation value equals to 0.5 (with p-value < 0.05), which removed 135 features (Fig. 1a). Most of the removed features regard the type of error counters and some of the quartiles about the dispense preparation time and deposit time of banknotes. Then, feature importance by embedded methods highlighted the most relevant features, as shown in Fig. 1b.

3.3 Pre-processing

Pre-processing techniques such as normalization and outlier removal were applied together with oversampling/undersampling strategies to face the imbalance issue in the classification task [10]. In particular, the following strategies were selected:

- SMOTE [6] to increase the number of faults. The number of faults samples generated was the one needed to match the number of non-fault samples;
- Random undersampling [8] to decrease the number of non-fault samples (random selection of 5000 samples);
- SMOTE + Random undersampling [11]: for increasing and decreasing both classes to the same number of samples.

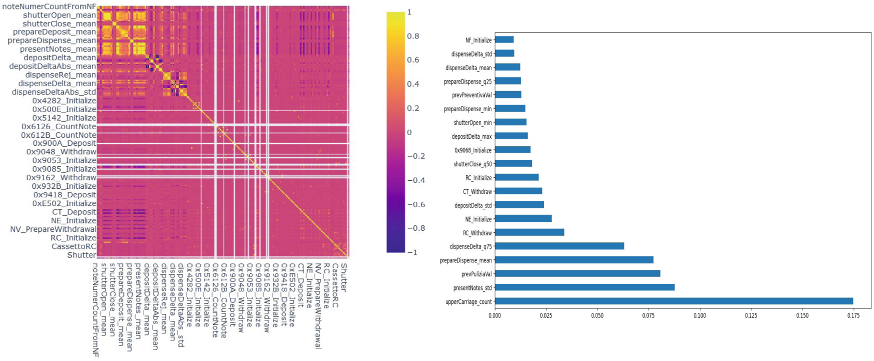


Fig. 1. Feature extraction and selection procedure: a) Pearson correlation analysis; b) a subset of the most important features.

3.4 Machine Learning Model

The selected ML algorithm to solve both binary and multi-class classification tasks is the Random Forest (RF) model. RF represents a variant of bagging proposed by [3] and consists of an ensemble of decision trees (DTs) generated by independent identically distributed random vectors. RF is modeled by sampling from the data observations and by changing tree-parameters in order to maximize the diversity among different trees. In classification problems, an ensemble of DTs is built, which aims to split the data into subsets that contain instances with similar values (homogeneous). For each subset, a random feature selection is carried out at each node of the tree. In addition to predictive performance, we have also to take into account another important factor for solving a PdM task, which is model interpretability. This requirement is satisfied by the RF algorithm, which allows providing a direct interpretation of the most discriminative features.

3.5 Experimental Procedure

We decided to compare our RF-based PdM approach with respect to other state-of-the-art ML approaches employed for solving PdM tasks. In particular we have considered the following models for comparing both binary and multi-class tasks:

- Logistic Regression (LR) [13];
- K-Nearest Neighbors (KNN) [15];
- Decision Tree (DT) classifier;
- Support Vector Machine (SVM) with Gaussian and linear kernel [2];
- SVM with Elastic Net penalty [2];
- Gaussian Naive Bayes (NB);
- XGBoost (XGB) [4].

All the ML models have been tested only with the preprocessed dataset. A 10-fold Cross Validation (10-CV) procedure stratified over machines and class values was performed in order to evaluate the performance of the RF model. Although this experimental procedure is computationally demanding, it ensures measuring the ability of the proposed algorithm to predict RUL across unseen machines. For each algorithm, the related hyperparameters were optimized by implementing a grid search in a nested 5-CV. Table 2 shows the different hyperparameters for the proposed ML models and all competitors' ML approaches, as well as the grid-search set. For LR, linear and gaussian SVM the λ penalty controls the 2-norm regularization. For the SVM Elastic Net $\alpha = \lambda_1 + \lambda_2$ and l1_ratio=$\frac{\lambda_1}{\lambda_1+\lambda_2}$ where λ_1 and λ_2 control separately the 1-norm and 2-norm regularizations. The main metrics selected for tuning model hyperparameters and evaluating the final models are the F1 score and the balanced accuracy (defined as the average of recall obtained on each class) [9], since we consider both True Positives and False Positives decisive factors for our task. Other metrics that have been computed are standard accuracy, precision, recall and mean absolute error (MAE).

Table 2. Range of hyperparameters (Hyp) for the proposed RF model and the other state-of-the-art ML approaches.

Model	Hyp	Range
RF	$n°$ of classification trees $n°$ of features to select max depth	$\{100, 200\}$ $\{2, 5, 10\}$ $\{25, 50, 100\}$
LR	λ	$\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$

KNN	$n°$ of neighbors
	Weight function
	distance metric
		$\{3, 5, 7, 9\}$
		{'uniform','distance'}
		{'euclidean','manhattan'}

DT	Split criterion max depth min $n°$ of leaf size	{'gini impurity','entropy'} $\{50, 100\}$ $\{1, 2, 3, 4, 5\}$
SVM_gaussian	λ	$\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$
SVM_elasticnet	α l1_ratio	$\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$ $\{0, 0.25, 0.5, 0.75, 1\}$
SVM_linear	λ	$\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$
NB	Variance smoothing	$\{10^{-9}, 10^{-8}, 10^{-7}\}$
XGBoost	Learning rate max $n°$ of estimators max depth $n°$ of features to select	$\{10^{-3}, 10^{-2}, 10^{-1}\}$ $\{50, 100, 200\}$ $\{50, 100\}$ $\{1, 2, 4\}$

4 Results

The predictive performance of the RF model is shown in Table 3 for binary task and Table 4 for multi-class approach. The values shown in tables are the mean across the respective metric resulting from the single CV folds. For the binary classification, it can be noted that the best prediction results were obtained by our RF in all metrics except recall. This trend is also confirmed by the best Area Under Area under the ROC Curve (AUC) value achieved by RF (see Fig. 2). Low recall values denote that the classifier presents a high number of false negatives, which may be due to the imbalanced distribution of the dataset. This is confirmed also by the F1-score, which is below 0.5. As regards the multi-class classification, it is possible to note how the model's performance significantly decreases. However, RF achieved the best results for F1 score and balanced accuracy, which are the most relevant metrics for our PdM task.

5 PdM Decision Support System for SIMPLE Project

The designed platform is composed of two server groups, each of which is composed of three servers and a Storage Area Network (SAN). The solution includes a hardware (HW) infrastructure consisting of two distinct server and storage

Table 3. Predictive performance of ML approaches for the binary approach.

Model	Accuracy	Precision	Recall	F1	BalAccuracy	MAE
LR	0.933	0.652	0.211	0.319	0.601	0.067
KNN	0.934	0.679	0.208	0.317	0.600	0.066
DT	0.825	0.183	**0.381**	0.244	0.618	0.175
SVM_gaussian	0.933	0.749	0.158	0.259	0.577	0.067
SVM_elasticnet	0.925	0.659	0.179	0.279	0.562	0.069
SVM_linear	0.927	0.535	0.216	0.305	0.601	0.073
NB	0.928	0.126	0.380	0.185	0.543	0.702
XGB	0.932	0.639	0.256	0.363	0.622	0.068
RF	**0.938**	**0.766**	0.243	**0.368**	**0.621**	**0.062**

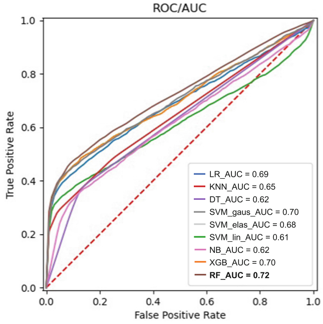


Fig. 2. ROC curves of ML methods for the binary approach.

Table 4. Predictive performance of ML approaches for the multi-class approach. In this case, binary metrics (precision, recall, F1) are averaged across classes.

Model	Accuracy	Precision	Recall	F1	BalAccuracy	MAE
LR	0.298	0.462	0.262	0.222	0.262	1.272
KNN	0.267	0.281	0.254	**0.256**	0.254	1.394
DT	0.297	**0.518**	0.258	0.173	0.258	1.151
SVM_gaussian	**0.304**	0.379	0.156	0.174	0.265	**1.132**
SVM_elasticnet	0.262	0.346	0.257	0.215	0.257	1.431
SVM_linear	0.302	0.317	**0.272**	0.247	0.262	1.294
NB	0.214	0.310	0.255	0.159	0.255	1.183
XGB	0.273	0.266	0.261	0.255	0.261	1.402
RF	0.291	0.294	0.266	**0.256**	**0.266**	1.343

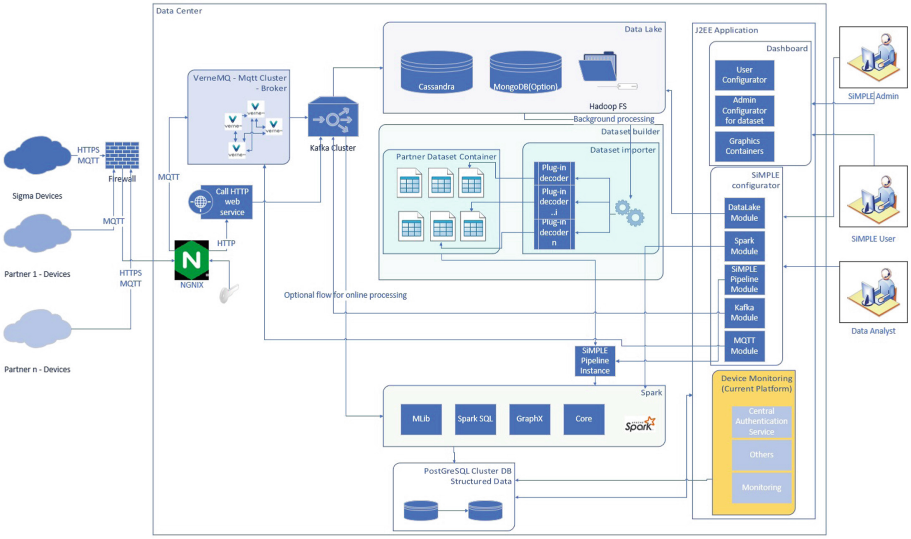


Fig. 3. System architecture diagram of predictive maintenance platform within SIMPLE project.

groups located in two different data centers, for ensuring the level of reliability required at the infrastructure level. The software technical specifications of the SIMPLE project led to the design of the system architectural scheme illustrated in Fig. 3. The platforms assure scalability and interactions with different PdM tasks and different companies enrolled in the SIMPLE project. Each task has their own constraints, characteristics and requirements, which have guided the technical choices on the communication protocol with the devices towards a solution that should be as generic as possible, involving the support of at least two protocols (MQTT and REST). A container-based deployment technology was the basic architectural solution employed, in order to quickly manage the scalability and the resources of containerised applications, check the integrity status of applications and manage corrections with automatic placement, restart, replication and scaling. For the orchestration and monitoring of each functional module identified (i.e. Nginx, VerneMQ, Cassandra...), the installation and configuration of the OKD (Kubernetes Container) platform on the identified HW infrastructure has been tackled. From this basis, the software solutions considered to be the most promising has been refined also according to the ML computing capacity required.

6 Conclusions

In this work we present a machine learning (ML) based solution for solving a predictive maintenance (PdM) task in an unexplored application, that is the remaining useful life (RUL) estimation for ATM devices. In particular, we proved how Random Forest (RF) algorithm is suitable to predict whether a machine will fail or not in the next 6 days. Decomposing the problem in multiple windows classification, the prediction task becomes more complex, even if RF remains the best model among other state-of-the-art ML algorithms. In addition, we propose the design of a PdM platform to integrate our ML model for the SIMPLE project. The container-based platform allows the integration of different ML algorithms that can be specifically conceived for solving different RUL tasks within a scalable and specific decision support system. However, approaching the RUL estimation as a nominal classification task leads to lose some relevant information about the risk of failure. Future work could be addressed by considering ML ordinal methodologies and recurrent neural networks to embrace both ordinal and temporal constraints in raw time series data.

Acknowledgements. This work was supported within the research agreement between Università Politecnica delle Marche and Sigma Spa for the project "Smart Manufacturing Machine with Predictive Lifetime Electronic maintenance (SIMPLE)" funded by Ministero dello Sviluppo Economico (Italia) - Fondo per la Crescita Sostenibile - Accordi per l'innovazione di cui al D.M. 24 maggio 2017. This work has been partially subsidised by "Agencia Española de Investigación (España)" (grant reference: PID2020-115454GB-C22/AEI/10.13039/501100011033), by "Consejería de Salud y Familia (Junta de Andalucía)" (grant reference: PS-2020-780) and by "Consejería de Transformación Económica, Industria, Conocimiento y Universidades (Junta de Andalucía) y Programa Operativo FEDER 2014-2020" (grant references: UCO-1261651 and PY20_00074). Víctor Manuel Vargas's research has been subsidised by the FPU Predoctoral Program of the Spanish Ministry of Science, Innovation and Universities (MCIU), grant reference FPU18/00358.

References

1. Ayvaz, S., Alpay, K.: Predictive maintenance system for production lines in manufacturing: a machine learning approach using IoT data in real-time. Expert Syst. Appl. **173**, 114598 (2021). https://doi.org/10.1016/j.eswa.2021.114598
2. Bilski, P.: Application of support vector machines to the induction motor parameters identification. Measurement **51**, 377–386 (2014). https://doi.org/10.1016/j.measurement.2013.12.013
3. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
4. Calabrese, M., et al.: Sophia: an event-based IoT and machine learning architecture for predictive maintenance in industry 4.0. Information **11**(4), 202 (2020). https://doi.org/10.3390/info11040202
5. Carvalho, T.P., Soares, F.A., Vita, R., Francisco, R.d.P., Basto, J.P., Alcalá, S.G.: A systematic literature review of machine learning methods applied to predictive maintenance. Comput. Ind. Eng. **137**, 106024 (2019). https://doi.org/10.1016/j.cie.2019.106024
6. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002). https://doi.org/10.1613/jair.953
7. Florian, E., Sgarbossa, F., Zennaro, I.: Machine learning-based predictive maintenance: a cost-oriented model for implementation. Int. J. Prod. Econ. **236**, 108114 (2021). https://doi.org/10.1016/j.ijpe.2021.108114

8. Hasanin, T., Khoshgoftaar, T.: The effects of random undersampling with simulated class imbalance for big data. In: 2018 IEEE International Conference on Information Reuse and Integration (IRI), pp. 70–79. IEEE (2018). https://doi.org/10.1109/IRI.2018.00018
9. Kelleher, J.D., Mac Namee, B., D'arcy, A.: Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies. MIT Press, Cambridge (2020)
10. Kotsiantis, S., Kanellopoulos, D., Pintelas, P., et al.: Handling imbalanced datasets: a review. GESTS Int. Trans. Comput. Sci. Eng. **30**(1), 25–36 (2006)
11. Mishra, S.: Handling imbalanced data: smote vs. random undersampling. Int. Res. J. Eng. Technol **4**(8), 317–320 (2017)
12. Nguyen, V., Seshadrinath, J., Wang, D., Nadarajan, S., Vaiyapuri, V.: Model-based diagnosis and RUL estimation of induction machines under interturn fault. IEEE Trans. Ind. Appl. **53**(3), 2690–2701 (2017). https://doi.org/10.1109/TIA.2017.2669195
13. Phillips, J., Cripps, E., Lau, J.W., Hodkiewicz, M.: Classifying machinery condition using oil samples and binary logistic regression. Mech. Syst. Sig. Process. **60**, 316–325 (2015). https://doi.org/10.1016/j.ymssp.2014.12.020
14. Schwendemann, S., Amjad, Z., Sikora, A.: A survey of machine-learning techniques for condition monitoring and predictive maintenance of bearings in grinding machines. Comput. Ind. **125**, 103380 (2021). https://doi.org/10.1016/j.compind.2020.103380
15. Susto, G.A., Schirru, A., Pampuri, S., McLoone, S., Beghi, A.: Machine learning for predictive maintenance: a multiple classifier approach. IEEE Trans. Ind. Inf. **11**(3), 812–820 (2014). https://doi.org/10.1109/TII.2014.2349359

The Impact of Content Deletion on Tabular Data Similarity Using Contextual Word Embeddings

José Pilaluisa[1] and David Tomás[2](✉)

[1] Faculty of Engineering, Physical Sciences and Mathematics,
Central University of Ecuador, Avenida Universitatia, Quito 170129, Ecuador
jpilaluisa@uce.edu.ec

[2] Department of Software and Computing Systems, University of Alicante,
Carretera San Vicente del Raspeig s/n, 03690 San Vicente del Raspeig, Spain
dtomas@dlsi.ua.es

Abstract. Table retrieval is the task of answering a search query with a ranked list of tables that are considered as relevant to that query. Computing table similarity is a critical part of this process. Current Transformer-based language models have been successfully used to obtain word embedding representations of the tables to calculate their semantic similarity. Unfortunately, obtaining word embedding representations of large tables with thousands or millions of rows can be a computationally expensive process. The present work states the hypothesis that much of the content of a table can be deleted (i.e. rows can be dropped) without significantly affecting its word embedding representation, thus maintaining system performance at a much lower computational cost. To test this hypothesis a study was carried out using two different datasets and three state-of-the-art language models. The results obtained reveal that, in large tables, keeping just 10% of the content produces a word embedding representation that is 90% similar to the original one.

1 Introduction

The rapid evolution of digital technologies has propitiated the arise of Industry 4.0, also known as the "The Fourth Industrial Revolution", whose main goal is the interconnection of all parts of companies to produce effective automation and smarter enterprises. Artificial intelligence (AI) is identified as a key element of this transformation, together with the growing accumulation of large amounts of data (*Big Data*), the use of algorithms to process it, and the massive interconnection of digital systems and devices.

A large amount of the information produced by industry takes the form of structured data, i.e., tabular data. Although tables are widely used, it is often difficult to automatically retrieve relevant information stored in this format, which is a basic step to extract the knowledge required to boost this fourth industrial revolution. Table retrieval is a task that has been largely studied in

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 250–259, 2023.
https://doi.org/10.1007/978-3-031-18050-7_24

the literature [17] and whose goal is to answer a search query with a ranked list of tables that are considered as relevant to that query [16].

In the last years, in order to overcome the limitations of basic content matching approaches to table retrieval, the use of language models have become a popular way to understand the semantics of data and provide better results for a given search. Language models, which represent probability distributions over words or sequences of words, have been largely employed in the field of natural language processing (NLP) over the years to perform tasks such as information retrieval, text classification and language generation. The goal of these models in table retrieval is to obtain vector representations of the tables (*word embeddings*) in a latent semantic space, allowing to compare them by calculating their distance using measures such as cosine similarity.

Recently, Transformer language models [14] have emerged as the current state-of-the-art in most NLP tasks. These models are becoming also popular in table retrieval due to their capability to obtain accurate word embedding representations of their content. Unfortunately, the use of Transformer models on large datasets can be computationally expensive, compromising the performance of table retrieval systems in production scenarios.

The present work states the hypothesis that the content of a table can be reduced (by dropping rows) without significantly altering the word embedding representation of that table. This implies that the same high quality semantic representation can be obtained by using only a portion of the original table, reducing in this way the computational cost of using Transformer models.

In this paper, a study has been carried out to test this initial hypothesis. Two different datasets of varied nature have been analysed. The first one is a massive dataset consisting of nearly 500K tables from Wikipedia, which typically contain a reduced number of rows and columns. The second one includes 100 tables extracted from an Open Data portal with hundreds of thousands rows. The results obtained reveal that, in the case of large tables, it is possible to randomly drop up to 90% of the rows from the dataset without significantly changing the word embedding representation of the table, largely reducing the computational cost of applying Transformer models to the task of table retrieval.

The remainder of this paper is structured as follows: Sect. 2 presents related work in the use of language models for table retrieval; Sect. 3 describes in detail the study proposed; Sect. 4 presents the experiments carried out and discusses the results; finally, Sect. 5 presents conclusions and future work.

2 Related Work

This section summarises the existing work in the use of word embeddings for table retrieval. Word embeddings are dense vectors that represent the meaning of a word as a point in a semantic space. They represent the distributional meaning of words, so that similar representations are learnt from words appearing in similar contexts. When comparing two terms, word embeddings overcome the problems of lexical approaches based on string similarity. Terms such as "city" and "location" could be considered as being very different in terms of string matching, but in a word embedding space these two terms may be closely related and considered as highly similar. Examples of this type of word representation are Word2vec [9] and fastText [2].

The aforementioned word embedding approaches build a global vocabulary using unique words in the documents, assigning a single representation for each word and ignoring that they can have different meanings or senses in different contexts. They are considered as static representations unable to capture the different senses of a word. On the other hand, recent contextual word embeddings are able to capture the different meanings of polysemous words, since each vector represents not a word but a sense. In this way, each word is represented with different word embeddings, one for each context in which the word can occur. Examples of these type of representation are ULMFit [8], BERT [5] and DeBERTa [7].

Recent approaches to table retrieval rely on word embeddings to represent tabular data, applying vector similarity measures to calculate the relevance between tables [11]. Depending on the type of search query, table retrieval may be classified as keyword-based or table-based search [16]. In the former, a set of keywords form the query, as is the case with traditional search engines such as Google. In the latter, the query is also a table, and the goal is to compute a similarity score between the input and candidate tables.

In this vein, Shraga et al. [13] used Word2vec as the source for semantic vectors. The information of the table was separated in four semantic spaces: description (title and caption), schema (column headings), records (table rows), and facets (table columns). Then, different neural network architectures were applied to each semantic space, including recurrent convolutional neural network (description), multilayer perceptron (schema), and 3D convolutional neural network (records and facets). Finally, these four semantic spaces were combined using a gated multimodal unit.

To retrieve tables compatible with an input table, Nargesian et al. [10] tried to estimate if the table contents belonged to the same domain. They applied three statistical models: intersecting values between two columns, semantic similarity between values mapping the columns to classes in an ontology, and using word embeddings to measure similarity between textual values.

The work in [4] used contextual word embeddings for table-based search. The authors used a pre-trained version of BERT, leveraging different information available in the table (both textual and numerical) to provide BERT with context: title, caption, column headings, and cell values.

Finally, in the related task of table integration, Cappuzzo et al. [3] proposed algorithms for obtaining local embeddings for data integration tasks. They described a graph-based representation that allowed specifying a rich set of relationships in the relational data. The results revealed promising results for data integration tasks both in supervised and unsupervised settings.

3 Research Method

As mentioned before, the goal of the study carried out in this work is to test the hypothesis that the content of tables can be reduced without altering the word embedding representations generated by language models. This reduction can significantly cut the computation time and resources required to apply these models on large tabular datasets.

A basic approach was used to reduce tables by randomly selecting a percentage of the original rows, obtaining the word embedding representation of the subtable, and comparing the resulting semantic vector with the word embedding of the original table. A high similarity implies that both the original and the reduced table are very close in the semantic space generated by the language model. Only the content of the cells were taken into account in order to create the word embedding representing a table. Neither the column names nor additional metadata were considered.

More specifically, the procedure to compare the original table with a subtable consists of: (i) obtaining a word embedding for each column of these two tables; (ii) each column in the original table is compared with the corresponding column of the subtable by calculating the cosine similarity between their word embedding representations; (iii) this comparison provides a value in the range $[0, 1]$, where 1 implies maximum similarity and 0 no similarity at all; (iv) the final similarity between both tables is calculated as the average similarity between their columns.

The following aspects were analysed in this study:

- The performance of contextual (e.g. Transformers) and non contextual (e.g. Word2vec) language models
- The effect of table reduction depending on the size (number of rows) of the tables
- The coverage of language models, i.e., the percentage of table content that has a word embedding representation in the semantic space defined by the models (known as the *vocabulary*)
- The impact of numerical and textual datatypes in the word embedding representation

Regarding this last aspect, language models have largely proven their capabilities dealing with string information, but the study on how well they represent numerical information is more limited. Previous studies in the field suggest that these models also capture the semantics of numbers and their relationships [15,18]. Different experiments were carried out in this work considering all the content of the table, only string columns, and only numerical columns to contrast their results.

4 Experiments

This section describes the experiments carried out to validate the hypothesis stated: the models employed, datasets gathered and results obtained.

4.1 Models

The study carried out includes both contextual and non contextual language models. The aim is to compare the ability of these two types of models to generate word embeddings of tables that are robust to content reduction. Three different language models were analysed:

- Word2vec [9]: non contextual embedding vectors pre-trained on part of Google News dataset, comprising about 100 billion words. The model contains 300-dimensional vectors for 3 million words and phrases, although in the experiments presented here only words were considered.
- fastText [2]: non contextual embedding vectors pre-trained on Wikipedia 2017, UMBC webbase corpus and statmt.org news dataset, comprising about 16 billion words. As in the previous case, vectors have 300 dimensions. The pre-trained model containing subword information was used in order to increase the coverage of the vocabulary of the model.
- SentenceBERT [12]: contextual embedding vectors using siamese and triplet network structures to derive semantically meaningful sentence embeddings. It was trained on 1 million sentence pairs. The model used[1] produces embedding vectors of 384 dimensions.

A remarkable difference between these two types of models is the use by contextual models of subword units instead of full words to represent the vocabulary of the problem. Word embeddings are built on the specific set of tokens available in the corpus used to create the vectors. When an out-of-vocabulary word occurs in a new text, word-based models provide no representation for it in the semantic space and thus the token is considered as unknown. In order to handle the large vocabularies common in natural language corpora, SentenceBERT uses the WordPiece subword segmentation algorithm. In it, the vocabulary is initialised with individual characters in the language and then the most frequent combinations of symbols in the vocabulary are iteratively added to the vocabulary. Thus, subwords have their own representation in semantic space, and previously unknown words can be assigned a representation by combining the vectors of their underlying subword units.

4.2 Datasets

Two datasets have been used in this study to analyse different scenarios. The first one consist of 492,090 tables obtained from Wikipedia. This represents a subset of the corpus gathered by [1], consisting of 1.6 million tables from the November 2013 XML dump of English Wikipedia. In the present work, tables with less than 10 rows were discarded. It is referred to as the *Wikitables* dataset in the experiments.

The second dataset contains a subset of 100 tables from the Chicago Data Portal,[2] the open data initiative from this city that comprises datasets for topics

[1] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2.

[2] https://data.cityofchicago.org/.

such as building, education, environment, events, public safety, administration and finances. It is called the *Chicago* dataset in the experiments. Tables with less than 1,000 rows and more than 1,000,000 were filtered out. In this way a dataset quantitatively different from *Wikitables* is provided. Table 1 summarises the main characteristics of both datasets.

Table 1. Statistics summary for *Wikitables* and *Chicago* datasets.

Feature	*Wikitables*	*Chicago*
Number of tables	492,090	100
Number of rows	14,106,528	10,303,767
Number of columns	2,812,631	1,818
Number of numerical columns	676,536	723
Avg. number of rows	28.67	103,037.67
Avg. number of columns	5.85	18.18
Avg. number of numerical columns	1.37	7.23
Max. number of rows	4670	975,010
Max. number of columns	237	119
Max. number of numerical columns	115	70

Wikitables includes a large number of medium size tables, while *Chicago* contains a small number of large size tables. The average number of rows in *Wikitables* is 28.67, whereas in *Chicago* this value is 103,037.67. The number of columns is also notably higher in *Chicago*, 18.18 compared to 5.85 in the other dataset.

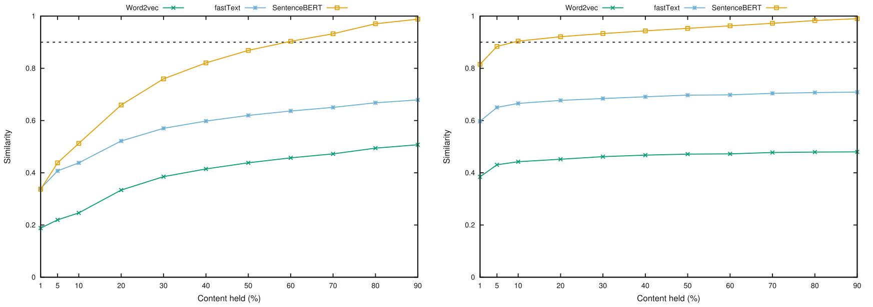
4.3 Results

In order to evaluate how the word embedding representation of a table changes when its content is reduced, a set of thresholds were established in terms of the percentage of rows randomly selected from the original table: 1%, 5% and multiples of 10 until 90%, where 1% means that 99% of the rows were dropped from the original table. For each table, 11 subtables were obtained (from 1% to 90%) and compared with the original one using the three language models mentioned above: Word2vec, fastText and SentenceBERT.

As described in the dataset section, tables with less than 10 rows were discarded in these experiments. In tables with less than 20 rows, 1% and 5% subtables were not calculated, as there are not enough rows to create these subsets. In tables with less than 100 rows, the 1% subtable was discarded for the same reason.

Regarding the models studied, in the case of Word2vec and fastText the word embedding representation of a column was obtained by averaging the word embedding vector of each word in that column, since these models only offer representations at word level. Averaging word embeddings is one of the most popular methods for combining embedding vectors, outperforming more complex techniques especially in out-of-domain scenarios [6]. Unlike these two models, SentenceBERT provides word embeddings also at sentence level with a limit of 256 input tokens. These means that, for tables with 256 rows or less (assuming a single token in each cell), the model automatically provides a single vector representing each column. In those situations were the number of tokens exceeded this limit, the tokens were processed in groups of 256 tokens, obtaining an embedding of each of these groups and finally averaging them to obtain a single vector for the column.

Figure 1 (a) shows the similarity achieved (y axis) by the three models tested in the *Wikitables* corpus when different percentages of rows were selected (x axis). The straight dashed line in the top corresponds to the 90% similarity threshold, which can be considered as a high similarity level. To test significance, a two-tailed paired t-test was used.



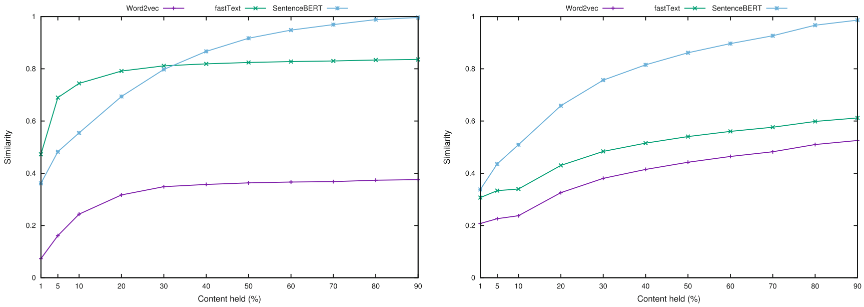
(a) (b)

Fig. 1. Similarity obtained by Word2vec, fastText and SentenceBERT with different percentages of rows kept in (a) *Wikitables* and (b) *Chicago* corpus. The dashed line in the top represents the 90% similarity threshold.

The results show that SentenceBERT significantly outperforms non contextual models ($\rho < 0.01$ in all subtables for *Chicago* corpus and in subtables over 5% in *Wikitables* corpus), being Word2vec embeddings the most negatively affected by the reduction of content. Using SentenceBERT, the 60% subtable surpassed the 90% similarity threshold with respect to the original table, which implies that 40% of the rows could be dropped and still obtain a word embedding that is 90% similar to the original one. With the same number of rows, Word2vec obtained only 45% similarity and fastText 64%.

Similar results were obtained on the *Chicago* dataset, as shown in Fig. 1 (b). In this case, with tables containing a significantly larger number of rows than in *Wikitables*, the word embedding representation is less sensitive to row deletion. In the case of SentenceBERT, keeping only 10% of the rows allows to obtain a word embedding that is 90% similar to the original one. In this same setting, Word2vec obtained 44% similarity and fastText 67%.

Figure 2 shows the similarity achieved using only (a) numerical columns and (b) string columns on the *Wikitables* corpus.[3] The results achieved in both cases by SentenceBERT are very similar. In the case of fastText, it is surprising the high similarity obtained using only numerical values, with a similarity close to 80% for the 20% subtable (this similarity was only 48% in the string only version). Finally, Word2vec obtained better results on average using string values.



(a)

(b)

Fig. 2. Similarity obtained using only (a) numerical columns or (b) string columns by Word2vec, fastText and SentenceBERT in *Wikitables* corpus.

It is important to note the coverage given by each model, i.e., the number of words in the table that were part of the vocabulary of the model and thus had a word embedding representation. As SentenceBERT uses subword information, every token has an embedding representation, providing 100% coverage for every numerical and string value in the datasets. In the case of Word2vec, the coverage was 53.24% for numerical columns and 56.13% for string columns. This limited coverage provides an explanation to the low performance achieved by the model with both datatypes. fastText covered 90.03% of numerical columns and 66.05% of string columns. These values correlate with the performance shown in Fig. 2, where using only numerical tokens obtained much better results ($\rho < 0.01$) for SentenceBERT and fastText than using only text tokens.

5 Conclusions and Future Work

This paper presented a study on how content deletion impacts the word embedding representation of tabular data. The goal was to demonstrate that a significant part of the content of a table can be deleted without affecting its representation in the semantic space provided by word embeddings.

[3] Similar results were obtained in *Chicago* dataset and are not included here due to space limitations.

The study showed that contextual models provide word embedding vectors that are less sensitive to content reduction. For instance, in the case of the *Chicago* dataset, keeping only 10% of the rows allowed to obtain word embeddings that were on average 90% similar to the original tables.

The impact of table reduction was less significant in tables with many rows. In the case of *Chicago* dataset (an average of 103,037.67 rows per table), keeping just 10% of rows provided an average similarity between original and reduced tables above 90%. In the case of *Wikitables* (29.67 rows per table), 60% of rows should be kept to achieve same similarity values.

Further experiments revealed that using only numerical or textual columns significantly affected the performance of Word2vec and fastText. SentenceBERT did not offer a significant difference between both datatypes.

Analysing the coverage of each model showed a clear correlation with performance. Word2vec offered the worst coverage and worst similarity values. In the case of fastText, there was a higher coverage for numbers with respect to text, which correlates with a significantly higher similarity using only numerical values.

These results provide a promising path to reduce computational costs in table retrieval using word embeddings. As a future work, rather than random selection, a more informed criterion for row deletion will be investigated. Additionally, an extrinsic evaluation on different datasets will be carried out to test how the table reduction affects the final performance of a table retrieval system using word embeddings.

Acknowledgements. This research has been partially funded by project "Desarrollo de un ecosistema de datos abiertos para transformar el sector turístico" (GVA-COVID19/2021/103) funded by "Conselleria de Innovación, Universidades, Ciencia y Sociedad Digital de la Generalitat Valenciana".

References

1. Bhagavatula, C.S., Noraset, T., Downey, D.: TabEL: entity linking in web tables. In: Arenas, M., et al. (eds.) The Semantic Web - ISWC 2015: 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I, pp. 425–441. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25007-6_25
2. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Trans. Assoc. Comput. Linguist. **5**, 135–146 (2017). https://doi.org/10.1162/tacl_a_00051
3. Cappuzzo, R., Papotti, P., Thirumuruganathan, S.: Embdi: generating embeddings for relational data integration. In CEUR (ed.) 29th Italian Symposium on Advanced Database Systems (SEDB), Pizzo Calabro, Italy (2021)
4. Chen, Z., Trabelsi, M., Heflin, J., Xu, Y., Davison, B.D.: Table search using a deep contextualized language model. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 589–598. Association for Computing Machinery, Online (2020)

5. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, pp. 4171–4186. Association for Computational Linguistics, Minneapolis (2019)
6. Gupta, S., Kanchinadam, T., Conathan, D., Fung, G.: Task-optimized word embeddings for text classification representations. Front. Appl. Math. Statis. **5**, 1–10 (2020)
7. He, P., Liu, X., Gao, J., Chen, W.: DeBERTa: decoding-enhanced BERT with disentangled attention. In: International Conference on Learning Representations, pp. 1–21, Online (2021)
8. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, pp. 328–339. Association for Computational Linguistics, Melbourne (2018)
9. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS 2013, pp. 3111–3119. Curran Associates Inc., Red Hook (2013)
10. Nargesian, F., Zhu, E., Pu, K.Q., Miller, R.J.: Table union search on open data. Proc. VLDB Endow. **11**(7), 813–825 (2018)
11. Thanh Tam Nguyen, Quoc Viet Hung Nguyen, Weidlich Matthias, and Aberer Karl. Result selection and summarization for web table search. In Proceedings of the 31st International Conference on Data Engineering (ISDE 2015), pp. 231–242. IEEE, Seoul (2015)
12. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 3982–3992. Association for Computational Linguistics, Hong Kong (2019)
13. Shraga, R., Roitman, H., Feigenblat, G., Cannim, M.: Web table retrieval using multimodal deep learning. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1399–1408. Association for Computing Machinery, Online (2020)
14. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30, pp. 5998–6008. Curran Associates, Inc., Long Beach (2017)
15. Wallace, E., Wang, Y., Li, S., Singh, S., Gardner, M.: Do NLP models know numbers? Probing numeracy in embeddings. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 5307–5315. Association for Computational Linguistics, Hong Kong (2019)
16. Zhang, S., Balog, K.: Ad hoc table retrieval using semantic similarity. In: Proceedings of the 2018 World Wide Web Conference, pp. 1553–1562. International World Wide Web Conferences Steering Committee, Lyon (2018)
17. Zhang, S., Balog, K.: Web table extraction, retrieval, and augmentation: a survey. ACM Trans. Intell. Syst. Technol. **11**(2), 1–35 (2020)
18. Zhang, X., Ramachandran, D., Tenney, I., Elazar, Y., Roth, D.: Do language embeddings capture scales? In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 4889–4896. Association for Computational Linguistics, Online (2020)

Deep Learning-Based Dementia Prediction Using Multimodal Data

David Ortiz-Perez[1](✉), Pablo Ruiz-Ponce[1], David Tomás[2], and Jose Garcia-Rodriguez[1]

[1] Department of Computer Science and Technology, University of Alicante, Carretera San Vicente del Raspeig s/n, 03690 San Vicente del Raspeig, Spain
{dortiz,pruiz,jgarcia}@dtic.ua.es

[2] Department of Software and Computing Systems, University of Alicante, Carretera San Vicente del Raspeig s/n, 03690 San Vicente del Raspeig, Spain
dtomas@dlsi.ua.es

Abstract. In this paper we propose a deep architecture to predict dementia, a disease which affects around 55 million people all over the world and makes them in some cases dependent people. To this end, we have used the DementiaBank dataset, which includes audio recordings as well as their transcriptions of healthy people and people with dementia. Different models have been used and tested, including Convolutional Neural Networks for the audio classification, Transformers for the text classification and a combination of both models in a multimodal one. These models have been tested over a test set, obtaining the best results from the text modality, achieving a 90.36% of accuracy on the detection of dementia task.

1 Introduction

Nowadays, around 55 million people in the world have dementia, which is more commonly seen in older people but can also affect younger ones. Dementia is a syndrome which affects normal cognitive function. The most common form of dementia is Alzheimer's disease, which represents 60–70% of the cases [1]. This syndrome can affect a different way to each patient and has three different stages: early stage, middle stage and late stage. Each stage can have different symptoms that can vary from losing track of time in the early stage, forgetting recent events or becoming confused at home in the middle stage and finally having difficulties recognizing relatives or friends in a late stage among others. The late stage of dementia will limit their autonomous life, so they will need a relative or a professional to take care of them.

Since dementia is usually related to older people, the number of patients who have dementia is expected to grow in the following years. This is a problem which will be more and more present in our society. Consequently, early disease detection is very important.

The contribution of this work is the creation of an architecture composed of different deep learning modules that process text transcriptions and audio

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 260–269, 2023.
https://doi.org/10.1007/978-3-031-18050-7_25

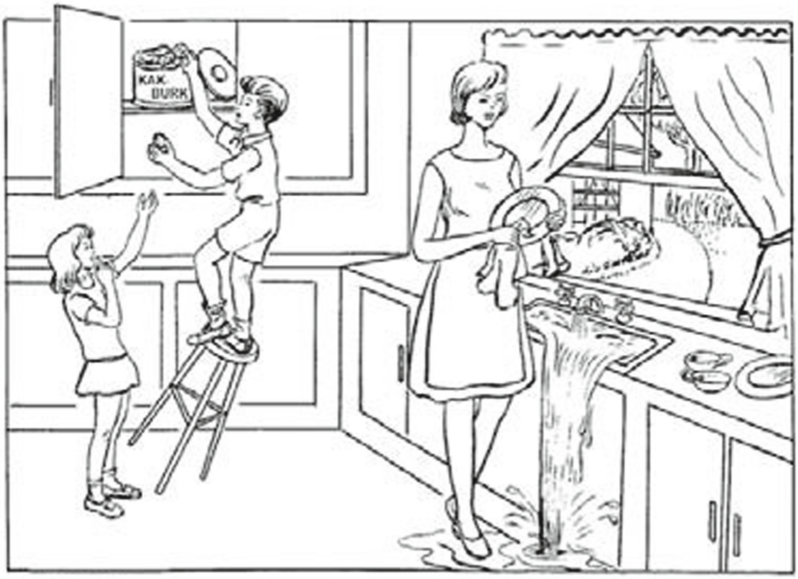
recordings. The main goal of this work is to develop a system that can predict symptoms of dementia, especially in the early stages of the disease. Even though there is no cure for dementia, there are treatments, with or without medicines, such as therapies that can help with the symptoms that they are dealing with. For this reason, the detection of this syndrome is important, since this detection and treatment can improve the quality of life of patients, and their relatives and friends. As a first step, on one hand, we researched the possible datasets where we could obtain data for this task, data about patients who suffer or may suffer from dementia in the future. On the other hand, we looked for the most interesting deep learning techniques for classification tasks and tested them. The most interesting dataset we found was DementiaBank [3], which contains data in different modalities like text and audio. We developed and tested implementations using both modalities of data, separately and combined. A detailed study of the different combinations suggested that the use of text information provides the highest accuracy.

The remaining of the paper is organized as follows: Sect. 2 presents the dataset used to test our proposal. In Sect. 3 we present the different approaches and validate them with experiments on the DementiaBank dataset showing our results in Sect. 4. Finally, in Sect. 5 we summarize our conclusions and propose further work.

2 DementiaBank Dataset

As this is a highly sensitive topic, there are not too many options available for public use. Even though there are some other options with other kinds of information, such as medical data, we have focused on the DementiaBank - Pitt Corpus [3]. This dataset contains the audio of the recording as well as a transcription of the dialogue between the interviewer and the patients. The range of age of the patients goes from 46 to 90 years, including patients from both genders. The statistics about the number of healthy and dementia affected patients including the number of samples of each one can be seen in the Table 1. Subjects were asked to describe an image shown to them. Specifically, the image used was the Cookie-Theft picture shown in Fig. 1.

This image has been used in clinical and experimental research, specifically in the field of mental and cognitive impairments. This experiment was designed to detect some of the signs of dementia, such as having difficulties choosing the right words, choosing the wrong ones, using related or substitute words, or even not finding a word at all. Other signs shown include using words with no meaning or not related to the conversation [2]. Another reason to have chosen this dataset for our project is the availability of the speech part of the patients describing an image. Moreover, this dataset includes multimodal data, which was one of the main goals of this investigation by evaluating if the simultaneous use of different modalities provides better results.



Copyright © 1983 by Lea & Febiger

Fig. 1. The Cookie Theft picture, from [12]

Table 1. Pitt corpus statistics

	Dementia	Control
Number of patients	194	99
Number of samples	309	243

Among the recent projects working over this dataset, we can mention the work of Tifani Warnita et al. [21] which was released in 2018. In this work, they used only the audio data of the Pitt corpus, and the model used was a gated convolutional network, with this model a 73.6% of accuracy was achieved. Another work that uses just audio data is the one presented by Rupayan Chakraborty et al. [4] in 2020. In their project, they proposed a model that analyses the audio clips in order to obtain audio biomarkers for the detection of dementia. There are also works working just over the text modality, as can be seen in the work of Sweta Karlekar et al. [11] released in 2018. In their work, the best results obtained were by the use of CNNs combined with RNNs and the POS-tagging transcriptions of the utterances. The best results were obtained in this work, achieving an accuracy of 91.1%, the data used was down-sampled because not every utterance had a POS-tagging transcription accompanying. There are also different approaches where both modalities are combined as in the case of the work of Amish Mittal et al. [17] in 2021. For this work, they used both modalities, using two different models and weighing their probability of dementia. For the audio model, a Mel Spectrogram combined with an audio based was used, and for the text model, different combinations for segment transcriptions and the full transcription were used. By using this model, they obtained an accuracy of 85.3%. This dataset have a smaller subset which has been balanced in terms of age and gender, called ADReSS challenge [13], and have also different approaches with different methods [6,15,16].

3 Approach

We have decided to split this dataset into two different sets: the training set with a size of 469 samples (representing 85% of the dataset) and a test set comprising 83 samples (representing the remaining 15%). As commented before, this dataset contains data in two different modalities: the audio of the recordings and their transcriptions. We developed different models for each modality in order to evaluate how well they work and finally combine them into a multimodal model trying to improve the results of the single modalities models.

3.1 Audio

The first approach analyzed the audio files, in the Fig. 2 the architecture of the implemented model is presented.

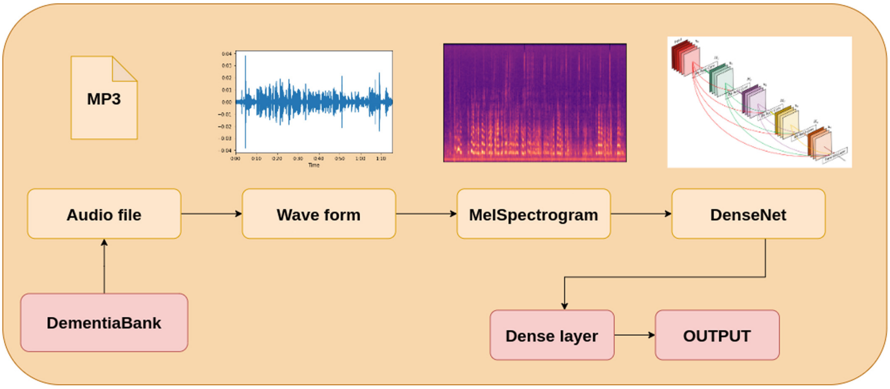


Fig. 2. Architecture of the audio model

Each audio file has been converted to its waveform, a graphical representation of the signals over time, and then converted to a Mel Spectrogram, a Spectrogram in a Mel Scale [19]. This Mel Scale is inspired by the way humans perceive sounds, differentiating the low frequency sounds more easily than the high frequency ones. In this scale, two equally distanced sounds in the pitch sound equally distanced to a listener. After this conversion, we used a Convolutional Neural

Network to process this spectrogram. This model will handle the spectrogram as an image. This is because these networks are used basically for image tasks, such as image classification. They apply different convolutions/kernels in order to extract features from the image. In this model, different pre-trained CNNs have been tested, such as MobileNet [9], DenseNet [10] and ResNet [7]. The best results were obtained with the DenseNet model, in the Sect. 4 the results of the audio model will be referencing the ones obtained by using this DenseNet. The final step of this model is a dense layer with the outputs of this CNN in order to get a final output, the prediction of the model (dementia or control). This implementation, based on Mel Spectrogram of an audio that combine it with a CNN achieved the state-of-the-art in some audio classification tasks [8,18]. For this reason, it has been used for our implementation.

3.2 Text

On the other hand, we tested the text modality, which was in a CHAT format [14], a format used by TalkBank in their corpora, like in this case the Pitt Corpus of DementiaBank. This CHAT format implies that is available not only for the transcription of the subjects, moreover the interviewer transcription was annotated as well as personal information from the patients and special flags representing pauses or mistaken words among others. For this reason, the original transcription files have been pre-processed in order to obtain a clean text transcription. Figure 3 shows the architecture of the proposed model for the text analysis.

As it can be seen in the architecture, the famous BERT [5] model has been used, a model which has achieved the state-of-the-art in many natural language processing tasks. This BERT model is based on Transformers [20], stacking different transformer encoders that will extract features from the text. The most interesting aspect of these transformers is the use of attention in order to establish relations between the different words in the sentence. In order to use the BERT model for this task, we fined-tune a pre-trained BERT model, the way used for fine-tuning is explained in the following paragraphs.

After a text tokenization process, the BERT model receives as input those encoded words and returns its embeddings, a different embedding for each word received as input, having a size of 768 each embedding. These embeddings are a way of representing a word in the natural language processing models, allowing it to establish similar representations to similar meaning words.

One way of fine-tuning this BERT model is to use the embedding of the "[CLS]" token (the first one) to a final dense layer for classification. This "[CLS]" embedding is the representation of the whole text, this is the reason why this embedding is used for fine-tuning. This was not the only method tested for the text representation in our experiment with the test set. The other model tested uses these embeddings to fit a bidirectional Long Short-Term Memory (LSTM). This type of network has proved to have good results for a task with sequences such as text, and before the introduction of transformers, they were the previous state-of-the-art in many NLP tasks. LSTMs work using the output of one input

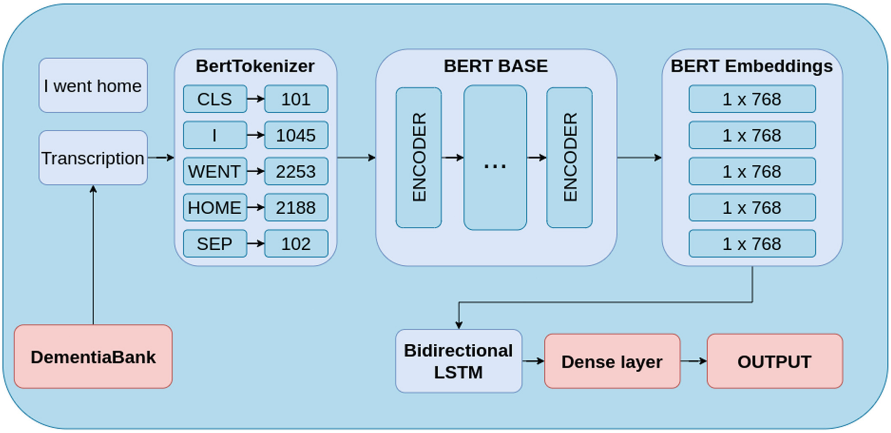


Fig. 3. Architecture of the model

(words or embeddings in this case) as the input of the following one, keeping information from the previous data. These models also have a mechanism to forget the irrelevant data from the previous segments and keep the important ones. Finally, the output of the LSTM is used as an input for a final dense layer to obtain the final output of the model. The comparison of the results of these two similar models will be seen in the Sect. 4

3.3 Multimodal

Finally, both implemented methods, text and audio, have been combined into a multimodal model to test if we can obtain better results. For this task, the two previously used models have been used concatenating their outputs before a final dense layer to get a final prediction.

3.4 Other Approaches

As we mentioned before, the best results on this dataset were achieved by combining text features with the POS-tagging of the text [11]. This is the reason why we have decided to try using the POS-tagging of the text as well to see what results we obtain. After obtaining the POS-tagging of the text by using the Python library spaCy[1], the features have been introduced in an embedding and a bidirectional LSTM. After this LSTM, a dense layer for a final classification has been used.

Since the CHAT format of the transcriptions has a lot of information and not only the plain text, there are special flags that represent, for example, a pause in the patient response or a mistaken word. As some symptoms of dementia are having difficulties finding certain words, this can lead to pauses to think, or

[1] https://spacy.io/.

using mistaken words, these special flags may give us some relevant information for this task. These special tokens have been counted and compared between control and dementia patients, the Fig. 4 shows how these flags are present in the different transcriptions of healthy people and people with dementia.

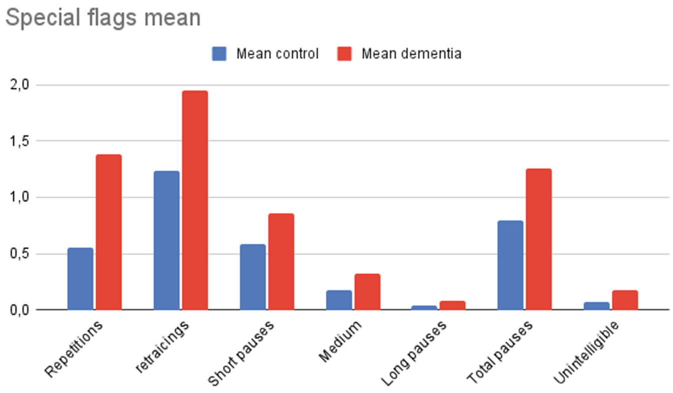


Fig. 4. Special flags mean

Among other flags, the ones that have shown differences between control and dementia patients are: *repetitions*, *retracing*, *pauses* and *unintelligible*. Other flags such as doubts have not shown a big difference between both.

4 Evaluation

To measure the results of the tested models, we used the accuracy in the test set. This accuracy is the percentage of a correct prediction among all the predictions. The obtained results by each model can be seen in the Table 2 excluding the models mentioned in Sect. 3.4. These models have been discarded since they did not achieve good results in our experiments. Both models did not even learn from the training set, resulting in random predictions by the model, with an accuracy of around 55% in both sets. Even though there is a notable difference as can be seen in the Fig. 4 in the number of mean *repetitions*, *retracing*, *pauses* and *unintelligible* flags between dementia and control people, this did not result into a good prediction model. Neither did the POS-tagging model, which obtained similar results. This is the reason to discard and not combine them with the other approaches, which did get good results.

Table 2. Comparison of the different tested models

Model	Description	Accuracy
Audio	Mel Spectrogram + CNN (DenseNet)	73.49%
Text 1	BERT embeddings + dense layer	84.33%
Text 2	BERT embeddings + bidirectional LSTM + dense layer	**90.36%**
Multimodal 1	Audio + Text 1	84.33%
Multimodal 2	Audio + Text 2	86.65%

The best results obtained by the different implemented models were given by the text model which used the BERT embeddings as well as a Bidirectional LSTM. Another important remark that we can obtain from these experiments is that we have obtained significantly better results from the text part of the dataset than from the audio one. In our single case with our experimentation and proposed models, the use of a multimodal model which will work over audio and text has not improved our previously obtained best results. The use of a bidirectional LSTM improves the results obtained from using just the embedding of the "[CLS]" token for this task.

Some other metrics that can be relevant in talking about the obtained results are the precision, recall and f1-score, which measures the quality of the model while predicting and facing dementia cases. In the case of the best accuracy obtained results, with the text model 2, all the three metrics got the same results, obtaining a 91.11% at each metric, slightly improving the obtained accuracy.

5 Conclusion

In this work, different models and approaches have been tested on the task of dementia detection. The detection of this dementia, especially in the early stages can help the patients in order to improve their quality of life through different treatments. In order to obtain the architecture of the different proposed models, research on the state-of-the-art approaches on similar classification tasks has been done. Among the different approaches we have implemented and tested we can find the following ones: the use of CNNs for audio classification and the use of transformers, with the BERT model, for text classification and finally a combination of both. The model that best works in this task has finally resulted in the use of transformers with the BERT model.

For future works related to this field, we plan to use this trained model with a different type of dataset related to other mental diseases, since there are different diseases which have similarities with dementia, such as Aphasia disease. Another future work that could be interesting is the addition of a video modality as well as the analysis of the different emotions shown in the different modalities. The aim of this would be to try to identify relations between the different expressed emotions to detect symptoms of dementia.

Acknowledgment. We would like to thank "A way of making Europe" European Regional Development Fund (ERDF) and MCIN/AEI/10.13039/501100011033 for supporting this work under the MoDeaAS project (grant PID2019-104818RB-I00). Furthermore, we would like to thank Nvidia for their generous hardware donation that made these experiments possible.

References

1. Dementia. https://www.who.int/news-room/fact-sheets/detail/dementia
2. Dementia and language, March 2022. https://www.alzheimers.org.uk/about-dementia/symptoms-and-diagnosis/symptoms/dementia-and-language
3. Becker, J.T., Boiler, F., Lopez, O.L., Saxton, J., McGonigle, K.L.: The natural history of Alzheimer's disease: description of study cohort and accuracy of diagnosis. Arch. Neurol. **51**(6), 585–594 (1994)
4. Chakraborty, R., Pandharipande, M., Bhat, C., Kopparapu, S.K.: Identification of dementia using audio biomarkers (2020). https://doi.org/10.48550/ARXIV.2002.12788. https://arxiv.org/abs/2002.12788
5. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR abs/1810.04805 (2018). http://arxiv.org/abs/1810.04805
6. Haulcy, R., Glass, J.: Classifying Alzheimer's disease using audio and text-based representations of speech. Front. Psychol. **11** (2021). https://doi.org/10.3389/fpsyg.2020.624137. https://www.frontiersin.org/article/10.3389/fpsyg.2020.624137
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR abs/1512.03385 (2015). http://arxiv.org/abs/1512.03385
8. Hershey, S., et al.: CNN architectures for large-scale audio classification. CoRR abs/1609.09430 (2016). http://arxiv.org/abs/1609.09430
9. Howard, A.G., et al.: MobileNets: efficient convolutional neural networks for mobile vision applications. CoRR abs/1704.04861 (2017). http://arxiv.org/abs/1704.04861
10. Huang, G., Liu, Z., Weinberger, K.Q.: Densely connected convolutional networks. CoRR abs/1608.06993 (2016). http://arxiv.org/abs/1608.06993
11. Karlekar, S., Niu, T., Bansal, M.: Detecting linguistic characteristics of Alzheimer's dementia by interpreting neural models. CoRR abs/1804.06440 (2018). http://arxiv.org/abs/1804.06440
12. Kokkinakis, D., Fors, K.L., Björkner, E., Nordlund, A.: Data collection from persons with mild forms of cognitive impairment and healthy controls-infrastructure for classification and prediction of dementia, May 2017
13. Luz, S., Haider, F., de la Fuente, S., Fromm, D., MacWhinney, B.: Alzheimer's dementia recognition through spontaneous speech: the ADReSS challenge. In: Proceedings of INTERSPEECH 2020, Shanghai, China (2020). https://arxiv.org/abs/2004.06833
14. Macwhinney, B.: The CHILDES project: tools for analyzing talk. Child Lang. Teach. Ther. **8** (2000). https://doi.org/10.1177/026565909200800211
15. Mahajan, P., Baths, V.: Acoustic and language based deep learning approaches for Alzheimer's dementia detection from spontaneous speech. Front. Aging Neurosci. **13** (2021). https://doi.org/10.3389/fnagi.2021.623607

16. Martinc, M., Pollak, S.: Tackling the ADReSS challenge: a multimodal approach to the automated recognition of Alzheimer's dementia, November 2020. https://doi.org/10.21437/Interspeech.2020-2202
17. Mittal, A., Sahoo, S., Datar, A., Kadiwala, J., Shalu, H., Mathew, J.: Multi-modal detection of Alzheimer's disease from speech and text. CoRR abs/2012.00096 (2020). https://arxiv.org/abs/2012.00096
18. Palanisamy, K., Singhania, D., Yao, A.: Rethinking CNN models for audio classification. CoRR abs/2007.11154 (2020). https://arxiv.org/abs/2007.11154
19. Roberts, L.: Understanding the MEL spectrogram, March 2020. https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53
20. Vaswani, A., et al.: Attention is all you need (2017)
21. Warnita, T., Inoue, N., Shinoda, K.: Detecting Alzheimer's disease using gated convolutional neural network from audio data, pp. 1706–1710, September 2018. https://doi.org/10.21437/Interspeech.2018-1713

Lightweight Models in Face Attribute Recognition: Performance Under Oclussions

Jaime Aznar-Espinosa[1](✉), Ángela Sánchez-Pérez[2], Jose Garcia-Rodriguez[1], and Javier Barrachina[2]

[1] Institute of Informatics Research, University of Alicante, Alicante, Spain
jae16@alu.ua.es, jgr@ua.es
[2] FacePhi Research, Alicante, Spain
{asanchezperez,jbarrachina}@facephi.com

Abstract. In this paper we will study the performance of several lightweight convolutional neural networks with respect to state-of-the-art models for facial attribute classification. Specifically, we will try to determine the attributes of gender, age and ethnicity. There are many models based on lightweight architectures, from which we have chosen MobileNet and EfficientNet. The results obtained match or improve the state of the art in gender and race, achieving good results in age classification as well. On the other hand, we have performed a comparative study of these classifications with respect to two datasets. The first dataset is UTK-Face which contains the facial images aligned and a higher number of individuals, having a lower total number of samples, while the second dataset is VGG-Face2 which has a much higher total number of samples, having fewer individuals than UTK-Face and with a lower quality facial alignment.

Keywords: Gender recognition · Age classification · Race classification · Efficient-net · MobileNet · UTKFace · MAAD-Face · VGGFace2

1 Introduction

Nowadays there is a wide variety of artificial intelligence systems dedicated to the extraction of facial features such as age, gender and ethnicity, both in images and videos (usually from video-surveilled locations). Therefore, many papers are published every year presenting new techniques and innovative models which are usually based on a complex system of convolutional neural networks. Due to this, the usage of this networks in environments with low computational capacity, such as video surveillance systems or mobile devices, is inefficient. For this reason, lightweight systems that are easily adaptable to new platforms and that are capable of achieving results close to the state of the art are increasingly sought after and needed.

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 270–279, 2023.
https://doi.org/10.1007/978-3-031-18050-7_26

The purpose of this paper is to compare the best models for gender, age and ethnicity with our implementations of EfficientNet [10] and MobileNet [4] which have been fine tuned on UTK-Face [14] and VGG-Face2 [1] datasets. On the other hand, we will choose the best models within our study and perform a brief comparison with specific images from MLFW (Masked LFW) [13], a dataset that presents images of aligned individuals with and without masks, to determine the effect that these masks have on attributes recognition.

The remaining of the paper is organized as follows: in Sect. 2 we revised related works, while Sect. 3 described our proposal which is validated in Sects. 4 and evaluated with masked faces dataset in Sect. 5. Finally in Sect. 6 we summarized our contributions and proposed further works.

2 Related Work

In this section we will describe the most relevant recent work on gender and age issues, with a brief note on ethnicity classifiers.

At current times, studies usually perform race classification as a support on both gender and age classification, as the studies focus on describing the variation in accuracy in both age and gender depending on which race the sample is categorized into. Therefore, we do not have data for models intended solely for race classification applied to faces.

There are two main approaches in age classifier. The first one is based on a regression approach, i.e., determining the exact age of the sample being analyzed. The second, and more common approach performs a classification within an age range, this is due to the high probability of failure in a discrete prediction. In most studies, regressive approaches are used, since having a higher error makes it more challenging to innovate.

Within these regressive models we would like to highlight the DADL (Deep Age Distribution Learning) model [5] which is based on the VGGFace architecture and uses the IMDB-Wiki and ChaLearn datasets, achieving an improvement in the state of the art by having a MAE of 1.75. On the other hand, it is interesting to mention a system approached from a different point of view, since it is composed of 3 integrated parts, which try to perform the full task of sample matching and classification: first of all it uses the Viola-Jones algorithm [12] to align the face, then it extracts the image features with the Pyramid Multi-Level (PML) and finally applies the SVR (Suppor Vector Regression) [3], obtaining a MAE of 4.23 working with the MORPHII [7] and PAL [6] datasets.

Models dedicated to gender estimation are usually very simple, as binary classification problems are usually easier to solve, so there is no need for innovation in models, but the focus is more on the inspection of a suitable dataset. One of the most used datasets is Celeb with accuracies of 93.3% using ALEXNet or 95% with VGGNet. Another of these datasets is UTK-Face, which we will use later, with an accuracy of 91.9% usingMobileNet (the version with the highest number of parameters and therefore less optimized) or 87% using the FaceNet network. As can be seen, the models used are widely studied models with well-known structures.

Therefore, our task is to propose a system that, using lighter models such as MobileNet in its base version or EfficientNet, improves or mimics the state of the art accuracies described above.

3 Description of the System

This section described the main features of the models and datasets used in our proposal.

3.1 Models

The first model chosen was EfficientNet in its pre-trained 'b0' base version, which has the smallest number of parameters and therefore is the lightest. In total, this network contains 210 parameters, of which we have decided to retrain the last 120 parameters after performing the relevant tests required, since the first 90 parameters are very useful when performing the feature extraction task.

The other model we have chosen has been MobileNet in the same way in its base version. This network contains fewer parameters than EfficientNet since in this version it only has 150 parameters, of which we have decided to retrain 80, for the same reason as in the previous case, both models were trained for ImageNet [2] challenges, so they are able to generalize adequately having a good feature extraction.

The same training parameters were used for both models: we employed the crossentropy loss function by adjusting the weights of each class due to the disparity present in the sample (each weight is restructured by dividing the majority class by the target class), the optimizer chosen is ADAM, a widely used optimizer, with a base learning rate of 0.001.

3.2 Datasets

After a study of the most relevant datasets for attribute recognition, we have selected the two datasets we are going to work, which will be described below.

The first of them is UTK-Face, this dataset has about 24000 images of different individuals and was designed to perform the recognition of the three attributes to be studied. As we can see in Fig. 1, it presents an equal distribution of data in the case of gender, however in the cases of age and ethnicity it presents a distribution that tries to simulate the real population distribution.

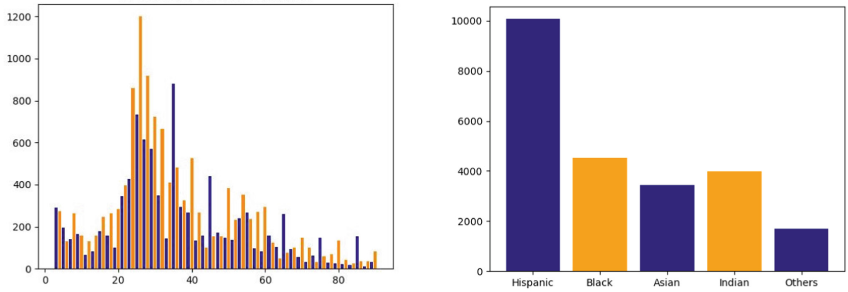


Fig. 1. UTK-face distributions. On the left the age distribution. On the right the race distribution

The second dataset is VGG-Face2, this dataset was designed to perform face recognition and does not provide information on the attributes of the sample. To solve this issue we used MAAD-Face [11] which is an attribute dataset of VGG-Face2. This dataset has 47 attributes for each sample including an annotation of gender divided in masculine and feminine, an annotation related to age divided into young, middle-aged and senior, and an annotation of race including white, black and asian. VGG-Face2 dataset is considerably more extensive than UTK-Face, possessing 3.3 million samples of up to 9 thousand identities.

Once the study of the sample distribution was done, we realized that since it was not initially designed for attribute recognition, the decompensation between classes was very noticeable, as can be seen in Fig. 2, so we decided to use only some parts of the dataset so that the relationship between classes would be more proportional (as we will discuss in the next section) and we could achieve promising results in evaluation.

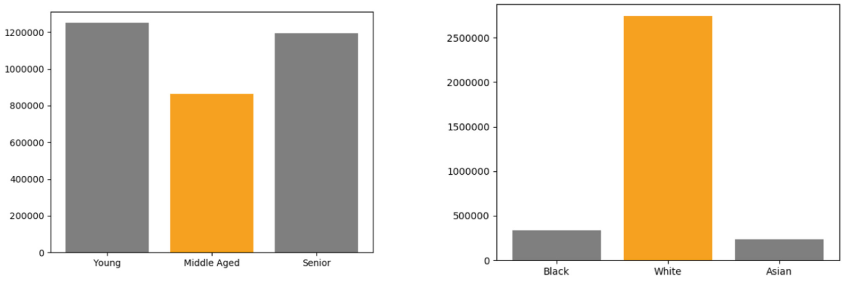


Fig. 2. VGG-Face2 distributions. On the left the age distribution. On the right the race distribution.

4 Experimental Setup

To validate our proposal, we trained and evaluated our models with state-of-the-arts datasets.

4.1 Training

In both networks the input image size is 224 × 224 with 3 color channels, however neither of the two datasets have this image size, since in VGG-Face2 the input sizes are variable and in UTK-Face the input has a fixed size of 200 × 200 × 3. Therefore, we apply transformations on the sample, performing a center crop that adds a black border on the images and we also apply a normalization of the data to help the network in the learning process.

When we had to choose the data distributions to be used in both training and evaluation, we had to make certain modifications to improve the performance of the models.

In the case of UTK-Face, the gender estimation did not require any adjustment, however in the age estimation we decided to reassign the weight value of the samples under 2 years old and over 90 years old, since in the first case they are relatively abundant samples that do not bring much value, and in the second case because they are a very marginal sample size that only bring confusion to the models. Finally, in the case of ethnicity, we decided to remove the 'Others' class and redistribute the samples into the according class, since it grouped 3 minor ethnicities with a very low sample size that could be included in one of the other classes.

In the case of VGG-Face2, due to the decompensation in age and race distribution (as can be seen in Fig. 2) and the repetition of samples (many images corresponded to the same individual from different poses but in the same context), we were forced to perform a training with 650 thousand samples from the original 3 million samples in order to have an adequate proportion between the classes and achieve a good generalization of the models.

To train the model we decided to perform a 10K-Fold, i.e. to perform a 90% distribution of samples for training and 10% for testing, always keeping the proportionality between classes, and using disjoint partitions in each of the iterations.

As we trained the model we realized the overfitting that was happening on the training set, due to the relative sparsity of the sample, so we implemented the early stopping technique, so that when we detected an increase in the loss of an epoch with respect to its predecessor we cut the training instantly.

4.2 Evaluation

When evaluating the results of EfficientNet we decided to use the metrics provided by f1-score where the precision metric indicates the quality of the model in class classification referring to the confidence with which it makes each prediction and the recall metric refers to the number of samples it is able to identify correctly, being the f1-score metric the union of both precision and recall. We will be comparing the model with both datasets to know which one is better for each problem. In the final part, we will present the comparative tables of the best available models for age and gender with our best classifiers from MobileNet and EfficientNet.

In the case of age classification, we decided to make a grouping of the ages in the UTK dataset so that a direct comparison with VGG-Face2 data could be made. We grouped in 3 sets: Young from 2 to 18, Middle-Aged from 19 to 65 and finally Senior from 65 to 90. In Table 1 we can see how the performance with UTK-Face is quite superior, largely due to the fact that VGG-Face2 is much more unbalanced in the ethnicity section. Within the left table inside Table 1 we can see how in the Middle-Aged class the precision and recall is much higher than in the other classes due to the difference in sample quantity.

Table 1. Age classification. On the left using EfficientNet with UTKFace. On the right, using EfficientNet with VGG-Face2

	Precision	Recall	F1-score
Young	0.83	0.87	0.85
Middle-aged	0.96	0.87	0.92
Old	0.60	0.96	0.84
Accuracy			0.89
Macro avg	0.80	0.96	0.84
Weighted avg	0.91	0.88	0.89

	Precision	Recall	F1-score
Young	0.78	0.76	0.77
Middle-aged	0.70	0.67	0.69
Senior	0.63	0.67	0.65
Accuracy			0.70
Macro avg	0.70	0.70	0.70
Weighted avg	0.70	0.70	0.70

In Table 2 we can see the f1-score of gender recognition being similar to the age classification table, however in this case the metrics are more similar because race is not such an important factor when estimating gender.

Table 2. Gender estimation. On the left using EfficientNet with UTKFace. On the right, using EfficientNet with VGG-Face2

	Precision	Recall	F1-score
Male	0.93	0.92	0.92
Female	0.92	0.92	0.92
Accuracy			0.92
Macro avg	0.92	0.92	0.92
Weighted avg	0.92	0.92	0.92

	Precision	Recall	F1-score
Male	0.88	0.87	0.87
Female	0.82	0.84	0.83
Accuracy			0.86
Macro avg	0.85	0.85	0.85
Weighted avg	0.86	0.86	0.86

In Table 3 we present the f1-score corresponding to the models dedicated to ethnicity classification. In this case we find the largest disparity between the models due to the proportion of VGG-Face2 samples, since 84% of the samples correspond to the White ethnicity. This disproportion means that for Asian and

Black ethnicity the precision and recall are very low and for white ethnicity it is up to 90%. In this model we exponentially increased the weights of the minority classes because in the first iterations of this training the accuracy disparity between the classes was much greater. In the left table inside Table 3 we can see the best classifier with UTK-Face suppressing the 'Others' class of the dataset that was decreasing the accuracy of the remaining classes, due to the noise it introduced to the network.

Table 3. Ethnicity classification. On the left, using EfficientNet with UTKFace. On the right, using EfficientNet with VGG-Face2

	Precision	Recall	F1-score
Hispanic	0.98	0.93	0.95
Black	0.94	0.97	0.95
Asian	0.93	0.97	0.95
Indian	0.91	0.95	0.93
Accuracy			0.95
Macro avg	0.94	0.95	0.95
Weighted avg	0.95	0.95	0.95

	Precision	Recall	F1-score
Black	0.46	0.36	0.40
Asian	0.18	0.49	0.26
White	0.90	0.82	0.86
Accuracy			0.78
Macro avg	0.51	0.56	0.51
Weighted avg	0.81	0.76	0.78

In Tables 4, 5 and 6 we can see the performance of our best models compared with the best models available.

Table 4. Best age classification models

Model	Accuracy
DEX [8]	0.91
MobileNets with single head	0.93
MobileNet (age gendernet)	0.914
EfficientNet - UTKFace (ours)	**0.931**
MobileNet - UTKFace (ours)	**0.917**

Table 5. Best gender estimation models

Model	Accuracy
DADL (regression model)	0.75
MobileNet (regression model)	0.6
FaceNet (regression model) [9]	0.5
EfficientNet - UTKFace (ours)	**0.91**
MobileNet - UTKFace (ours)	**0.875**

Table 6. Best ethnicity classification models

Model	Accuracy
EfficientNet - VGG-Face2 (ours)	0.79
MobileNet - VGG-Face2 (ours)	0.82
EfficientNet - UTKFace (ours)	0.96
MobileNet - UTKFace (ours)	0.88

5 Evaluation with Masked Faces

In this section, we propose a brief comparison of the predictions made by our best models in each of the attribute classifications presenting images with facial occlusions with respect to their original non-occluded version. To perform this experiment, we have used the MLFW dataset [13] which presents several versions of the same image, a version of the original image and then several editions with masks artificially superimposed.

The results of this analysis are similar as can be seen in Table 7, both in these samples that we have presented and in the rest of the tests performed, but the confidence in the masked version result is much lower. The predictions in the unmasked versions were very reliable, similar to the UTK training, but in the masked versions the proximity between the classes was bigger, which generated confusion in the model estimation. In the case of gender recognition we have a 70% of coincidence, while in the age classification it is 80% having predicted Middle-Aged in all erroneous cases, and finally the ethnicity estimation has a 65% of coincidence with the Hispanic prediction being made in the majority of the erroneous cases.

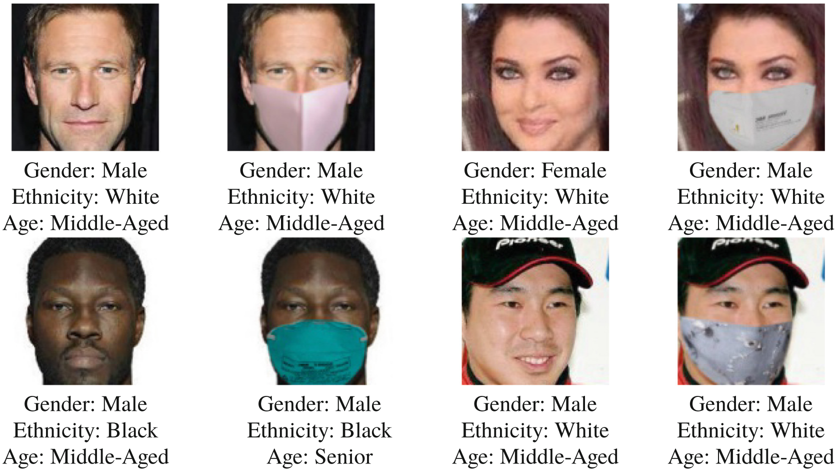


Table 7. Accuracy comparison between occluded and non occluded samples

Classified attribute	Accuracy (non occluded samples)	Accuracy (occluded samples)
Gender	0.91	0.7
Age	0.931	0.8
Ethnicity	0.96	0.65

6 Conclusions and Future Work

In this paper, we have proposed the use of several lightweight systems in the recognition of attributes in facial images, making a comparison with the best available systems. We have trained MobileNet and EfficientNet with two different datasets to make a comparison between the accuracies obtained using each dataset, the first dataset used was UTK-Face with a lower number of samples and a higher quality in terms of image preprocessing, and on the other hand VGG-Face2 with a higher number of samples of lower quality. Finally, we have made a comparison selecting our best system for the detection of each attribute with some samples of the MLFW dataset to know how mask occlusions affect the model predictions.

In our experiments, the results obtained from the training of EfficientNet have been superior to the best results of MobileNet using the same training procedures for both systems. It should be noted that the time cost of training and evaluation of the MobileNet model is slightly lower than the cost of our EfficientNet model.

In future work, we want to explore the classification of these attributes in facial images with occlusions, as this is a field that is still underdeveloped due to the scarcity of datasets with occlusions that have information about their facial attributes. We also want to explore the possibility of having the three classifiers together to perform a joint prediction with a reliability similar to that obtained with the three systems separately.

Acknowledgements. We would like to thank “A way of making Europe” European Regional Development Fund (ERDF) and MCIN/AEI/10.13039/501100011033 for supporting this work under the MoDeaAS project (grant PID2019-104818RB-I00). Furthermore, we would like to thank Nvidia for their generous hardware donation that made these experiments possible.

References

1. Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A.: VGGFace2: a dataset for recognising faces across pose and age. CoRR, abs/1710.08092 (2017)
2. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)

3. Drucker, H., Burges, C.J.C., Kaufman, L., Smola, A., Vapnik, V.: Support vector regression machines. In: Proceedings of the 9th International Conference on Neural Information Processing Systems, NIPS 1996, Cambridge, MA, USA, pp. 155–161. MIT Press (1996)
4. Howard, A.G., et al.: MobileNets: efficient convolutional neural networks for mobile vision applications. CoRR, abs/1704.04861 (2017)
5. Huo, Z., et al.: Deep age distribution learning for apparent age estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2016
6. Park, D.C., Minear, M.: A life span database of adult facial stimuli. Behav. Res. Methods Instrum. Comput. **36**, 630–633 (2004). https://doi.org/10.3758/BF03206543
7. Ricanek, K., Tesafaye, T.: MORPH: a longitudinal image database of normal adult age-progression. In: 7th International Conference on Automatic Face and Gesture Recognition (FGR 2006), pp. 341–345 (2006)
8. Rothe, R., Timofte, R., Van Gool, L.: Deep expectation of real and apparent age from a single image without facial landmarks. Int. J. Comput. Vis. **126**(2–4), 144–157 (2018). https://doi.org/10.1007/s11263-016-0940-3
9. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: a unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 815–823 (2015)
10. Tan, M., Le, Q.: EfficientNet: rethinking model scaling for convolutional neural networks. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning, Volume 97 of Proceedings of Machine Learning Research, pp. 6105–6114. PMLR, 09–15 June 2019
11. Terhorst, P., Fahrmann, D., Kolf, J.N., Damer, N., Kirchbuchner, F., Kuijper, A.: MAAD-Face: a massively annotated attribute dataset for face images. IEEE Trans. Inf. Forensics Secur. **16**, 3942–3957 (2021)
12. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, vol. 1, p. I (2001)
13. Wang, C., Fang, H., Zhong, Y., Deng, W.: MLFW: a database for face recognition on masked faces. arXiv preprint arXiv:2109.05804 (2021)
14. Zhang, Z., Song, Y.: UTK-Face: large scale face dataset. Technical report (2018)

Small Vessel Detection in Changing Seaborne Environments Using Anchor-Free Detectors on Aerial Images

Pablo Ruiz-Ponce(✉), David Ortiz-Perez, and Jose Garcia-Rodriguez

Department of Computer Science and Technology, University of Alicante,
Carretera San Vicente del Raspeig s/n, 03690 San Vicente del Raspeig, Spain
{pruiz,dortiz,jgarcia}@dtic.ua.es

Abstract. We propose an anchor-free object detector to accurately locate small vessels in changing seaborne environments for search and rescue operations using aerial images. Using the YOLOX architecture, we have achieved high accuracy and real-time inference on the SeaDronesSee dataset. In order to face the high imbalance present in the dataset, the variations of the dataset and a weighted loss have been proposed and implemented. Additionally, the proposed method has been tested on unseen images from similar datasets achieving promising results and proving the robustness of the solution.

1 Introduction

Each day hundreds of people risk their lives on different seas and oceans all around the globe in order to run away from wars, poverty and other miseries. Most vessels used are not robust enough to resist the whole crossing and, for this reason, it is crucial to locate those small vessels as fast as possible in order to provide humanitarian aid.

Different approaches have been proposed to detect vessels in seaborne environments. The most common solutions for solving similar tasks use satellite or SAR (Synthetic Aperture Radar) images to locate the vessels. While those methods provide very good results, they are not suited for this particular task because they are very expensive and they cannot provide real-time information. We propose the use of an anchor-free detector only trained on regular aerial imagery as a robust solution. This approach provides real-time and robust predictions on changing environments and scale changing targets; being successfully tested on different datasets. In addition, our proposal is able to accurately locate shipwrecked people with and without life jackets in the surroundings of the vessels.

The remaining of the paper is organized as follows: in Sect. 2 we review the current solutions to the vessel detection problem and the available datasets.

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 280–288, 2023.
https://doi.org/10.1007/978-3-031-18050-7_27

In Sect. 3 we present our approach to solve this task. We explain the conducted experiments and results on different datasets. Finally, in Sect. 5 we summarize our conclusions and propose further work.

2 Related Work

Object detection is a very well-known problem. There are several optimal solutions and architectures able to solve this task. Even for detecting vessels, multiple methods and datasets have been successfully tested. However, detecting small vessels in real-time for search and rescue operations entails certain difficulties.

2.1 Vessel Detection

Detecting ships on images is being used in many applications nowadays. Some of the most common ones are: borders and harbours surveillance [3], illegal fishing detection, autonomous sailing [4] and maritime trade routes monitoring. For this task, the current state-of-the-art approaches are based on CNN object detectors. While there are some proposals that try different approximations like time-series and attention [10], the most common solutions are based on two-stage object detectors like R-CNN [7] and Faster-RCNN [13] and one-stage object detectors like the YOLO [12] family. In order to deal with small size vessels, there are some proposals involving, time-series [11], attention mechanisms [5], backbone modifications [10] and anchor-free detectors.

2.2 Datasets

Although the architectures proposed are very standard, the difficulty of the task is to acquire data to train those models. In the field of vessel detection, the most common type of data available is satellite images and SAR (Synthetic Aperture Radar) images. The most noticeable dataset using satellite images is the Airbus Ship Detection [2] dataset. Using SAR imagery there are several options like SSDD [21], LS-SSDD-v1.0 [20], SAR-Ship-Dataset [18] and xView3 [1]. The problem with these kinds of images is that, even when they provide very good results in detecting big vessels, there are very expensive to obtain and it is almost impossible to get real-time data; being useless for real-time search and rescue missions. For this task, the best option is to use standard RGB images obtained by conventional cameras. Nevertheless, the usual images of this kind are fixed surveillance cameras located on the coast and harbours like SeaShips [15] and Boat Re-ID [16]. Despite all the difficulties, there are 2 datasets that can be used for the objective of this research (Figs. 1, 2, and 3).

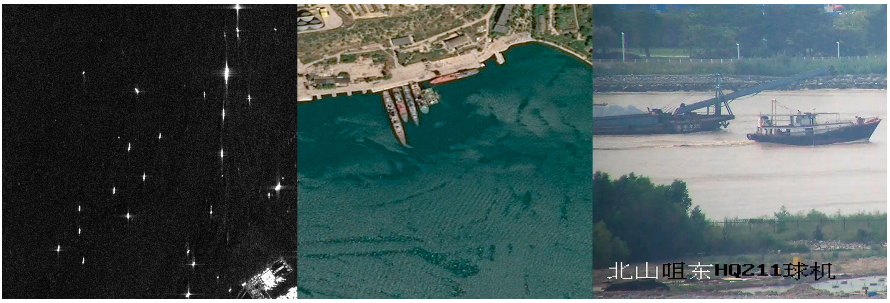


Fig. 1. LS-SSDD-v1.0 (SAR), Airbus Ship Detection (Satellite) and SeaShips (Surveillance camera) datasets example images

2.2.1 Seagull

The first dataset that fitted our proposal was the Seagull [14] dataset. It contains 366698 images extracted from 6 different videos captured by a UAV. It contains 6489 instances of boats in a maritime environment. It was a very good starting point. Nevertheless, the main problem was that all the vessels' instances and environments were very similar and the resolution of the images is quite low (256×256 pixels).

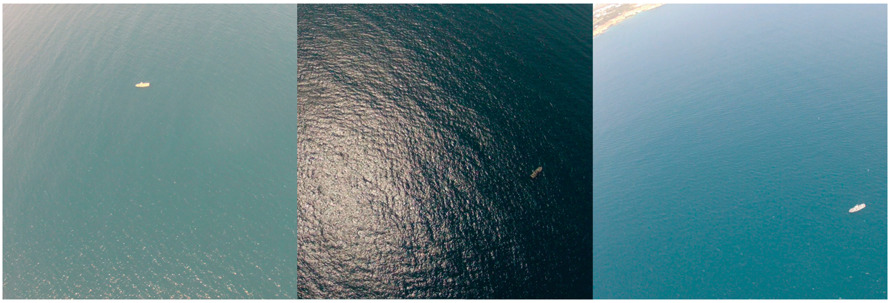


Fig. 2. Seagull dataset example images

2.2.2 SeaDronesSee

Fortunately, by the end of 2021, the SeaDronesSee [17] dataset was released. This dataset contains 54000 images with 400000 instances with a very high resolution (3840×2160 to 5456×3632 pixels) using a UAV. The different instances were captured at different altitudes, environments and viewing angles. Apart from the object detection, this dataset allows the task of object tracking and multi-object tracking. In addition to the vessels, they are also floaters and swimmer instances on the dataset. All these features make this dataset the most suitable option.

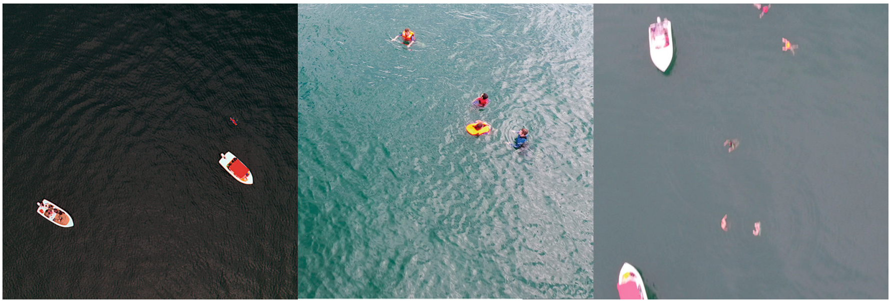


Fig. 3. SeaDronesSee dataset example images

3 Methodology

When training an object detector, several aspects should be taken into account. In this particular research, some modification has been made to the dataset and the architecture in order to achieve good results. In this section, those changes will be discussed.

3.1 Dataset Manipulation

The first problem that we encountered using the SeaDronesSee dataset is the number of instances of each class. As can be seen in the Fig. 4, there is a noticeable imbalance that will be prejudicial during the training of the model.

Our initial approach to this problem was the creation of different datasets, merging similar classes, in order to create evenly distributed datasets as can be seen in Table 1. As the main objective was to detect vessels, the first version only contained labels of the type *boat*. For the second version, 2 more classes were added. In addition to the boats, the *floater* and the *floater in a boat* classes were included. All the people on the water were included in the *floater* class (including the life jacket), while the instances of people inside the boat were merged into the *floater in a boat* class. Lastly, in order to test our model with the benchmark provided by the creators of the dataset, a third version was created. This version is almost identical to the original dataset. The only difference between the two is that the life jacket class was removed and those labels were included in the floaters class. While this change provides much more balanced data, the accuracy lost due to this change is minimal. The traditional data augmentation was considered as an option to balance the classes. However, it was discarded as the underrepresented classes always appear on the images surrounded by the majority classes.

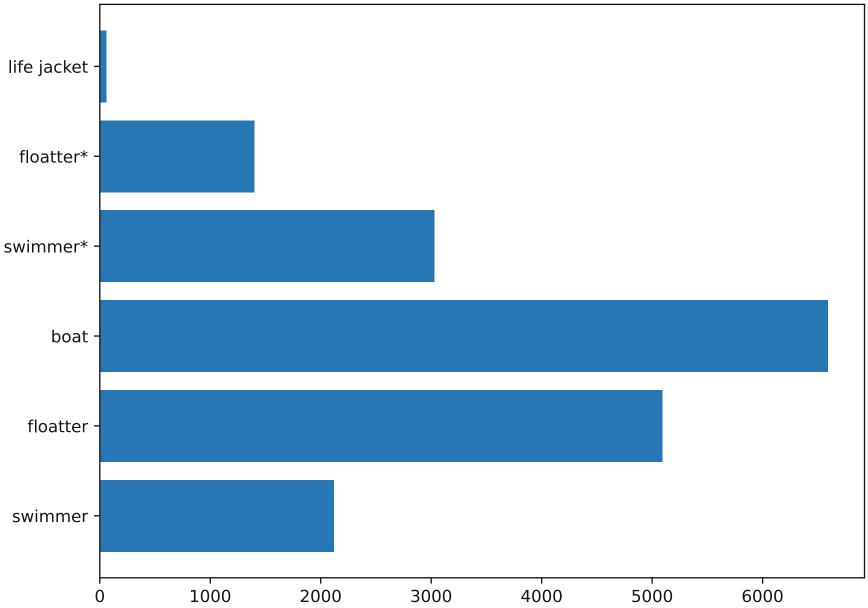


Fig. 4. Class distribution on the SeaDronesSee training set

Table 1. Number of instances of each class on the different versions of the original SeaDronesSee dataset. The * represents that this instance is inside a boat

Dataset	Swimmer	Floater	Boat	Swimmer*	Floater*	Life jacket
SeaDronesSee	2120	5092	6593	3027	1401	62
SeaDronesSeeBoats			6593			
SeaDronesSee3		7274	6593		4428	
SeaDronesSee5	2120	5154	6593	3027	1401	

3.2 YOLOX

Among all the possible architectures to perform object detection, the chosen one was YOLOX [6]. This detector is part of the YOLO family, which provide excellent results and fast inference becoming the most suitable choice. In fact, the SOTA model for this dataset was YOLOv5 [8]. Nevertheless, the choice of YOLOX over other models was based on a specific problem inherit form the task to perform. While the rest of the versions are anchor-based detectors, YOLOX is an anchor-free detector. Anchor-based detectors have proven to be an optimal approach for object detection tasks using a predefined number of anchors to locate the different objects in the scene. However, this kind of detector has problems with objects candidates with large scale variations. This scenario is very common in UAV and aerial images [19], as the camera is not in a fixed position or altitude. Taking into account that the dataset and objective of this project is to provide robust real-time inference on aerial images, using an anchor-free detector seemed the best choice.

Aiming to mitigate the class imbalance of the dataset, the class loss function on the head of the architecture has been modified. A ponderated loss has been used depending on the class predicted. The original loss function used in the model is a BCEWithLogitsLoss.

$$\ell_c(x, y) = L_c = \{l_{1,c}, \dots, l_{N,c}\}^\top$$

$$l_{n,c} = -w_{n,c}\left[p_c y_{n,c} \cdot \log \sigma(x_{n,c}) + (1 - y_{n,c}) \cdot \log(1 - \sigma(x_{n,c}))\right]$$

In this case, the parameter p_c has been used to ponderate the weight. The p_c parameter is a tensor with a length equal to the number of classes. The weights for each class have been calculated by dividing the number of instances of this class by the number of instances that are not from this class.

4 Experimentation

Once the architecture has been modified and the different datasets have been created, we trained our models and carried out various experiments. We trained our models 200 epochs starting from the pre-trained weights of the YOLOX-s architecture. The input size of the network was increased to 960 × 960 pixels due to the high resolution of the images on the dataset. The Average Precision obtained for each class on the evaluation set can be found in Table 2. The mAP (mean Average Precision) and mAR (mean Average Recall) at different IoU (Intersection over Union) values and ranges can be found in Table 3. Using the NVIDIA Tesla K80 as inference device, we have achieved real-time inference at 20 FPS.

As can be seen in Fig. 5 the results obtained by the trained models are quite promising. Nevertheless, our model has difficulties detecting people inside the

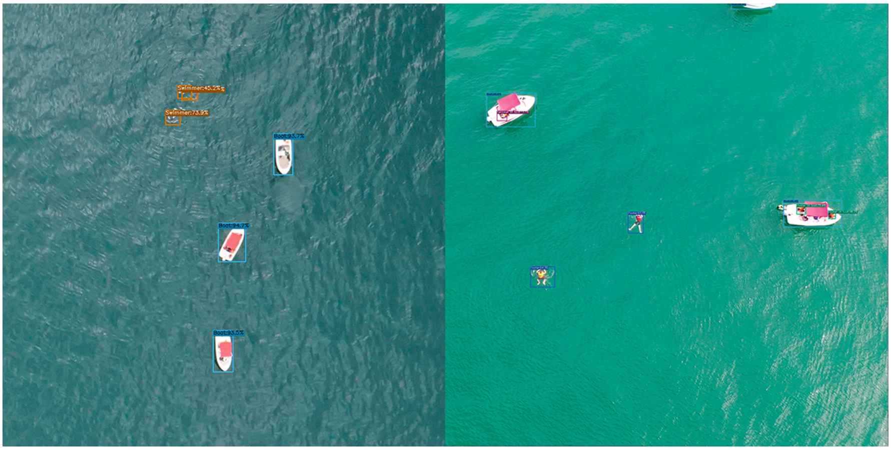


Fig. 5. Inference on the SeaDronesSee validation set using the SeaDronesSee5 model

vessels. This can be considered a minor issue as, in real-world scenarios, the priority is to localize the vessels and the people that may have fallen to the sea.

Table 2. Model Average Precision for each class using the validation set

Model name	Swimmer	Floater	Boat	Swimmer*	Floater*	Life jacket
SeaDronesSeeBoats			73.310			
SeaDronesSee3		42.378	73.939		11.332	
SeaDronesSee5	38.556	40.475	73.695	12.247	13.925	

Table 3. Model mAP (mean Average Precison) and mAR (mean Average Recall) at different IoU (Intersection over Union) values and ranges using the validation set

Model name	mAP			
	IoU = 50:0.05:95	IoU = 50	IoU = 75	IoU = 50:0.05:95
SeaDronesSeeBoats	0.733	0.957	0.879	0.786
SeaDronesSee3	0.425	0.725	0.430	0.524
SeaDronesSee5	0.358	0.666	0.328	0.457

The trained models have also been tested on other datasets to evaluate their robustness. On datasets such as LS-SSDD-v1.0 and SeaShips, the models are unable to detect any object correctly due to the visible differences to the training datasets. Nevertheless, on datasets such as Seagull and Airbus Ship Detection, the models provide accurate predictions. As the models have not been trained

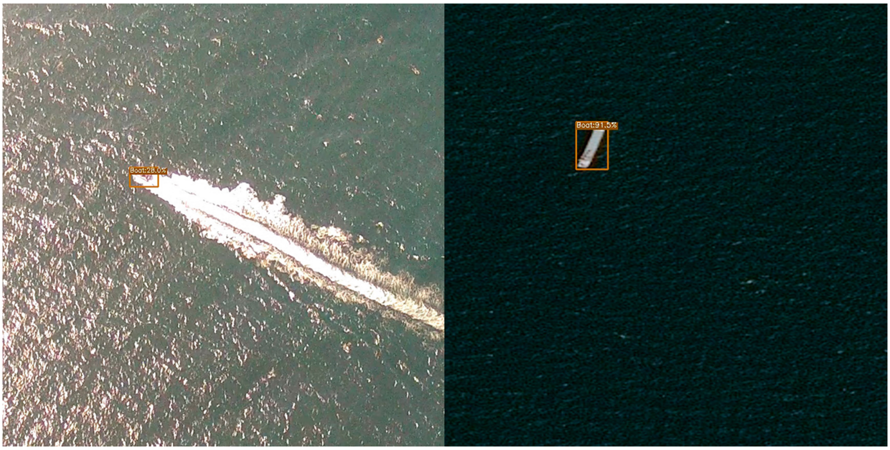


Fig. 6. Predicted labels on the Seagull and Airbus Ship Detection datasets using the SeaDronesSeeBoat model

using those datasets, several false positives are produced during the inference due to slight differences in the environment. The results obtained demonstrate the robustness of the proposed method on different seabornne environments using aerial images (Fig. 6).

The results obtained by the SeaDronesSee5 model have been submitted to the online testing platform provided by the creators of the datasets. The mAP obtained on the test set is slightly lower than the obtained on the validation set. Due to the opacity of the testing platform, we are still analyzing the results in order to improve the performance on this specific set.

5 Conclusion

We presented the use of an anchor-free one-stage object detector to locate small vessels in a seaborne environment aiming to help in search and rescue missions. We have extensively reviewed and tested different datasets, obtaining diverse models capable of performing real-time detection of vessels and shipwrecked people in changing environments. Those models have been tested on unseen images from different datasets achieving promising results and proving the models' robustness.

Despite the favourable results, there is still room for improvement. Regarding the chosen architecture, we believe that the results could be improved by adjusting some training hyper-parameters. For future work, we would also try other methods to mitigate the unbalanced dataset. Finally, to increase the robustness of the solution, we propose the use of synthetic data [9] as well as the combination of similar datasets.

Acknowledgment. We would like to thank "A way of making Europe" European Regional Development Fund (ERDF) and MCIN/AEI/10.13039/501100011033 for supporting this work under the MoDeaAS project (grant PID2019-104818RB-I00). Furthermore, we would like to thank Nvidia for their generous hardware donation that made these experiments possible.

References

1. xview3: Dark vessels. https://iuu.xview.us/dataset. Accessed 30 May 2022
2. Airbus: Airbus ship detection challenge. https://www.kaggle.com/competitions/airbus-ship-detection/data. Accessed 30 May 2022
3. Cane, T., Ferryman, J.: Evaluating deep semantic segmentation networks for object detection in maritime surveillance. In: 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1–6 (2018). https://doi.org/10.1109/AVSS.2018.8639077
4. Chen, Z., Chen, D., Zhang, Y., Cheng, X., Zhang, M., Wu, C.: Deep learning for autonomous ship-oriented small ship detection. Safety Sci. **130**, 104812 (2020). https://doi.org/10.1016/j.ssci.2020.104812
5. Dubey, S., Olimov, F., Rafique, M.A., Jeon, M.: Improving Small Objects Detection using Transformer (2021). https://doi.org/10.36227/techrxiv.16921000.v2

6. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: YOLOX: exceeding YOLO series in 2021. arXiv preprint arXiv:2107.08430 (2021)
7. Girshick, R.B., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. arXiv preprint arXiv:1311.2524 (2013)
8. Jocher, G., et al.: ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference (2022). https://doi.org/10.5281/zenodo.6222936
9. Kiefer, B., Ott, D., Zell, A.: Leveraging synthetic data in object detection on unmanned aerial vehicles. arXiv preprint arXiv:2112.12252 (2021)
10. Li, H., Deng, L., Yang, C., Liu, J., Gu, Z.: Enhanced yolo v3 tiny network for real-time ship detection from visual image. IEEE Access **9**, 16692–16706 (2021). https://doi.org/10.1109/ACCESS.2021.3053956
11. Porto Marques, T., et al.: Size-invariant detection of marine vessels from visual time series (2020). https://doi.org/10.1109/WACV48630.2021.00049
12. Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A.: You only look once: unified, real-time object detection. arXiv preprint arXiv:1506.02640 (2015)
13. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497 (2015)
14. Ribeiro, R., Cruz, G., Matos, J., Bernardino, A.: A data set for airborne maritime surveillance environments. IEEE Trans. Circuits Syst. Video Technol. **29**(9), 2720–2732 (2019). https://doi.org/10.1109/TCSVT.2017.2775524
15. Shao, Z., Wu, W., Wang, Z., Du, W., Li, C.: Seaships: a large-scale precisely-annotated dataset for ship detection. IEEE Trans. Multim. **20**, 1 (2018). https://doi.org/10.1109/TMM.2018.2865686
16. Spagnolo, P., Filieri, F., Distante, C., Mazzeo, P.L., D'Ambrosio, P.: A new annotated dataset for boat detection and re-identification. In: 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1–7 (2019). https://doi.org/10.1109/AVSS.2019.8909831
17. Varga, L.A., Kiefer, B., Messmer, M., Zell, A.: Seadronessee: a maritime benchmark for detecting humans in open water. arXiv preprint arXiv:2105.01922 (2021)
18. Wang, Y., Wang, C., Zhang, H., Dong, Y., Wei, S.: A sar dataset of ship detection for deep learning under complex backgrounds. Remote Sens. **11**(7), 765 (2019). https://doi.org/10.3390/rs11070765
19. Yang, J., Xie, X., Shi, G., Yang, W.: A feature-enhanced anchor-free network for UAV vehicle detection. Remote Sens. **12**(17), 2729 (2020). https://doi.org/10.3390/rs12172729
20. Zhang, T., et al.: Ls-ssdd-v1.0: a deep learning dataset dedicated to small ship detection from large-scale sentinel-1 sar images. Remote Sens. **12**(18), 2997 (2020). https://doi.org/10.3390/rs12182997
21. Zhang, T., et al.: Sar ship detection dataset (SSDD): official release and comprehensive data analysis. Remote Sens. **13**(18), 3690 (2021). https://doi.org/10.3390/rs13183690

Improving Malware Detection with a Novel Dataset Based on API Calls

Manuel Torres(✉), Rafael Álvarez, and Miguel Cazorla

Universidad de Alicante, Alicante, Spain
mtm41@alu.ua.es, {ralvarez,miguel.cazorla}@ua.es

Abstract. In this paper, we analyze current methods to distinguish malware from benign software using Machine Learning (ML) and feature engineering techniques that have been implemented in recent years. Moreover, we build a new dataset based on API calls gathered from software analysis, conforming more than 30000 samples belonging to malware as well as benign software. Finally, we test this dataset with an existing model that achieves accuracy rates close to 97% with a different, smaller dataset, identifying interesting results that can open new research opportunities in this field.

1 Introduction

Nowadays it is known that malware behavior can rapidly mutate, incorporate mechanisms and algorithms from different malware, or use new techniques that have not been discovered yet. As indicated by Alazab et al. [1], one of the biggest threats posed by malware is the use of code obfuscation, predominantly metamorphic and polymorphic techniques (with or without encryption), rendering traditional signature-based malware detection useless. Furthermore, researchers like Yuan et al. [2] state that those antivirus which employs this type of detection have a probability between 25% and 50% of successfully identifying malware as malicious software if it uses an obfuscation technique, which is a low rate to assume. Therefore, a more intelligent approach is required to determine if a sample is really malicious or not with better accuracy than basic systems based on checking specific features. In fact, considering the increasing success in the application of ML, automated malware detection is an excellent opportunity to take advantage of these techniques. Within the scope of ML-based malware detection, there are several possible approaches:

- Based on network traffic analysis. Many papers focus on methods related to traffic classification, most relevant ones include techniques such as traffic matching using predefined rules or the identification of patterns based on certain features [3]. The second is more versatile when it comes to solving problems in relation to the use of advanced evasion techniques that do not match predefined rules; although other problems persist, like circumventing encryption or choosing the right features in order to get the best results. Moreover, current modern malware may not generate any traffic during large intervals of time until certain required information is found.

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 289–298, 2023.
https://doi.org/10.1007/978-3-031-18050-7_28

- Based on binary analysis. Most of these approaches focus on extracting features from the executable through static analysis, considering characteristics such as PE header content, entropy, OPCode sequence, or imported DLLs to generate feature vectors and determine the nature of the binary [4]. One of the biggest problems with this approach is the fact that malware developers introduce countermeasures to prevent obtaining the actual malware binary code, usually using obfuscation techniques to completely or partially hide the real code within an executable or DLL.
- Based on behavior analysis. In most published research, the key is in the collection of API calls executed on the operating system while the software is being run; on the contrary, other aspects such as operating system or network events are usually less relevant [5]. Therefore, in order to collect this information, dynamic analysis is necessary, requiring a third-party software that runs within the same operating system as the malware, injecting itself into suspicious processes and extracting relevant information. This is a difficult task nowadays because a significant portion of modern malware implements anti-debugging or anti-Virtual Machine (anti-VM) mechanisms.

The rest of the paper is organized as follows: In Sect. 2 we review existing research that employs ML-based techniques for malware behavior detection. Next, in Sect. 3 we analyze and describe a previously published ML architecture for malware detection based on behavior, we propose a novel dataset also. In Sect. 4 we include the experimentation and discuss the results obtained using our proposed dataset. Finally, in Sect. 5, we give some conclusions and directions for future work.

2 Related Work

In this section we review some approaches that have been published in recent years related to the area of behavior-based malware detection using API calls.

In 2010, Trinius et al. [6] published a new method to represent software API calls in the form of a meta language called MIST (Malware Instruction Set), their objective was to use this method to have a common representation format for malware analysis since, at that time, there were a lot of emerging datasets related to malware identification. Notice that a representation method abstract enough to describe current and future malware could significantly improve research and collaboration in this field.

At first, this metalanguage was based on an hexadecimal trace with four levels of priority (category of the operation, affected instance, details about the instance and arguments) and twenty different categories. In Fig. 1 we can see a system call successfully executed by a software sample with the objective of loading the NTDLL library, in addition to other descriptive data such as the size of the library and involved memory addresses. As mentioned above, representing malware behavior requires a dynamic analysis provided by a third party tool; in this case, the method was coupled to the CWSandbox platform, following the common approach to extracting behavior features represented in Fig. 2.

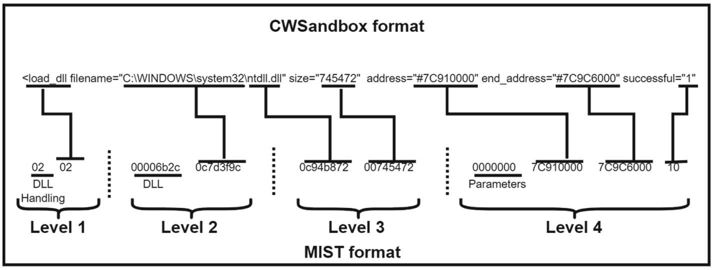


Fig. 1. Format conversion from CWSandbox to MIST.

On the other hand, it is interesting to highlight the contribution of Dima Rabadi et al. [7], which focus their experiments on the importance of the API call arguments, in contrast to others that only takes into account API call names and their frequency. They developed two feature engineering methods to represent software features, one of them treats each API call and its arguments as a single feature and the other one treats each API call and each of its arguments as a different feature. After converting these features into a binary vector and applying a MurmurHash3 function, this information is then used as input for an XGBoost model, which achieves a 96.7% accuracy in a test dataset containing about 2972 samples. Hence, this approach supports the idea of making features more specific or, in other words, taking into account arguments that could radically change the purpose of the call.

Another existing problem is concerning dataset availability, it appears that there are very few generally available in the field of malware, especially with sufficient quality. Moreover, there are several that are still in use by the scientific community that are very outdated, for example the widely recognized KD99 or NLS-KDD, as Gamage et al. indicate in their survey [8]. It is difficult to find datasets with good quality that represent malware behavior, including API calls. One of the causes of this situation is that it is hard to establish standard malware datasets, as malware features evolve rapidly, leading to a situation where researchers tend to generate their own dataset.

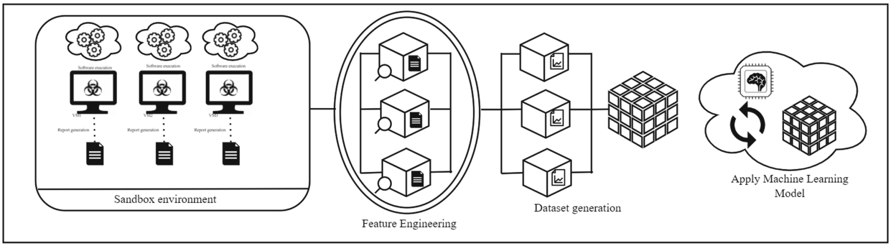


Fig. 2. Typical approach for malware detection based on its behavior

3 Proposal

In this section, we describe in detail the method presented by Zhaoqui Zang et al. [9] pointing out advantages and disadvantages. Also, we explain how a new dataset of benign and malicious software behavior with samples provided by real users has been generated, which is the main contribution for this paper[1]. The addition of this dataset to the scientific community could be key in order to improve malware detection using behavioral patterns, not only in Windows, but also in Linux.

We can assume that capturing software behavior in a format that can be easily understood by a ML model and also enable pattern recognition would be helpful in order to perform a successful classification between malicious and benign software or, at least, to establish groups of similar behavior. To achieve this, the format of that dataset should satisfy the following guideline:

1. It should be able to capture patterns, so it must be grouped into categories that define various types of actions that pursue the same final goal. Otherwise, if we were capturing individual actions, it would be practically impossible for them to be repeated, due to the variability of the arguments, obfuscation and the large number of options to do the same procedure in an actual operating system.
2. These categories must be sufficiently abstract and robust to not be altered with minimal changes when they are carried out, preventing cheating or detection avoidance.

Zhaoqui Zhang et al. [9] described the process that they followed to generate a model based on deep learning that classifies samples into malware or benign classes. Regarding the dataset format, it is based on basic JSON formatted reports generated by the CuckooSandbox tool that represent sample behavior. These samples are run inside a VM that contains an agent which records information about the actions done, sending them to a remote host. In the end, these actions are reflected in a single report for each execution of the sample and the VM is reset to an initial state before proceeding to the execution of another suspicious piece of software, avoiding any dependencies from previous executions. We can see this behavior represented in Fig. 2.

3.1 Format of the Samples

Reports contain relevant information related to sample execution and its identification, for example:

- The “info” and “target” sections provide basic data as start and end dates of the software execution, operating system where it was executed, type of software (i.e. exe, dll, etc.). Additionally, we have a “network” section where we obtain a summary of the connection attempts done by the software.

[1] https://drive.google.com/file/d/1SMBL3wvyFDOHXxC9zz3IRLcp1BdW8ZqL/view?usp=sharing.

- The “metadata”, “strings” and “signatures” sections represent the static analysis of the software in question, focusing on comparing hashes with antivirus databases and strings extracted from the binary.
- Finally, the “behavior” section details each action carried out by the processes related to the sample, splitting API calls in categories that represent the operation scope (i.e. file, crypto, network, etc.).

3.2 Used Datasets

As we can see in the left section of Table 1 (Base Dataset), Zhaoqui Zhang et al. [9] generated a dataset of about 57786 samples, divided into April and May of 2020, the months in which they were supposedly collected. These sets are subdivided into benign and malicious samples.

Table 1. Datasets distribution

Base dataset				Proposed dataset			
	Benign	Malicious	Total	OS	Benign	Malicious	Total
April	10160	15609	25769	Windows	14458	25178	39636
May	20552	11465	32017	Linux	5327	25804	31131
Total	30712	27074	57786	Total	19785	50982	70767

It should be noted that all of these examples were run in a Windows 7 operating system. This dataset is referenced in this paper as *Base Dataset*.

Zhaoqui Zhang et al. [9] applied feature hashing to the API names and their categories by applying a hash function proposed by Weinberger et al. [10] to convert them into a binary frame of limited length. From this result, they obtain two features; the API and category names hashed in addition to argument’s value and its frequency.

To generate our proposed dataset, we obtained reports from Cuckoo Sandbox Online environment (https://cuckoo.cert.ee/), which is a public platform where everyone can upload their samples and obtain behavior reports. Considering the problem of knowing with certainty whether the obtained report corresponded to a malware or benign sample, we checked the rating given by Cuckoo Sandbox (from 0 to 10) and decided the initial nature of the software according to the following:

- From 0 to 3: Benign software (not malware)
- From 3 to 7: Uncertain software (not known with certainty if it is malware or not)
- From 7 to 10: Malicious software (malware).

To reaffirm the maliciousness of each software sample, since we could not trust only on this measure, their signature was also verified against multiple antivirus vendors. In this way, uncertain samples were discarded from the final dataset and only benign and malicious samples were taken into account, trying to ensure that the dataset has as little noise as possible and that the samples used were correctly categorized. It was also decided to eliminate the samples that contained very little information because they had not been able to run.

Finally, it has been possible to obtain a dataset with a total of 39636 samples corresponding to software executions, both malicious and benign, with an average duration of 6.4 min per sample, both in Windows 7 and Linux operating systems, as we can see in the right section of Table 1 (Proposed Dataset).

3.3 Used ML Model

Regarding the ML model used by Zhaoqui Zhang et al. [9], as we can see in Fig. 3, a Batch Normalization phase (Sergey Ioffe et al. [11]) is applied to the output of the Feature Hashing, which is a binary vector of at most 102 bits long, with the aim of reducing internal covariant shift, the dependence of gradients on

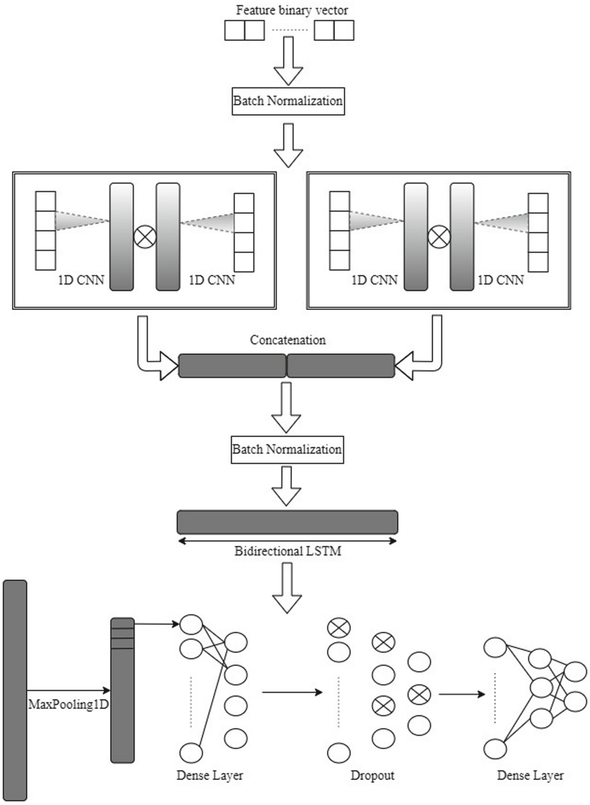


Fig. 3. ML model workflow

the scale of the parameters of their initial values, removing the need for Dropout and regularizing the entry. Then four 1D convolutions are applied and a XOR is done between pairs, finally, the two resulting outputs are concatenated. This method tries to be an approach of Gated Convolutional Networks (Dauphin et al. [12]), with the aim of ordering the features in a hierarchical way (based on trees) and having a greater abstraction in large datasets with a higher performance. A Batch Normalization is reapplied to the final result to regularize it.

The next step of the method applies a Bidirectional LSTM, which is a type of RNN (Recurrent Neural Network) that pursues the objective of analyzing the input on a recurring basis. As we have commented previously, this type of models assume the input as a chain where an element E_i will depend on the $E_{y<i}$ based on the idea that patterns can be identified based on the sequence of the sample. In this case, as it is bidirectional, it takes into account the sequence both in reverse and inverse [13]. In the classification module a GlobalMaxPooling1D is applied in conjunction with a ReLU activation function to finally apply a Dropout to the result, trying to reduce the over fitting and a second dense layer for reducing as well the dimension of the vector to 1.

Finally, it is important to consider that the model used by Zhaoqui Zhang et al. [9] is using a K Fold Cross Validation, a method based on splitting validation data into different K areas or folds. Therefore, there are K iterations varying the chosen area for testing, the others are used for training [14]. Thus, following this method they try to validate their model with different segments of samples, showing that model accuracy remains with low variance.

4 Results

In this section, we test the Zhaoqui Zhang et al. [9] model performance with their own dataset as well as with ours, discovering interesting results that could open new opportunities of research.

We trained the Zhaoqui Zhang et al. model using their training portion of the dataset that they provided, which results in four H5 files corresponding to each K fold generated in the validation phase. In essence, every H5 file is a different model that has been validated with distinct data. This is an important aspect, because we have to use each generated model with the test dataset in order to calculate a deviation value, which should not be very high.

Looking at the left section of Fig. 4, the model achieves stats closer to the ones declared in Zhaoqui Zhang et al. paper, pointing out a really low false positive rate (FPR) and a high area under the curve (AUC) value. Both values are relevant if we compare them with each other, since AUC tells us that the model will give a higher score of maliciousness to a malware sample than a benign one with a probability of around 98%, which matches with the low false positives range [15]. On the contrary, we can see a not very high recall value, which indicates that the model has room for improvement in terms of identifying malware samples. This situation interferes beneficially with the model's accuracy result due to the issue that we see in Table 1, where a higher number of benign samples were used on the dataset, about double the number of malware samples. Finally, we should note Recall values in Fig. 4, where there is a difference from 3% up to 16% between different K Fold executions, which is an indication of imbalance regarding model classification.

Our proposed dataset has more malware samples than benign, thus we are able to focus the experiments with Zhaoqui Zhang et al. model to confirm the drawbacks that we mentioned previously. Observing the achieved results with our proposed dataset in the right part of Fig. 4, we can notice a big decrease in terms of performance, achieving a low value for TPR, not even close to the value near to 73% that we can obtain with the Base Dataset, while FPR is keeping a good rate (around 0.1%). This situation shows us that the model really has problems to classify malware samples, since doubling the malware proportion for test has decreased considerably the achieved accuracy. In fact, with a TPR value as low as this, we can come to the conclusion that malware samples used in the Base Dataset for training are very different from the ones present in our Proposed Dataset.

It should be noted that we test Zhaoqui Zang et al. [9] model only with our samples executed on Windows, since it has been trained with API calls belonging to this operating system; in fact, when we try to use this model with our Linux samples, it completely fails classification, achieving accuracy rates lower than 50%. From this behavior we can conclude that the model might be highly dependent on API function names, deviating far away from the benefits of using representational formats like MIST and leading to a situation where the model is no longer functional because the system (Windows in this case) has implemented new DLLs or API calls not recognized by such model. Definitely, focusing our features more on patterns and operation categories, which is indeed a good cue in identifying software objectives, can give us a better robustness in these situations and, maybe, a higher accuracy as well.

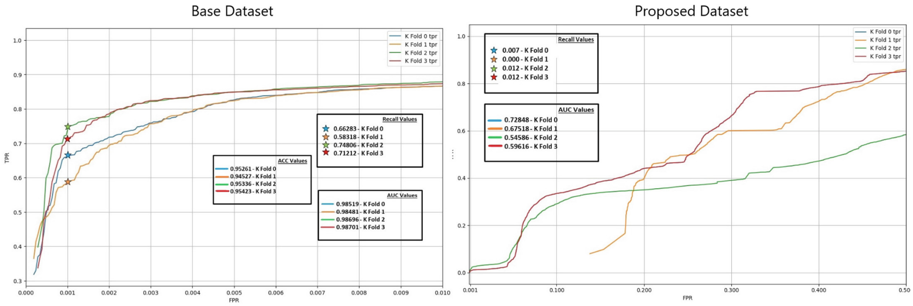


Fig. 4. ML model results with Base Dataset over different K Folds

5 Conclusion

Regarding the results obtained with Zhaoqui Zang et al. model, even though we have confirmed their published values when employing their dataset, we have identified some problems, such as the fact that the model is actually classifying benign software samples better than malware ones. Added to the fact that this type of samples have a bigger weight on the test data, and we can suspect that the results will be very different when we include a higher number of malware samples in the test data. In fact, this is something that we see when the model is tested against our proposed dataset, which is more focused on malware than the one proposed by them, highlighting the good performance of the model when identifying benign software, but a notable room for improvement in terms of malware classification as well, since it is in this case where the binary classifier fails the most.

Future research can include new Feature Engineering methods, centered more on the API call function name itself and the provided parameters, so our ML model focuses on detecting behavior patterns along a timeline. Moreover, instead of collecting mere statistical data about occurrences of our chosen features, we could try to identify common malware behavior signals for gathering internal data, persisting on the system, etc. and build statistical information about these patterns. Creating a common format for software behavior representation that is abstract enough and independent from the operating system, following the example of MIST, would be a great achievement for further research improvements in this field. Finally, the inclusion of a ML model using CNN is key for future research and development in this field, but some tweaks would be needed to overcome performance issues related to the non-linearity of string based datasets, pointing out that this kind of technique has shown its best performance with 2D images.

References

1. Alazab, M., Venkatraman, S., Watters, P., Alazab, M., Alazab, A.: Cybercrime: the case of obfuscated malware. In: Global Security, Safety and Sustainability & E-Democracy, pp. 204–211 (2012)
2. Yuan, X.: PhD Forum: deep learning-based real-time malware detection with multi-stage analysis. In: 2017 IEEE International Conference On Smart Computing (SMARTCOMP), pp. 1–2 (2017)
3. Wang, W., Zhu, M., Zeng, X., Ye, X., Sheng, Y.: Malware traffic classification using convolutional neural network for representation learning. In: 2017 International Conference on Information Networking (ICOIN), pp. 712–717 (2017)
4. Singh, J., Singh, J.: A survey on machine learning-based malware detection in executable files. J. Syst. Archit. **112**, 101861 (2021). https://www.sciencedirect.com/science/article/pii/S1383762120301442
5. Sami, A., Yadegari, B., Rahimi, H., Peiravian, N., Hashemi, S., Hamze, A.: Malware detection based on mining API calls. In: Proceedings of the 2010 ACM Symposium on Applied Computing, pp. 1020–1025 (2010). https://doi.org/10.1145/1774088.1774303

6. Trinius, P., Willems, C., Holz, T., Rieck, K.: A malware instruction set for behavior-based analysis. None (2009). https://madoc.bib.uni-mannheim.de/2579/
7. Rabadi, D., Teo, S.: Advanced windows methods on malware detection and classification. In: Annual Computer Security Applications Conference, pp. 54–68 (2020). https://doi.org/10.1145/3427228.3427242
8. Gamage, S., Samarabandu, J.: Deep learning methods in network intrusion detection: a survey and an objective comparison. J. Netw. Comput. Appl. **169**, 102767 (2020). https://www.sciencedirect.com/science/article/pii/S1084804520302411
9. Zhang, Z., Qi, P., Wang, W.: Dynamic malware analysis with feature engineering and feature learning. Proc. AAAI Conf. Artif. Intell. **34**, 1210–1217 (2020). https://ojs.aaai.org/index.php/AAAI/article/view/5474
10. Weinberger, K., Dasgupta, A., Langford, J., Smola, A., Attenberg, J.: Feature hashing for large scale multitask learning. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 1113–1120 (2009). https://doi.org/10.1145/1553374.1553516
11. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference On Machine Learning, vol. 37, pp. 448–456 (2015). https://proceedings.mlr.press/v37/ioffe15.html
12. Dauphin, Y., Fan, A., Auli, M., Grangier, D.: Language modeling with gated convolutional networks. In: Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 933–941 (2017). https://proceedings.mlr.press/v70/dauphin17a.html
13. Young, T., Hazarika, D., Poria, S., Cambria, E.: Recent trends in deep learning based natural language processing [review article]. IEEE Comput. Intell. Magaz. **13**, 55–75 (2018)
14. Refaeilzadeh, P., Tang, L., Liu, H.: Cross-Validation. In: Encyclopedia of Database Systems, pp. 1–7 (2016)
15. Fawcett, T.: An introduction to ROC analysis. Pattern Recogn. Lett. **27**, 861–874 (2006). https://www.sciencedirect.com/science/article/pii/S016786550500303X. ROC Analysis in Pattern Recognition

Identifying Places Using Multimodal Social Network Data

Luis Lucas(✉), David Tomás, and Jose Garcia-Rodriguez

Institute of Informatics Research, University of Alicante, Alicante, Spain
{luis.lucas,dtd,jgr}@ua.es

Abstract. Social media is an interesting source of information, specially when physical sensors are not available. In this paper, we explore several methodologies for the geolocation of multimodal information (image and text) from social networks. To this end, we use pre-trained neural network models for the classification of images and their associated texts. The result is a system that allows creating new synergies between image and text in order to geolocate information that has not been previously geotagged by any other way, which is a potentially relevant information for several purposes. Different experiments have been done revealing that, in general, text information is more accurate and relevant than images.

Keywords: Multimodal classification · Location-based retrieval · Transformers · Social networks

1 Introduction

We live in an age where machine learning based technologies and the analysis of large amounts of data are used in multiple domains, reaching our daily lives. For example, we are surrounded by a large number of recommender systems in all areas. There are many different techniques to process large amounts of data to obtain predictive models. Even though our society generates huge amounts of data, there is also great difficulty in obtaining it, since most of it is generated through proprietary applications from private companies that want to preserve the exploitation rights. From these data sources, those that could provide more information or potential knowledge in almost any domain are social networks. Some studies propose the use of data from social networks for any conceivable task. For example, a field that is currently of great interest is the detection of fake news in social networks [17]. There is a lot of work to be done in this area. In this paper, we propose a methodology to geolocate information in social networks through the combined use of images and text[1].

First, we have established a set of baselines from pre-trained state-of-the-art models to geolocate only text, only images, and the combination of both.

[1] Code available at https://github.com/matrox1000/geolocation.

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 299–308, 2023.
https://doi.org/10.1007/978-3-031-18050-7_29

Subsequently, we propose a series of conditional ensembles to detect in what cases we should pay more attention to text or image in order to improve the results of the baselines. We used as a benchmark the InstaCities1M dataset [9], which contains one million associated pairs of images and text from the Instagram social network, extracted by queries related to the 10 most populated English speaking cities over the world. It comprises 100,000 image-text pairs for each city.

The remainder of the paper is structured as follows: Sect. 2 shows related work; Sect. 3 outlines the elements of the system that we use in our experiments; Sect. 4 describes the image and text classification, the ensemble systems workflow and the validation process with the results obtained; finally, in Sect. 5 we show the information obtained, discuss conclusions and suggest future work.

2 Related Work

There are numerous works related to multimodal machine learning as well as multimodal classification tasks. This section highlights some of the most relevant works in this area.

In the field of multimodal machine learning, [2] proposes two evolutions of existing models, VL-T5 and VL-BART, for vision and text to generate text. In [11], they proposed a multimodal system that uses both the text and visual content on Twitter to classify information during emergencies. For this purpose, they use LSTM and VGG-16 for tweet texts and images, respectively. Along the same line, other proposals [12] defined models to detect disasters in cultural heritage from social media information, but in this case only from images. In a completely different field, [1] used the relationship between images and their associated texts in social media to try to detect sarcasm. To this end, they build a single vector from the features of the image and the embeddings of the text. Finally, in [7] the authors proposed a common image-text embedding space for training a bidirectional network. They provide retrieval results on Flickr30K and MSCOCO datasets that considerably exceed the state-of-the-art.

With respect to classification into unsupervised or semi-supervised techniques, in [15] the authors proposes a method for the integration of natural language understanding in image classification to improve classification accuracy by making use of associated metadata. On the other hand, in [3] they proposed a novel deep learning-based multimodal fusion architecture for classification tasks, which guarantees compatibility with any kind of learning models, deals with cross-modal information carefully, and prevents performance degradation due to partial absence of data. Finally, the work in [6] presented simple models that combine information from image a and text to classify social media content. In this work, a pooling layer and an auxiliary learning task is used to learn a common feature space.

3 Description of the System

This work proposes a system capable of forecasting the location referenced by a combination of image and text extracted from social networks, either because there is a reference in the image, in the text, or in both. The input of the system is a combination of text and image and the output is the location. The objective of this system is to build synergies between text and image. Although there are times that text may be more relevant than image in the prediction or vice versa, there is also situations where only the combination of both data allows making the correct prediction.

To perform the experimentation, we have reduced the problem to a multiclass classification task over 10 classes, corresponding to the 10 cities in the InstaCities1M dataset, formed by pairs of images and text obtained from the Instagram social network associated with one of the 10 most populated English speaking cities all over the world. For its construction, a search was performed on the text of the city's name. It contains 100k images-text pairs for each of the 10 cities, resulting into a 1 million images-text set.

We have chosen this dataset for the following reasons: first, it is formed by recent social media data. The text associated with the images is the description and the hashtags written by the photo up-loaders. Second, each image is associated to one city, so we can use this weakly labeled image classes to evaluate our experiments. All images were resized to 300 × 300 pixels.

For the classification of the text component, we use the BERT language model [4] and for the classification of the images we use RESNET18 [10]. Subsequently, we process the prediction vectors into an ensemble that provides the final prediction.

Figure 1 shows the data flow of the system, whose components are detailed in the next section.

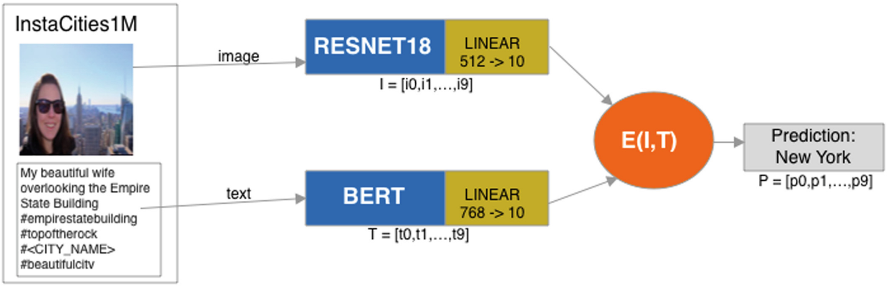


Fig. 1. Components of the proposed classification system.

Table 1. Accuracy obtained with BERT in the text component depending on the selected textual anonymisation strategy.

Anonymisation strategy	Accuracy
Original	76.61%
Categorical	64.17%
Removal	63.44%

4 Experimental Setup

For this work, first of all, we have established a baseline series using state-of-the-art models in both text and image classification.

4.1 Baselines

For the text component we used the pre-trained BERT model [4] from the Hugginface library.[2] This model consists of a pre-trained bidirectional transformer using a combination of masked linguistic modeling target and next-sentence prediction on a large corpus. We fine-tuned the model following the recommendation by the authors, placing an output linear layer that transforms the 768 outputs into 10, corresponding to the number of classes (cities). We trained with a dropout of 0.3 [8] and used the optimizer AdamWW [20].

For all training we have considered an Early Stopping system with a patience of 3, so that epochs vary depending on the improvement or not in the validation dataset. Finally, we have considered 3 versions as baselines depending on the anonymisation strategy[3] used for the text-only experiments: raw text, categorical anonymisation and removal anonymisation. Results are shown in Table 1.

For the visual side, we used two different approaches in classification in order to contrast the results. On the one hand, we used a convolutional neural network (CNN), RESNET18, with 18 layers deep. Specifically, we used a pretrained version of the network trained on more than one million images from the ImageNet database [16]. This pretrained network can classify images into 1,000 object categories, such as keyboard, mouse, pencil, or different breeds of animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224 × 224. We have added a final linear layer to obtain 10 outputs corresponding to the 10 cities to be classified and a fine-tuning with Early Stopping, dropout of 0.3, batch size of 32, adaptive learning rate [19] and ADAM optimizer.

[2] https://huggingface.co/bert-base-uncased.

[3] If the name of the city to predict is on the text, it is removed.

Table 2. Input image size, numbers of epochs used for fine-tuning and accuracy obtained with RESNET18 and ViT models in the image component.

Model	Input	Epochs	Accuracy
RESNET18	256×256	7	27.40%
ViT	224×224	4	11.93%

The other experiment, carried out for the vision part, consists of a transformer-based approach, Vision Transformer (ViT) [5], which achieves excellent results compared to state-of-the-art convolutional networks while requiring substantially fewer computational resources to train. For the training task we have used the same configuration as the experiment with the RESNET18 and the same architecture, adding a final linear layer with 10 outputs to the base variant of 12 layers with a patch of 16.

Table 2 shows that results are significantly worse than those obtained by text. This may be due to the fact that images may have less references to the city than text. We show different examples in the next section.

4.2 Combining Image and Text

Once we have configured the baselines based on state-of-the-art models, where we obtain the best possible classification results using only one type of data as input (either images or text from the dataset), the objective is to merge the output of both models in such a way that we can obtain an improvement in accuracy. To this end, first we will do an exploratory analysis of the results.

First of all, if we look at some of the misclassified images in the Fig. 2, we can see that a model based only on images could hardly make a correct prediction.

On the other hand, texts are more appropriate to get a better prediction. We can see the texts of the images in the previous figure in Table 3.

Despite the anonymisation of the name of the cities/targets, we can see references in the text that could help the model to make a correct prediction. For instance, if we look at Image 2, we can see in the text clear references to the region and country where the city to be predicted is located. An example of misclassification is shown in Image 1. In the text there is a reference to the country where the city is located, but since there are more cities of the same country among the target classes, the prediction ends up being incorrect.

On the other hand, there are images that have a clear reference to the target city and are correctly classified with the image-based model only. We can see some examples in Fig. 3.

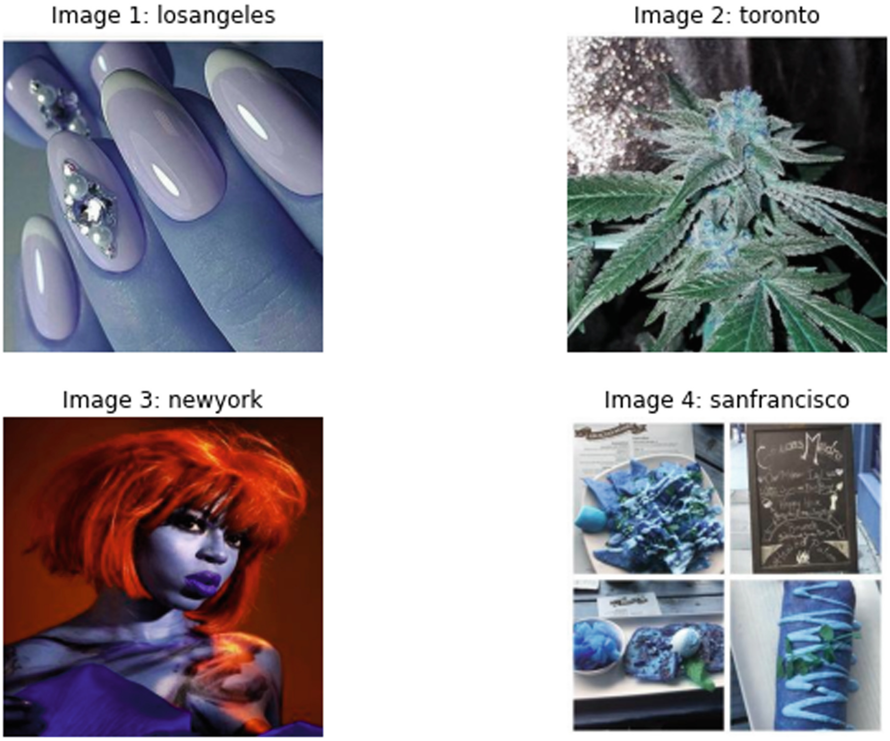


Image 1: losangeles

Image 2: toronto

Image 3: newyork

Image 4: sanfrancisco

Fig. 2. Examples of images misclassified.

4.3 Evaluation

Table 4 shows the results of all tested systems order by accuracy. All the ensembles improve the accuracy obtained with a single model. The best result is obtained with the sum of the output vectors of both models without normalization.

The second and third best results are obtained by weighted sums, giving two times more weight to the probabilities obtained with text than to those obtained by image, and 80% to text and 20% to image, respectively.

The ensemble at position Top-4 consists of the sum of the previously normalized L1 text and image probability vectors.

The most complex ensemble we have elaborated reaches the top 5 and is based on comparing the confidence of the results obtained with text and image. To do that, the first step is to normalize the output vectors of the models between -1 and 1. In that way, we can compare the best result of each prediction and keep the best one. For example, if the image model predicts city X with 0.6 and the text model predicts city Y with 0.3, we discard the text-based prediction and keep the image-based one.

Table 3. Examples of predictions based on original text component not anonymised.

<table>
<tr><th>Image</th><th>Target</th><th>Pred. text</th><th>Pred. image</th></tr>
<tr><td>1</td><td>losangeles</td><td>newyork</td><td>singapore</td></tr>
<tr><td colspan="4">#paradise #amazing #classy#chic #beautiful #wonderful#love #pretty #beautiful #nails#art #fashion #time #good#great #divine #sublime#shinny #skinny #cute #sweet#makeup #losangeles #usa#trendy #fancy</td></tr>
<tr><td>2</td><td>toronto</td><td>toronto</td><td>singapore</td></tr>
<tr><td colspan="4">Our Ultra Rare Strawberry Las Vegas Bubba Kush! Grown by #sincityseeds#homeofthedank #slvbk #strawberrylasvegasbubbakush #medicalmarijuana #cannabiscommunity #cannabis #weed #stoners#blaze #marijuana #ganja#stoner #maryjane #alberta#quebec #canadianstoner#surrey #toronto #montreal#ontario #ottawa#britishcolumbia #calgary#canadian</td></tr>
<tr><td>3</td><td>newyork</td><td>toronto</td><td>chicago</td></tr>
<tr><td colspan="4">Tired from today's preparation and shooting but it was worth it. Shout out to my team member @kissmybootyque for the makeup, shout to @jazzi_juice out for modeling and being patient with the body painting and all, and really be shout to @bohemian.photo for pointers on lighting and a great studio experience. Pics coming soon Mua/ assistant: @kissmybootyque Model: @jazzi_juice #ART#artistikmind #fashion#fashion week #newyork#vogue #blackgirlmagic#superbowl #nymodel#beautiful #afropunk#fashionillustrator...</td></tr>
<tr><td>4</td><td>sanfrancisco</td><td>sanfrancisco</td><td>sydney</td></tr>
<tr><td colspan="4">Happy Easter! Vegan for the animals, for the environment & for my health# House made chips tossed in a creamy salsa roja with cashew cream & cilantro. Mariposa bakery toast dipped in plantain batter & grilled topped with maple syrup, citrus cashew whipped cream, pecans & fresh fruit Tempeh chorizo, caramel onions, red peppers, roasted potatoes, black beans and cashew chipotle aioli served with smoked tomato salsa and ranchera. #vegan #instavegan#vegansofig #veganaf#vegansofinstagram#vegangirl #plants ...</td></tr>
</table>

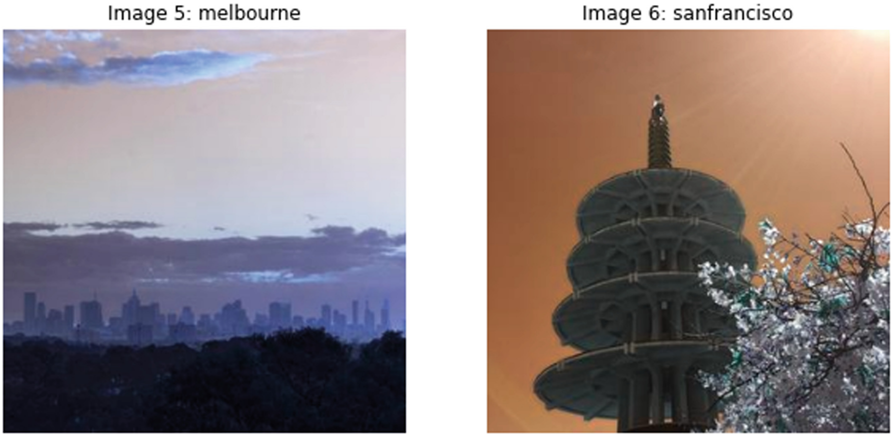


Fig. 3. Examples of images correctly classified based only in image model

The last two experiments listed in the table show the best results obtained with only text and only image, respectively.

Table 4. Accuracy obtained with the mixing ensemble experiments.

Top	Ensemble	Top 1 Acc	Top 3 Acc
1	$P = T + I$	0.66394	0.84640
2	$P = 2 * T + I$	0.66208	0.84547
3	$P = 0.8 * T + 0.2 * I$	0.65578	0.84173
4	Normalized SUM L1	0.64470	0.83392
5	Top predic value between text/image	0.64193	0.82933
6	BERT only text	0,64174	0.82942
7	RESNET18 only image	0.27395	0.52431

5 Conclusions and Future Work

In this work, we presented several proposals to geolocate image-text pairs from a social network. First, we established some baselines based only on text or image, comparing then the systems we developed combining both data types. After an exploration of the dataset based on the results obtained with the base models, we observed that synergies can be established between both data by means of ensembles. Finally, we presented the results obtained from five fusion systems. We can affirm that the multimodal models proposed improve in all cases the results obtained with a single data input. Although it seems that the improvements are not impressive, the accuracy is improved by almost 2.

This is an initial experimentation with margin for improvement. For example, we still need to exploit the conditional ensemble part by balancing the weight of attention to text and image, instead of simply dropping the worst result. Other alternatives should also be explored, such as replacing the ensemble with a mixed model that could be trained in conjunction with the text-only and image-only models.

It has also been left for future work to explore the use of multimodal neural network models such as VisualBERT [13] or LXMERT [18] and their adaptation to multiclass classification. These models have not been tested in this first approach as their architectures have been designed for other types of tasks such as Image Question Answering (QA).

Finally, a promising avenue of research is the integration of this system with other capable of detecting or cataloging specific types of places for a specific purpose such as the tourism sector, as it was done in our previous work [14]. In this way we could not only detect tourist sites limited to a region or city, but we could do it globally, being able to obtain as complementary data its geolocation.

Acknowledgement. This research has been partially funded by project "Desarrollo de un ecosistema de datos abiertos para transformar el sector turístico" (GVA-COVID19/2021/103) funded by Conselleria de Innovación, Universidades, Ciencia y Sociedad Digital de la Generalitat Valenciana and "A way of making Europe" European Regional Development Fund (ERDF) and MCIN/AEI/10.13039/501100011033 for supporting this work under the "CHAN-TWIN" project (grant TED2021-130890B-C21). We also would like to thank Nvidia for their generous hardware donations that made these experiments possible.

References

1. Cai, Y., Cai, H., Wan, X.: Multi-modal sarcasm detection in Twitter with hierarchical fusion model. In: ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, pp. 2506–2515 (2020)
2. Cho, J., Lei, J., Tan, H., Bansal, M.: Unifying vision-and-language tasks via text generation. PMLR **139**, 1931–1942 (2021)
3. Choi, J.H., Lee, J.S.: EmbraceNet: a robust deep learning architecture for multimodal classification. Inf. Fusion **51**, 259–270 (2019)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, vol. 1(MLM), pp. 4171–4186 (2019)
5. Dosovitskiy, A., et al.: An image is worth 16 × 16 words: transformers for image recognition at scale (2020)
6. Duong, C.T., Lebret, R., Aberer, K.: Multimodal classification for analysing social media (2017)
7. Image-Text Embeddings, Wang, L.: Learning deep structure-preserving. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (Figure 1), pp. 5005–5013 (2016)
8. Fan, A., Grave, E., Joulin, A.: Reducing transformer depth on demand with structured dropout, vol. 103, pp. 1–15 (2019)
9. Gomez, R., Gomez, L., Gibert, J., Karatzas, D.: Learning to learn from web data through deep semantic embeddings. In: Leal-Taixé, L., Roth, S. (eds.) ECCV 2018. LNCS, vol. 11134, pp. 514–529. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11024-6_40
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 770–778, December 2016
11. Kumar, A., Singh, J.P., Dwivedi, Y.K., Rana, N.P.: A deep multi-modal neural network for informative Twitter content classification during emergencies. Ann. Oper. Res. 0123456789 (2020)
12. Kumar, P., Ofli, F., Imran, M., Castillo, C.: Detection of disaster-affected cultural heritage sites from social media images using deep learning techniques. J. Comput. Cult. Heritage **13**(3), 1–31 (2020)
13. Li, L.H., Yatskar, M., Yin, D., Hsieh, C.-J., Chang, K.-W.: VisualBERT: a simple and performant baseline for vision and language, vol. 2, pp. 1–14 (2019)

14. Lucas, L., Tomás, D., Garcia-Rodriguez, J.: Sentiment analysis and image classification in social networks with zero-shot deep learning: applications in tourism. In: Sanjurjo González, H., Pastor López, I., García Bringas, P., Quintián, H., Corchado, E. (eds.) SOCO 2021. AISC, vol. 1401, pp. 419–428. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-87869-6_40
15. Miller, S.J., et al.: Multi-modal classification using images and text multi-modal classification using images and text. SMU Data Sci. Rev. **3**(3), 6 (2020)
16. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vision **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
17. Saquete, E., Tomás, D., Moreda, P., Martínez-Barco, P., Palomar, M.: Fighting post-truth using natural language processing: a review and open challenges. Expert Syst. Appl. **141**, 112943 (2020)
18. Tan, H., Bansal, M.: LXMert: learning cross-modality encoder representations from transformers. In: EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference, pp. 5100–5111 (2019)
19. You, K., Long, M., Wang, J., Jordan, M.I.: How does learning rate decay help modern neural networks? (2019)
20. You, Y., et al.: Large batch optimization for deep learning: training BERT in 76 minutes (2019)

Monitoring Human Performance Through Deep Learning and Computer Vision in Industry 4.0

David Alfaro-Viquez[1], Mauricio-Andres Zamora-Hernandez[1(✉)], Manuel Benavent-Lledo[2], Jose Garcia-Rodriguez[1], and Jorge Azorín-López[2]

[1] University of Costa Rica, San Pedro, Costa Rica
{david.alfaro,mauricio.zamorahernandez}@ucr.ac.cr
[2] University of Alicante, Alicante, Spain
{mbenavent,jgr,jazorin}@ua.es

Abstract. The advent of Industry 4.0 is revolutionizing manufacturing processes through techniques that can optimize the decision-making based on manufacturing data. Monitoring the whole production process from raw material input to the final product includes the production process itself and the human resources that carry it out. One of the key aspects of this decision-making process is the monitoring of human performance. This paper presents an architecture for real-time monitoring of manufacturing activities including the operator performance. As a case study, the assembly of an electro-pneumatic circuit has been taken as an experiment and a deep learning model has been trained to take as a reference the assemblies performed by experts, in addition to the standard times of these, to identify both hand trajectory and the position of the objects. The deviations obtained with respect to these references are attributable to the operator's experience or fatigue.

1 Introduction

Industry 4.0 (I4.0) is the combination of advanced production and operations techniques with intelligent technologies, through the exchange of information between parties to generate added value to the production processes [6,22]. According to Lu (2021), I4.0 can be used in manufacturing systems in vertical and horizontal integration, across value chains in product life cycle with the integration of digital processes [16].

Many of these processes are oriented to decision making and automatic controls through data. One of the main uses of automatic control using Deep Learning (DL) in I4.0 is the quality inspection through Computer Vision (CV) [10,17,19,24]. The quality control through CV is mainly used in verification control of shape, colors or other characteristics that can be observed by a camera [10,17,19,24]. However, several researchers have proposed uses beyond just analyzing quality through CV in I4.0, incorporating analysis and evaluation in the manufacturing of products with DL techniques [7,10,18]. Different works have

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 309–318, 2023.
https://doi.org/10.1007/978-3-031-18050-7_30

extended the use of CV into more innovative applications, instead of just determining the quality of components or products, incorporating DL techniques to evaluate the entire assembly process [5,18,20,26]. The use of DL and CV leads to an increased productivity by reducing errors in assembly processes or identifying them before moving on to other stages of the production process [26]. One of the key issues affecting operator performance in manufacturing systems is fatigue.

For fatigue identification, several works have been developed considering safety in driving environment with the purpose of having smarter cars. It is common analyzing the human face by taking reference markers of the face such as the eyes and mouth of people. For example, in [15], deep learning algorithms are used to identify fatigue in drivers by analyzing the frequency with which drivers blink and yawn, as well as the orientation of the head. Another method to detect fatigue is proposed by [9], it analyzes electroencephalic waves (EEG) and electro ocular signals (EOG) as signals to detect fatigue in drivers, these signals serve as input for a LTSM network. In [14], a convolutional neural network is used to analyze the image taken of the driver. From this image, information from the eyes and mouth is extracted, different parameters such as PERCLOS are extracted to identify the state of the eyes, likewise, the mouth characteristic parameter m-close, and the face orientation parameter ph-down are obtained, these parameters are used as input for the LTSM network whose output is the fatigue level. Proposals such as [15] can be pointed out, where a fuzzy logic system is used to obtain as system output the fatigue level of the person, as example, it can be known if the signs of fatigue correspond to mild, moderate or severe fatigue.

Other important aspect for the progress of Industry 4.0 is human action recognition. In this area, many works has been developed for applications in surveillance, tracking, healthcare, and human computer interaction [1]. These applications can be extended in the manufacturing field, for example in human robot interaction, citing the work of [25], which indicates that visual observation of the operator's movements provides a context of the activity being performed, knowing this context in a timely manner increases safety and efficiency in human robot collaboration. They develop a CNN convolutional neural network adapted from AlexNet for action recognition, the work was tested on the assembly of a motor, having a 96% accuracy on action recognition. In the research work by [23], the use of synthetic videos for activity tracking and recognition is discussed. It uses 3D monocular reconstruction of the human body to generate synthetic videos and label activities. A new data generation methodology, SURREACT, is introduced to train spatio-temporal CNNs for action classification. It is also important to highlight the research proposal of [21] where they use a dataset such as UTD-MHAD to recognize surveillance related activities. In their research they perform several experiments and can to recognize theft related actions.

In the specific area of human monitoring in assembly operations, deep learning can be used to reduce errors and rework. Zamora et al. [26] uses deep learning as a tool to monitor assembly activities to assist the operator and point out the correct procedure and tool to increase productivity. The use of tools such as augmented reality is also effective in the process of assisting the operator in their work. In [11], they propose a deep learning model trained through synthetic images of tools, and through the augmented reality system assists the operator in their work. In this research the errors and time spent in the process are reduced by 32.4% and 33.2% respectively. Other sectors such as the construction industry have also benefited from deep learning tools, as demonstrated by [18]. Monitoring assembly operations, object detection and recognition also play an important role. Research proposals such as [4], use the YOLO (You Only Look Once) sensing system to create an architecture to recognize interactions of people in a room to help people with memory loss recollect their daily routines. An important activity in sensing is being able to locate the position and orientation of objects, they use color identification for position and depth analysis to estimate orientation [8].

After reviewing the state of the art, to the best of our knowledge, no works were found that allow representing manufacturing processes with time indicators or productivity levels. Therefore, it was necessary to generate a proposal for this representation. Our proposal consists of extending our previous work [26] by incorporating operator fatigue analysis to determine errors and production time affectation. This is to generate a performance monitoring system that the operator and managers can use for decision making in real time.

The remaining of the paper is organized as follows; Sect. 2 describes and finally the recognition of objects. Section 3 describes the research proposal for this work as an extension of the research carried out by Zamora-Hernandez et al. [22]. Section 3 details the experiments carried out for the validation of the work. Finally, Sect. 4 summarize the conclusions of the work carried out.

2 Model Proposal

The proposal of this work is to extend our previous research by Zamora-Hernandez et al. [26]. Therefore, we propose to incorporate fatigue analysis and standard time evaluation to establish productivity indicators, probability of fatigue failure as an added value to the previous research. Specifically, the purpose of the research is to generate a performance monitoring system, by contemplating the correlation of fatigue and the propensity of failures due to fatigue. So, we can generate calculations of productivity indicators and their respective variance with respect to standard time. In manufacturing processes, it is necessary to take measurements and controls for decision making. Therefore, the first part of the research focuses on the presentation of the activities so that a Deep Learning architecture can measure the fulfillment of the activities.

The proposed model is composed by four modules: manufacturing description language interpreter, manufacturing process analyzer, fatigue analyzer and recommendations generator (see Fig. 1).

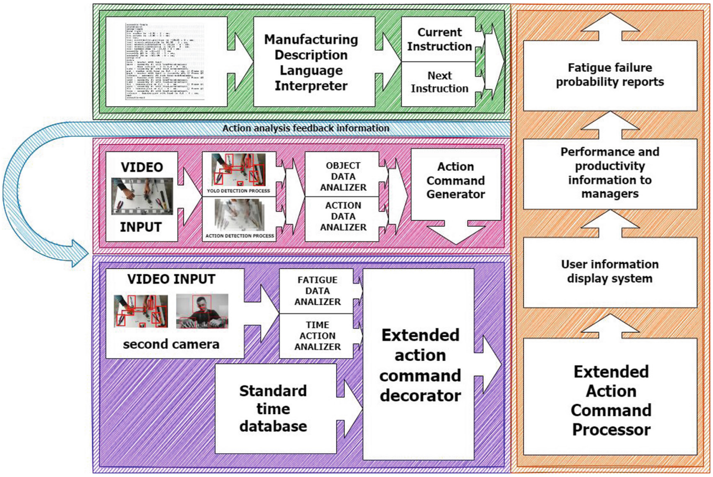

Manufacturing Description Language Interpreter
Current Instruction
Next Instruction
Action analysis feedback information
VIDEO
INPUT
YOLO DETECTION PROCESS
ACTION DETECTION PROCESS
OBJECT DATA ANALIZER
ACTION DATA ANALIZER
Action Command Generator
VIDEO INPUT
second camera
FATIGUE DATA ANALIZER
TIME ACTION ANALIZER
Standard time database
Extended action command decorator
Fatigue failure probability reports
Performance and productivity information to managers
User information display system
Extended Action Command Processor

Fig. 1. Research proposal components (Color figure online)

2.1 Manufacturing Description Language

The first section of the architecture is related to the representation of the actions, as can be seen in Fig. 1 in the green colored part. It is necessary to provide to the system the instructions of how the tasks should be performed, the MDL (Manufacturing Description Language) [27] is used as a base, which is also a previous proposal made by the research team. For this research, it was necessary to restructure the syntax and incorporate the information of the duration of the standard time of the task.

Therefore, it was necessary to readjust the Manufacturing Description Language Interpreter to interpret the new version of the language. In addition, the outputs of this module were left in the same format as the original version, only adding metadata including the duration of the tasks.

Moreover, it is necessary to redefine how to document the activities. For this purpose, a new high-level sentence structure will be used. For this purpose, we seek to follow the structure: Action_task = subject + verb + predicate + (duration). The predicate would have its own structure: Predicate = object that receives or executes the action + detail of the action.

The subject can be the operator, machine or tool that generates the task.

- E.g. Operator screws with the electric screwdriver the bolt (15 s)
- Operator flips the component with both hands (3 s)

2.2 Manufacturing Process Analyzer

The second section of our proposal is the part in charge of processing the video from the camera that monitors the assembly actions and generates real-time information of the operator's actions. It can be seen in Fig. 1 in the red colored part. This module was incorporated in the new proposal to measure the time of each action. This is a necessary input for performance monitoring in the following modules.

The output of this module is an "Action Command", which is a type of record containing the action information that was performed by the human. It contains the present elements, such as tools, accessories, or other components needed in the assembly process, and also the time information, in seconds, in which the activity was concluded.

2.3 Fatigue Analyzer

The third module, which is the core of the new proposal, is the fatigue analyzer. In the diagram shown in Fig. 1, it can be seen in purple.

This section is fed by the information generated by the previous modules. With the 'Manufacturing Description Language Interpreter', it is generated the action control networks, which will be detailed in the next paragraphs. It also uses the information generated in the second module, with this input it can be determined which task is currently being executed, as well as the time information used to complete that activity. From this same module, it takes advantage of the information from the primary camera that obtains raw information of the operator's actions on the assembly to obtain the characteristics of the hand movements in order to determine if there are variations in the way the tasks are performed. These variations will be identified through DL networks. If there are variations in the actions, it will be a sign of fatigue in the human operator.

From the input coming from the Manufacturing Description Language Interpreter, this module generates the following elements: First, a weighted undirected graph is constructed with the instructions written with the language. Each node of the graph is linked by two arcs, where one arc is the value of the probability of passing between nodes and the second arc is the standard time that the task should be performed.

The standard time is the basis that will be used as a control parameter for the complete analysis. For the operation of the fatigue analysis, other parameters are integrated that can be determined using computer vision through a set of cameras arranged to evaluate the operator and his work environment.

This module also analyzes the variations of the measured times in conjunction with the standard time to determine fluctuations. The fluctuations together with the camera input, which collects information from the operator's gestures, calculate performance indices of the operator's work.

The results of this analysis are incorporated into the 'Action Command' generated in the previous module in the form of the 'decorator' pattern, where the additional layer to the AC contains information about the operator's performance.

2.4 Recommendations Generator

The last module is in charge of taking all the inputs from the previous modules to generate the recommendations and the performance monitoring system. In the diagram shown in Fig. 1, it can be seen in orange color. This module initially takes the inputs generated by the CDM analyzer to build a graph that determines the routing sequences for the correct assembly. By incorporating the standard time, the system can also calculate the time required for assembly.

From the above modules, the system takes inputs from the vision systems (the raw information from the camera to the work area and operator), the results of the actual operation, fatigue analysis, real time, standard time used as a reference. With the above inputs, predictions are generated using DL networks to determine performance monitoring results. In addition, information on probability of failures due to fatigue is provided. The system is also able to adjust the recorded standard times by incorporating the experience factor for task repetition. A feature inherited from previous research is allows maintain a real-time notification system to the user about the status of the task being contemplated. This information also includes whether the job is performing below or above the standard time.

In Fig. 1, a light blue arrow can be seen. This is not a module of the solution; it represents the system feedback to update the status of the current task execution and optimization parameters.

As mentioned above, this is an extension of the work of [26]. In this work we intend to detect the objects that are part of the assembly to be performed, as well as the user's hands and their trajectory. A system has been trained and fed with assemblies developed by experts, which are used as a reference both for the estimated assembly times and for the system to recognize the correct sequence of the operations to be performed, so that any deviation that may exist from these expert references would be attributable to the inexperience of the users or to their fatigue, so that both the time of the operation and the sequence of the activities to complete it would be evaluated.

3 Experimentation

To test the proposed model, an experiment was carried out in the Robotics Laboratory of the Industrial Engineering degree at the Interuniversity Campus of Alajuela, University of Costa Rica. The purpose of the experiment was to obtain the standard times to perform the assembly operation of an electro-pneumatic circuit. To take these times, 3 types of operators were measured: user 1 is the professor of the course, that is, an experienced user; user 2 is a student with an average knowledge of pneumatics, and user 3 is a user with no experience in the subject of electropneumatics. The units in the table where the results are shown are in seconds.

The purpose of this time measurement is to know the average time that a person takes to perform an operation, to be later analyzed in the determination of fatigue, that is, to analyze how the processing time is impacted when the person is fatigued (Table 1).

Table 1. Time analysis of user 1 (s)

Observations	Day 1	Day 2
1	64,2	73,8
2	63,9	75,6
3	70,1	64,7
4	66,9	72,4
5	66,0	66,5
6	65,9	71,4
Mean	66,2	70,7
Deviation	2,2	4,3

Table 2 shows the detail of the time analysis of user 1, who is an expert user, his average time for the total assembly of the circuit is 1 min and 6 s for day one and 1 min and 10 s for day two, time samples were taken on two different days, on day one the sampling was done in the morning while for day two it was done in the afternoon.

When comparing these times with the times obtained from users 2 and 3, who are less experienced users, the time difference is more than 1 min difference for user 2 and more than 4 min difference for user 3. These time differences are attributable to lack of experience, and in turn provide a basis for distinguishing when a user deviates from the standard time, the reasons for which may be lack of experience and operator fatigue.

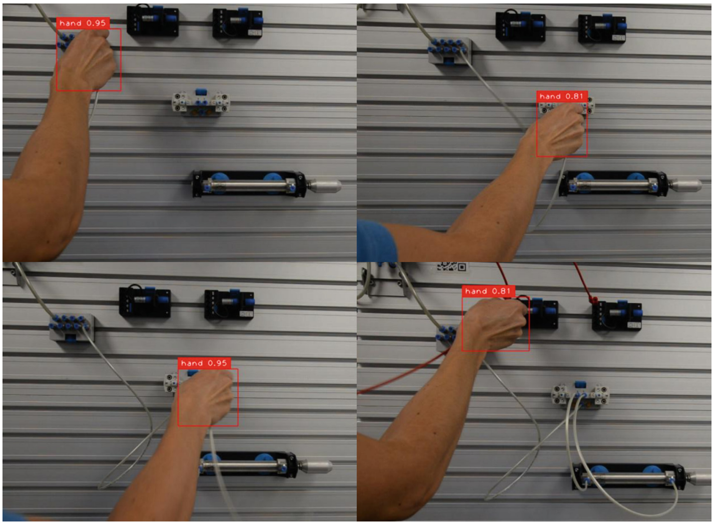


Fig. 2. Assembly operations

In general, the system use as reference the data of an expert user, in terms of execution time of the operations, and the activities to be performed by the operator. So that they can be used as a pattern to identify possible deviations both in time and in activities to be performed due to fatigue.

Table 2. Time analysis of user 2 and user 3 (s)

User 2	User 3
122,75	352,32

To train the deep learning model and be able to recognize the assembly tasks of the circuit, a segmentation of the steps to complete the assembly was performed and using YOLO the hands and their position are identified, to extract the sequence of the activities and be able to assist the operator in the correct order of the operations.

This system can recognize the person's hand and its position, which will be the inputs to identify the activity to be performed by the operator, for example: laying cable, this part will be handled by the Deep Activity Description Vector (D-ADV) system [12].

Figure 2 shows the steps necessary to complete the pneumatic circuit, by using YOLO the hands and their position are detected to track the trajectory of the hands and verify the order of the operations being executed.

4 Conclusions

This paper proposes an extension of our previous work [26] to include a performance monitoring system, supported by the evaluation of operator fatigue. This in turn has an impact on the incidence of failures and assembly time variation. Specifically, an architecture for real-time monitoring of manufacturing activities is presented.

A series of experiments for the assembly of an electro-pneumatic circuit were carried out to determine if the hypotheses proposed really generated value to the area of work design and operations research by having better performance monitoring criteria and improving decision making within a production plant. The experiment determined in the experimentation that there were variations in performance (when evaluating the information provided by the system compared to the previous definition of standard time and a manual analysis of the actions), proving that the premises do influence the work of the operators.

In future work, the ability to detect signs of fatigue in the operator's face will be included in the model, so that it will not only follow the trajectory of the hands to assist in the operations but will also detect when a process is poorly performed due to operator fatigue.

References

1. Al-Faris, M., Chiverton, J., Ndzi, D., Ahmed, A.I.: A review on computer vision-based methods for human action recognition. J. Imaging **6**(6), 46 (2020). https://doi.org/10.3390/jimaging6060046
2. Ansari, S., Naghdy, F., Du, H., Pahnwar, Y.N.: Driver mental fatigue detection based on head posture using new modified reLU-BiLSTM deep neural network. IEEE Trans. Intell. Transp. Syst. **23**(8), 10957–10969 (2021)
3. Azorin-Lopez, J., Saval-Calvo, M., Fuster-Guillo, A., Garcia-Rodriguez, J.: A novel prediction method for early recognition of global human behaviour in image sequences. Neural Process. Lett. **43**(2), 363–387 (2015). https://doi.org/10.1007/s11063-015-9412-y
4. Fernández, I.S.M., Oprea, S., Castro-Vargas, J.A., Martinez-Gonzalez, P., Garcia-Rodriguez, J.: Estimating context aware human-object interaction using deep learning-based object recognition architectures. In: Sanjurjo González, H., Pastor López, I., García Bringas, P., Quintián, H., Corchado, E. (eds.) SOCO 2021. AISC, vol. 1401, pp. 429–438. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-87869-6_41
5. Gellert, A., Sorostinean, R., Pirvu, B.C.: Robust assembly assistance using informed tree search with Markov chains. Sensors **22**(2), 495 (2022). https://doi.org/10.3390/s22020495
6. Gerekli, İ., Çelik, T.Z., Bozkurt, İ.: Industry 4.0 and smart production. TEM J. **10**(2), 799–805 (2021). https://doi.org/10.18421/TEM102-37
7. Ghasemi, Y., Jeong, H., Choi, S.H., Park, K.B., Lee, J.Y.: Deep learning-based object detection in augmented reality: a systematic review. Comput. Ind. **139**, 103661 (2022). https://doi.org/10.1016/j.compind.2022.103661
8. Hadfield, J., Koutras, P., Efthyrniou, N., Potamianos, G., Tzafestas, C.S., Maragos, P.: Object assembly guidance in child-robot interaction using RGB-D based 3D tracking. In: IEEE International Conference on Intelligent Robots and Systems, pp. 347–354 (2018)
9. Jiao, Y., Deng, Y., Luo, Y., Lu, B.L.: Driver sleepiness detection from EEG and EOG signals using GAN and LSTM networks. Neurocomputing **408**, 100–111 (2020). https://doi.org/10.1016/j.neucom.2019.05.108
10. Kazemian, A., Yuan, X., Davtalab, O., Khoshnevis, B.: Computer vision for real-time extrusion quality monitoring and control in robotic construction. Autom. Constr. **101**, 92–98 (2019). https://doi.org/10.1016/j.autcon.2019.01.022
11. Lai, Z.H., Tao, W., Leu, M.C., Yin, Z.: Smart augmented reality instructional system for mechanical assembly towards worker-centered intelligent manufacturing. J. Manuf. Syst. **55**, 69–81 (2020). https://doi.org/10.1016/j.jmsy.2020.02.010
12. Borja-Borja, L.F., Azorin-Lopez, J., Saval-Calvo, M., Fuster-Guillo, A.: Deep learning architecture for group activity recognition using description of local motions. In: International Joint Conference on Neural Networks (IJCNN) 2020, pp. 1–8 (2020). https://doi.org/10.1109/IJCNN48605.2020.9207366
13. Li, H., Wang, Y., Nan, Y.: Motion fatigue state detection based on neural networks. Comput. Intell. Neurosci. **2022**, 1–10 (2022). https://doi.org/10.1155/2022/9602631
14. Liu, M.Z., Xu, X., Hu, J., Jiang, Q.N.: Real time detection of driver fatigue based on CNN-LSTM. IET Image Proc. **16**(2), 576–595 (2022). https://doi.org/10.1049/ipr2.12373

15. Liu, Z., Peng, Y., Hu, W.: Driver fatigue detection based on deeply-learned facial expression representation. J. Vis. Commun. Image Represent. **71**, 102723 (2020). https://doi.org/10.1016/j.jvcir.2019.102723
16. Lu, Y.: The current status and developing trends of industry 4.0: a review. Inf. Syst. Front. (2021). https://doi.org/10.1007/s10796-021-10221-w
17. Lukinac, J., Mastanjević, K., Mastanjević, K., Nakov, G., Jukić, M.: Computer vision method in beer quality evaluation-a review. Beverages **5**(2), 1–21 (2019). https://doi.org/10.3390/beverages5020038
18. Pal, A., Hsieh, S.H.: Deep-learning-based visual data analytics for smart construction management. Autom. Constr. **131**(August), 103892 (2021). https://doi.org/10.1016/j.autcon.2021.103892
19. Reich, S., Teich, F., Tamosiunaite, M., Wörgötter, F., Ivanovska, T.: A data-driven approach for general visual quality control in a robotic workcell. J. Phys. Conf. Ser. **1335**(1), 012013 (2019). https://doi.org/10.1088/1742-6596/1335/1/012013
20. Riedel, A., et al.: A deep learning-based worker assistance system for error prevention. Adv. Prod. Eng. Manage. **16**(4), 393–404 (2021). https://doi.org/10.14743/apem2021.4.408
21. Ryu, J., Patil, A.K., Chakravarthi, B., Balasubramanyam, A., Park, S., Chai, Y.: Angular features-based human action recognition system for a real application with subtle unit actions. IEEE Access **10**, 9645–9657 (2022). https://doi.org/10.1109/ACCESS.2022.3144456
22. Ullah, A.S.: What is knowledge in Industry 4.0? Eng. Rep. **2**(8), 1–21 (2020). https://doi.org/10.1002/eng2.12217
23. Varol, G., Laptev, I., Schmid, C., Zisserman, A.: Synthetic humans for action recognition from unseen viewpoints. Int. J. Comput. Vision **129**(7), 2264–2287 (2021). https://doi.org/10.1007/s11263-021-01467-7
24. Villalba-Diez, J., Schmidt, D., Gevers, R., Ordieres-Meré, J., Buchwitz, M., Wellbrock, W.: Deep learning for industrial computer vision quality control in the printing industry 4.0. Sensors (Switzerland) **19**(18), 1–23 (2019). https://doi.org/10.3390/s19183987
25. Wang, P., Liu, H., Wang, L., Gao, R.X.: Deep learning-based human motion recognition for predictive context-aware human-robot collaboration. CIRP Ann. **67**(1), 17–20 (2018). https://doi.org/10.1016/j.cirp.2018.04.066
26. Zamora-Hernández, M.-A., Castro-Vargas, J. A., Azorin-Lopez, J., Garcia-Rodriguez, J.: Deep learning-based visual control assistant for assembly in Industry 4.0. Comput. Ind. **131**, 103485 (2021). https://doi.org/10.1016/j.compind.2021.103485
27. Zamora-Hernández, M.-A., Ceciliano, J.A.C., Granados, A.V., Vargas, J.A.C., Garcia-Rodriguez, J., Azorín-López, J.: Manufacturing description language for process control in industry 4.0. In: Advances in Intelligent Systems and Computing, vol. 1268, pp. 790–799. AISC (2021). https://doi.org/10.1007/978-3-030-57802-2_76

Automatic Fish Size Estimation from Uncalibrated Fish Market Images Using Computer Vision and Deep Learning

Pau Climent-Pérez(✉), Alejandro Galán-Cuenca,
Nahuel Emiliano García-d'Urso, Marcelo Saval-Calvo, Jorge Azorin-Lopez,
and Andres Fuster-Guillo

Department of Computer Technology, University of Alicante, Alicante, Spain
{pau.climent,a.galan,nahuel.garcia,msaval,jazorin,fuster}@ua.es

Abstract. Fisheries around the world show an overexploitation, which has led communities to find management strategies to tackle the problem. However, strategies are often taken on the basis of statistical data of dubious real-world utility. To address this problem, accurate biomass extraction calculations are required. The fish market is the place where vessels disembark their catches daily, and therefore a valuable point of contact to retrieve this information. Many small-sized fisheries, as stated by FAO, are a majority in some areas, and small fish markets have more difficulties installing fixed industrial cameras. This paper contributes to these efforts by proposing a complete workflow for fish size regression from uncalibrated images from a mobile camera using fish instance segmentation and classification data provided by a pretrained neural network. Ground truth fish sizes are calculated via homography, and used for comparison. The results show a mean absolute error of 1.7614±2.7633 cm using the CatBoost regressor, and even better at 1.2713±2.0616 cm when considering some calibration parameters at the input.

1 Introduction

Fisheries management is a complex process involving a variety of public and private stakeholders. Having detailed data on when and where a particular species is caught is key to improve management. Fisheries stock assessment can avoid overfishing and reduce the threat of the sustainability of the fishing industry. Bradley et al. [3] discuss the importance of accurate data and feedback to improve the effectiveness of fisheries management by ensuring that shared data access by fishermen and managers helps them to work towards a mutually agreed goal.

According to FAO [5], small-scale vessels represent over 80% of the fleet in the Mediterranean area. The Regional Plan of Action for Small-Scale Fisheries (RPOA-SSF) calls for improving knowledge to ensure adequate monitoring of SSFs, including catches. Most of these Small-Scale Fisheries (SSFs) are managed

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 319–329, 2023.
https://doi.org/10.1007/978-3-031-18050-7_31

in direct contact with local fishing authorities. D'Armengol et al. [4] underlined the interest of adopting co-management models between stakeholders, which should be economically acceptable for fishers. Palmer et al. [14] highlight the importance of having science-based evidence to assess the stock, and thus the importance of adoption of frequent and high-resolution automatic monitoring of landings.

Wholesale fish markets undertake significant effort, often manually, to count catches per species and measure fish sizes [14]. The cost of this manual labour incorporates errors per se in the obtained data, due to fatigue, human precision capacity, etc. Monitoring systems based on computer vision to automatically collect measurement data might reduce these errors, yet these are still an exception in the fishing industry. The work of Gladju et al. [11] reviews different applications, including fisheries management aspects such as catch monitoring, that makes fisheries more effective in assessing fish stock management. Recent advances in the area of deep learning and computer vision are being exploited to develop automatic species identification and carving systems based on this methodology [1,6,17].

This work focuses on the problem of automatic fish identification and measuring in Small-Scale Fisheries (SSF). These are often associated to small-scale fish markets (SSFMs). In these markets, pre-installed camera systems are not always available, which means that temporary, mobile acquisition systems are used without any calibration of the conditions. For this reason, we tackle the identification and measuring of fish species in fish markets from uncalibrated images. To address this problem, the proposal regresses the fish size in centimetres from the uncalibrated images, using an instance segmentation and species classification step carried out with the network presented in [8], and then use this information for size estimation. On the other hand, in order to train the regressors, the ground truth needs to be obtained. We use visual metrology and homography techniques to estimate it, as it is explained in Sect. 2.1.

Image classification determines the class of an image as a whole or the different image regions, i.e. detected objects, in it (table, tray, fishes), whilst instance segmentation tells apart each element per class in the image (fish 1, fish 2, fish 3...). Deep neural networks have become highly successful in classification [13] and instance segmentation [7,12].

Our proposal is meant to work in real-world SSFMs, where the pace of work is usually accelerated, and fish weighing, and processing time has stringent time constraints. Therefore, real-time instance segmentation is of paramount importance. This is achived through the use of Yolact/Yolact++ (You Only Look At Coefficients) [2], which is an improvement over the well-known YOLO architecture.

The contribution of this paper, as summarized in Fig. 1, is a system that estimates the actual size of fishes in SSFMs from uncalibrated mobile camera images. The system is composed by an instance segmentation and classification module (Sect. 2.1) that provides the necessary information (area in pixels, bounding box, and species) for the regression module (Sect. 2.3) to calculate the actual fish size in centimetres. Since the images are uncalibrated we cannot have the actual size of the fishes, hence the regressor is trained using a ground truth generator module which uses visual metrology and homography to estimate the fish size in centimetres from the original uncalibrated image, the fish tray size (which is known), and the fish size in pixels (obtained from YOLACT).

2 Fish Size Measuring Methodology

As explained, the method proposed in this paper is aimed at reducing the costly and time-consuming process of image calibration for fish size estimation in unconstrained images of fish trays in small-scale fish markets.

The workflow is as follows: The input to the proposed size regression method consists of the output of a neural network that has been pretrained for fish instance segmentation, including classification of each instance into its species (class label). This neural network is trained using human-labelled tray corner information, as well as fish silhouettes (per fish specimen, i.e. per instance). The fish instance segmentation also provides the area of the fish in pixels, as well as a bounding box in pixel coordinates.

In the proposed methodology, fish sizes are estimated by a regressor. This is a supervised method that requires ground truth to be available (i.e. in this case fish sices). However, since there are thousands of trays, with several thousand fish specimens that need to be measured physically for ground truth labelling, the task becomes unfeasible. To avoid manual labelling of fish sizes, an automatic ground truth generator using visual metrology (homography estimation, more specifically), is employed instead. That is, knowing the size of real objects in the scene (such as the tray the fish are presented in during the auction, in this case), an image transformation function can be calculated so that the fish specimens are also be converted, and therefore, the bounding boxes obtained by the neural network element can be converted from pixels to centimetres too. This information is then used to guide the regressor during training.

The main contribution of this paper is therefore the accurate regression of fish sizes from segmentation data inferred by a neural network, and using homography-estimated fish sizes as ground truth for comparison during training. Figure 1 shows an overview of the whole workflow for the proposed methodology, including the different components it consists of. Please note the upper part of the schematic represents the size prediction using instance segmentation and subsequent regression. The lower part, i.e. visual metrology, is used to generate ground truth that will be used only during the training stage. For the final fish size estimation (yellow box in Fig. 1), different regressors are explored, from well-established ones such as support vector machines (SVMs), to some other alternatives such as gradient boost regressor (GBR), etc. This will be explained with further detail in Sect. 2.3. All regressor methods used are fed the same input, i.e. the bounding box and segmentation mask of the image with fishes in trays, along with the fish class (that can act as a prior, given the variability of fish sizes depending on fish species). Using both segmentation and bounding box simplifies the estimation of fish lengths, because using bounding boxes

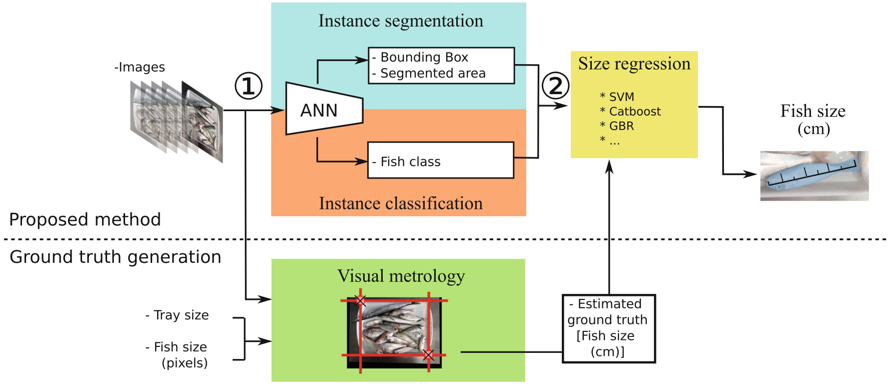


Fig. 1. General workflow of the proposed methodology for fish size estimation. The upper part shows the workflow from the input images to the estimated fish sizes. The workflow consists of two blocks: Instance segmentation and classification (in blue and orange, from a pretrained neural network), and the size regression presented in this paper (yellow). The bottom part shows a visual metrology block (green), i.e. the ground truth generation process, which is used on the upper workflow during training.

only ignores the actual value of area of the fish in the box, and only using the mask does not provide an actual perception of the fish aspect ratio (length vs height). The instance segmentation and classification (blue and orange boxes) is performed using a YOLACT network that is briefly explained in Sect. 2.1.

2.1 Inference of Input Data: Instance Segmentation and Classification

Starting from an image showing several fish on a tray (denoted as ① in Fig. 1), the data is passed on to a pretrained neural network. In the proposed method, a YOLACT architecture is pretrained for fish instance segmentation as described in [8]. The outputs of this network (denoted as ② in Fig. 1) are then used by the uncalibrated fish size regression method presented in this paper. These outputs, consisting of instance segmentation masks, bounding boxes, and species labels, become the input data to the regressor (yellow box in Fig. 1), which provides the final real-world fish sizes.

However, in the DeepFish and DeepFish 2 projects (see http://deepfish.dtic.ua.es) biomass estimation (i.e. fish volume and total weight of catches) needs to be considered too. Therefore, one approach to achieve this is to estimate the size of the fish, and use this to estimate the biomass given the existing correlation between fish size (length) and weight. These correlations are expert knowledge in the field of marine biology, and are out of the scope of this paper. Also, from this point of view of biomass estimation, which is one of the main end goals, fish size as an area (in cm^2) might be of interest. This drives the present work to focus on fish size estimation, as a useful cue for biomass estimation, given the existing, well-known correlation.

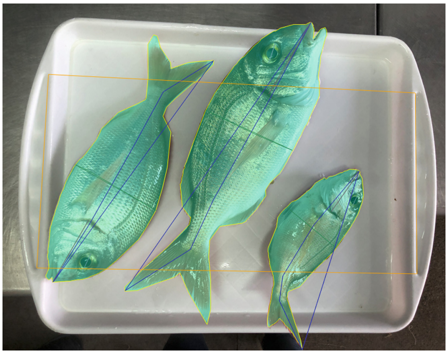


Fig. 2. Human-labelled image, showing the ground truth for the input data used by the size regression block in the presented workflow.

2.2 Ground Truth Size Estimation via Visual Metrology

At the time of acquiring the size ground truth from the dataset, four important aspects have been considered:

- Knowledge of the **real dimensions of the tray**. The objects of the image can be scaled with measures of the real world. The width and height of each tray type is known.
- **Physical points of the tray.** The measures among them are known in advance in centimetres. These are marked as a yellow rectangle in the ground truth labelling tool (see Fig. 2).
- **Segments of the fish size**. In the ground truth labelling framework, a polygon representing the size of the fish from its mouth to its tail is assigned to each specimen (taking into account the possible curvature of the spine). It is represented as a blue polygon (see Fig. 2).
- Use of **homography** estimation techniques to reproject the image onto a plane, and therefore reduce errors due to arbitrary rotations due to the acquisition of the images, that vary in angle and distance, since most were taken using a mobile phone camera.

With these considerations, the sizes used for ground truth for the regression method can be calculated with an average error of **2.72%** compared to the actual sizes of a subset of fish specimens measured by experts using a fish measuring device (i.e. an *ichthyometer*).

2.3 Fish Size Regression

With all the previous data acquired and inferred (① and ②), the size of fish specimens can be regressed. This will involve calculating a final output (Fig. 1, right-most) consisting of a size in centimetres for every set of segmentation mask, bounding box, and fish species label.

Independently of the regression method used, some common considerations were taken regarding the data in order to improve the results:

- **Conversion of segmentation**. The instance segmentation is extracted as a set of coordinates (x, y), which is not appropriate for a regression model to use as an input because of the large (and varying) number of points. The solution used is converting this polygon to a single number, calculating the enclosed area.
- **Transformation of bounding box**. Most of the images in the dataset will have different angles and distance from the tray, the same bounding box in pixels from other point of view could have nothing in common in terms of real-world size. To avoid that situation, the initial point (x, y) and the width and height have been used, instead of the initial point (x, y) and final point (x, y).

3 Experimental Evaluation

Dataset Description: To carry out the experiments set out in this paper, the DeepFish dataset has been used. It consists of 1,291 images of trays of fish (7,339 specimens) from a small-scale fish market in el Campello (Alicante, Spain). Images were labelled at the pixel level (per fish instance, i.e. each specimen). There are fishes from 59 different species, with 60 class labels provided (due to sexual dimorphism of one species). For further details, please refer to the public dataset repository, and related paper [9].

Experimental Setup: The objective of the proposed experiments is to select the most appropriate regression model for the fish size estimation, including learning the image calibration during training from the expected output results. Five different experiments have been envisaged:

1. Select a subset of best-performing regressors.
2. Reduce the selection further by checking hyperparameter tuning.
3. Select the best-performing algorithm, when adding normalization.
4. Verify the best results using 10 k-fold loops.
5. Compare the results to those with additional tray corner (calibration) information.

The regression models used for these experiments which have a better fit to the data are **GBR** [18], **Extra Trees** [10] and **CatBoost** [15]. However, the presence of **SVM** [16] with radial kernel, i.e. the best kernel for this situation, is used to compare a standard, average and common regressor with the output of the others. Moreover, SVM with lineal kernel has been compared too, for better perspective of a model with lower but correct performance.

4 Results and Discussion

Battery of Regressors: On the first experiment, the performance of 25 different models is analysed over the dataset.

Analysis of Hyperparameters: From the previous results, the six best-performing methods have been selected to check their accuracy on the data when parameter tuning is used. As stated, the SVM model has also been included as a baseline, for comparison. Results can be seen in Table 1.

Table 1. Comparison between results of the best six regression models considered (and SVM), without (**left**), and with parameter tuning (**right**).

Regressor	Without parameter tuning				With parameter tuning			
	MAE (cm)	MSE	R^2	MAPE	MAE (cm)	MSE	R^2	MAPE
Extra Trees	1.8613	8.7115	0.7694	0.1173	1.8108	8.8154	0.769	0.1152
GBR	1.8504	9.3102	0.7544	0.1166	1.8339	8.7386	0.7705	0.116
CatBoost	1.8506	8.8161	0.7668	0.1172	**1.8033**	**8.6005**	**0.7742**	**0.1144**
LightBM	1.9224	9.5624	0.7471	0.1201	1.8780	8.9229	0.7649	0.1188
Random Forest	1.8830	9.5934	0.7474	0.1175	1.8329	9.0251	0.7645	0.1161
XGBoost	1.9853	9.8369	0.7409	0.1252	1.8452	8.8572	0.7662	0.1158
SVM	2.0654	157.179	0.6136	0.1311	1.9343	10.1436	0.736	0.1244

Level of Normalization: Given the importance of data normalization, the next step has consisted in selecting the appropriate normalization for the data. The results of this step are reflected in Table 2. At this moment, comparing only the tree best models previously mentioned, and SVM as a reference model. Intuitively, normalized data works better with regression algorithms.

The reason behind the better performance of MinMax over the standard algorithm could be caused by the characteristics of some outliers. Commonly, outliers are caused due to errors on the measurement or uncommon instances. However, in this dataset, there are some species like *Sphyraena sphyraena*, which has a presence of 124 instances, i.e. 2% of the total number of instances, with values oscillating from 25 cm to 83 cm, with an average of 45 ± 12.62 cm. Meanwhile, the dataset contains measures from 5 cm to 83 cm, with an average of 17 ± 6.91 cm. This means every instance of *Sphyraena sphyraena* is an outlier inside the global data, given that it is bigger than most of the other fish species considered. In opposition to other algorithms, MinMax does not change the shape of the distribution, preventing reduction in weight or importance of outlier instances in the model.

Best Performance Results: Once the best-performing models have been identified, an additional set of results (Table 3, left) has been obtained, in which the input data is scaled using MinMax, and a 10 *k*-fold validation has been carried

Table 2. Comparative of MAE in centimetres between the best regression models analysed and different normalization of the data input and output.

Regression model	No scaling	Standard scaling input	MinMax scaling input	MinMax scaling I/O
GBR *k*-fold	1.8564	1.8564	**1.8539**	1.854
Extra Trees *k*-fold	2.0052	1.9969	**1.9857**	2.0119
SVM Tuned *k*-fold	4.3581	1.8471	**1.8195**	21.5391
CatBoost *k*-fold	1.7954	1.7710	**1.7920**	1.7824

out. The metrics shown are the mean from a loop of 10 different 10 *k*-fold seeds, to avoid possible situational errors due to causality. SVM with a linear kernel (i.e. "SVM Linear") has also been included to get a better perspective of the improvement in performance shown by other models. In fact, the results for the algorithms with MinMax scaling are different in this section due to a more stringent hyperparameter tuning, at the expense of longer running times, i.e. more computationally expensive.

Table 3. Final results with the best regression models analysed with the 3 original inputs (bounding box in pixels, segmentation area in pixels, species label), **left**; and adding calibration inputs, i.e. the 4 squares markers of the tray (x, y), **right**.

Regression model	Without calibration		Including calibration	
	MAE (cm)	R^2	MAE (cm)	R^2
GBR 10 *k*-fold	1.8501 ± 3.0099	0.7613	1.3304 ± 2.0937	0.8740
Extra Trees 10 *k*-fold	1.9715 ± 3.0396	0.7462	1.4098 ± 2.3239	0.8531
SVM Lineal 10 *k*-fold	2.6711 ± 4.4582	0.4746	1.8234 ± 3.3598	0.6996
SVM Radial 10 *k*-fold	1.8741 ± 3.1885	0.7307	1.2994 ± 2.2449	0.8620
CatBoost 10 *k*-fold	$\mathbf{1.7614 \pm 2.7633}$	**0.7926**	$\mathbf{1.2713 \pm 2.0616}$	**0.8840**

Addition of Calibration Parameters as Input: Finally, the last experiment consists in repeating the previous evaluation with the best models, but now using additional input parameters, more precisely, the coordinates in pixels of the tray corners, which are used to find the ground truth of the output parameter, i.e. the size, with visual metrology techniques. These four $2D$ points (x, y) help the regressor better understand the data and reduce errors in the bounding boxes and segmentation areas obtained that are caused by the perspective, the distortion, and angle variations. As mentioned before, this configuration works better, however, it will not be possible to get these tray reference points in real-time during runtime of the actual system, since these parameters have been labelled manually, and are acquired from the ground truth data in the experiments. As shown in the right column on Table 3, this information helps improve the results, and is therefore useful in any scenarios where it is possible to gather these points automatically.

5 Conclusions

The main contribution of this article is the proposal of a method based on a full workflow to estimate fish sizes in small-scale fisheries markets. The input to the regressor to get the size of fish specimens is the information extracted from an instance segmentation neural network. Visual metrology is used as a ground truth fish size estimation, that is used to direct the training of the fish size regressor, that uses uncalibrated data. This paper has introduced a novel method for size prediction of fish specimens from pixel-based measurements as input obtained directly from uncalibrated images. This avoids the use of visual metrology (used here only as ground truth), that requires camera calibration or at least landmark (corner) detection of a known real-world object. In some situations the installation of fixed fully calibrated cameras is unfeasible, as is the case in some small-scale fish markets (SSFMs) that lack the space or resources, and in which pictures from uncalibrated mobile cameras taken from diverse angles and distances is much more practical.

Using the DeepFish dataset for validation, it has been proven that the most suitable model in terms of generalization capabilities and minimum error is the CatBoost regressor, specially for the case in which the input is normalized with the MinMax approach and using hyperparameter tuning. We have shown that results can further be improved when using tray corner points as inputs to the regressor, as this information can be useful to correct the segmentation areas and bounding box inputs from the deep neural network, which produced uncalibrated results. However, since the proposed system relies mainly on uncalibrated images, only the ground truth (training) pictures had these markers available; that is, new trays captured by the system in real-time actually lack these references. An automated tray corner location model would therefore be helpful. In future works, which will include larger-scale fish markets, it is planned to migrate from uncalibrated and mobile to calibrated and fixed cameras, since it does seem to help increase performance from 1.7614 ± 2.7633 to 1.2713 ± 2.0616. Furthermore, regression from fish silhouette coordinates, and volume estimation are also left as future work. To the best of our knowledge, there is no study in the literature about regression techniques to calculate size in uncalibrated images, using visual metrology with the support of homography to estimate ground truth, applied to the fishing sector. This research will be useful to the industry 4.0 applied to small-scale fisheries and small or traditional fish markets, or even other fields in which regression is needed, where installation of fixed cameras and their calibration for size measuring is not a possibility.

Acknowledgements. This work was developed with the collaboration of the Biodiversity Foundation (Spanish Ministry for the Ecological Transition and the Demographic Challenge), through the Pleamar Programme, co-financed by the European Maritime and Fisheries Fund (EMFF). Deepfish/Deepfish 2 projects.

References

1. Álvarez-Ellacuría, A., Palmer, M., Catalán, I.A., Lisani, J.L.: Image-based, unsupervised estimation of fish size from commercial landings using deep learning. ICES J. Mar. Sci. **77**(4), 1330–1339 (2020). https://doi.org/10.1093/icesjms/fsz216
2. Bolya, D., Zhou, C., Xiao, F., Lee, Y.J.: Yolact++ better real-time instance segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **44**(2), 1108–1121 (2022). https://doi.org/10.1109/TPAMI.2020.3014297
3. Bradley, D., Merrifield, M., Miller, K.M., Lomonico, S., Wilson, J.R., Gleason, M.G.: Opportunities to improve fisheries management through innovative technology and advanced data systems. Fish Fish. **20**(3), 564–583 (2019). https://doi.org/10.1111/faf.12361
4. d'Armengol, L., Prieto Castillo, M., Ruiz-Mallén, I., Corbera, E.: A systematic review of co-managed small-scale fisheries: Social diversity and adaptive management improve outcomes. Glob. Environ. Chang. **52**, 212–225 (2018). https://doi.org/10.1016/j.gloenvcha.2018.07.009
5. FAO: The State of Mediterranean and Black Sea Fisheries 2020. FAO (2020). https://doi.org/10.4060/cb2429en
6. French, G., et al.: Deep neural networks for analysis of fisheries surveillance video and automated monitoring of fish discards. ICES J. Mar. Sci. **77**(4), 1340–1353 (2019). https://doi.org/10.1093/icesjms/fsz149
7. Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Martinez-Gonzalez, P., Garcia-Rodriguez, J.: A survey on deep learning techniques for image and video semantic segmentation. Appl. Soft Comput. **70**, 41–65 (2018). https://doi.org/10.1016/j.asoc.2018.05.018
8. García-d'Urso, N.E., Galán-Cuenca, A., Climent-Pérez, P., Saval-Calvo, M., Azorin-Lopez, J., Fuster-Guillo, A.: Efficient instance segmentation using deep learning for species identification in fish markets. In: International Joint Conference on Neural Networks (IJCNN) (accepted) (2022)
9. García-d'Urso, N.E., et al.: The deepfish computer vision dataset for fish instance segmentation, classification, and size estimation. Sci. Data **9**(1), 287 (2022)
10. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Mach. Learn. **63**(1), 3–42 (2006). https://doi.org/10.1007/s10994-006-6226-1
11. Gladju, J., Kamalam, B.S., Kanagaraj, A.: Applications of data mining and machine learning framework in aquaculture and fisheries: A review. Smart Agricult. Technol. **2**, 100061 (2022). https://doi.org/10.1016/j.atech.2022.100061
12. Hafiz, A.M., Bhat, G.M.: A survey on instance segmentation: state of the art. Int. J. Multimedia Inf. Retrieval **9**(3), 171–189 (2020). https://doi.org/10.1007/s13735-020-00195-x
13. Minaee, S., Boykov, Y.Y., Porikli, F., Plaza, A.J., Kehtarnavaz, N., Terzopoulos, D.: Image segmentation using deep learning: a survey. IEEE Trans. Pattern Anal. Mach. Intell. 1 (2021). https://doi.org/10.1109/TPAMI.2021.3059968

14. Palmer, M., Álvarez Ellacuría, A., Moltó, V., Catalán, I.A.: Automatic, operational, high-resolution monitoring of fish length and catch numbers from landings using deep learning. Fish. Res. **246**, 106166 (2022). https://doi.org/10.1016/j.fishres.2021.106166
15. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A.: Catboost: unbiased boosting with categorical features. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 31. Curran Associates, Inc. (2018)
16. Suthaharan, S.: Support vector machine. In: Machine Learning Models and Algorithms for Big Data Classification. ISIS, vol. 36, pp. 207–235. Springer, Boston, MA (2016). https://doi.org/10.1007/978-1-4899-7641-3_9
17. Vilas, C., et al.: Use of computer vision onboard fishing vessels to quantify catches: the iobserver. Mar. Policy **116**, 103714 (2020). https://doi.org/10.1016/j.marpol.2019.103714
18. Zemel, R., Pitassi, T.: A gradient-based boosting algorithm for regression problems. In: Leen, T., Dietterich, T., Tresp, V. (eds.) Advances in Neural Information Processing Systems, vol. 13. MIT Press (2000)

Vehicle Overtaking Hazard Detection over Onboard Cameras Using Deep Convolutional Networks

Jorge García-González[1,2](✉), Iván García-Aguilar[1,2], Daniel Medina[3], Rafael Marcos Luque-Baena[1,2], Ezequiel López-Rubio[1,2], and Enrique Domínguez[1,2]

[1] Department of Computer Languages and Computer Science, University of Málaga, Bulevar Louis Pasteur, 35, 29071 Málaga, Spain
{jorgegarcia,ivangarcia,rmluque,ezeqlr,enriqued}@lcc.uma.es

[2] Biomedical Research Institute of Málaga (IBIMA), C/ Doctor Miguel Díaz Recio, 28, 29010 Málaga, Spain

[3] Institute of Communications and Navigation, DLR (Deutsches Zentrum fr Luft- und Raumfahrt e.V.), DLR. Kalkhorstweg 53, 17235 Neustrelitz, Germany
daniel.ariasmedina@dlr.de

Abstract. The development of artificial vision systems to support driving has been of great interest in recent years, especially after new learning models based on deep learning. In this work, a framework is proposed for detecting road speed anomalies, taking as reference the driving vehicle. The objective is to warn the driver in real-time that a vehicle is overtaking dangerously to prevent a possible accident. Thus, taking the information captured by the rear camera integrated into the vehicle, the system will automatically determine if the overtaking that other vehicles make is considered abnormal or dangerous or is considered normal. Deep learning-based object detection techniques will be used to detect the vehicles in the road image. Each detected vehicle will be tracked over time, and its trajectory will be analyzed to determine the approach speed. Finally, statistical regression techniques will estimate the degree of anomaly or hazard of said overtaking as a preventive measure. This proposal has been tested with a significant set of actual road sequences in different lighting conditions with very satisfactory results.

1 Introduction

Nowadays, data plays a critical role in just about every aspect of our lives. Therefore, it can be helpful to analyze and make use of it to improve each discipline. One of the most common causes of injury and death worldwide is traffic accidents because they are uncertain and can occur in any place and at any time.

Numerous systems have been developed that are either part of the road or installed in the vehicle itself to reduce the number of accidents. Advances included in the last group are, for instance, collision avoidance, lane detection,

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 330–339, 2023.
https://doi.org/10.1007/978-3-031-18050-7_32

lane departure systems, parking assistants, etc. These systems are mainly composed of sensors and cameras located in strategic areas of the vehicle. Visual data contains rich information compared to other information sources. Thus, it can play a vital role in detecting accidents, traffic jams, and other anomalies. The data collected by these systems is processed using computer vision algorithms and machine learning. When a potential driving risk is detected, the time available to react to it is usually very short. Therefore, the effectiveness of collision avoidance systems depends directly on the time required to identify the anomaly. If a short amount of time is taken for possible crash detection, then a longer amount of time is available for driver warning or evasive actions to be performed. Applied to the field of object detection with convolutional neural networks, there is a multitude of pre-trained models available, which can be classified according to the time required. The first group follows the classical flow, based on setting up a series of candidate regions and then inferring each of them. Models such as EfficientNet [11] or CenterNet [14] stand out among others. However, this type of model is not feasible for real-time detection due to the high computational time required. The second group proposes detection algorithms focused on minimizing computing time and thus speeding up detection. Models such as Single Shot MultiBox Detector (SSD) [7], Faster R-CNN [10], or Yolo [1] can be highlighted. In the field of vehicle detection and autonomous driving, Convolutional Neural Networks (CNNs) have accomplished huge outcomes. Traffic analysis has been a recurring topic of interest for researchers in recent years. Numerous studies focused on detecting vehicles [3,8,9] or tracking and counting [4,13] promoting the development of new proposals in this field.

Finding traffic anomalies is a challenge due to the subjectivity of their definition. Based on the detection of anomalies in static highway cameras, *Earnest et al.* propose a framework for achieving a high Detection rate on general road-traffic surveillance footage [5]. The framework is based on local features such as trajectory intersection, velocity calculation, and their anomalies. It can detect accidents correctly with 71% detection rate on the accident videos obtained under various ambient conditions. According to the environmental conditions of the surroundings, *Wang et al.* sets up a framework for detecting different types of accidents in a mixed traffic environment with low-visibility conditions [12]. First, an image enhancement algorithm known as Retinex is applied. Next, a previously trained Yolov3 is applied to detect different situations including fallen pedestrians or vehicle rollovers. Finally, a set of features was developed from the Yolo results, based on which a decision model for crash detection was trained. In the field of anomaly detection in cameras installed in the car, we find works such as the one proposed by *Trung-Nghia et al.* who design an accident detection network called Attention R-CNN [6]. This network consists of two streams. The object characteristic stream employs the attention mechanism that exploits local and global contextual levels to recognize the object characteristic property. *Choi et al.* propose a car crash detection system, based on ensemble deep learning and multimodal data from dashboard cameras [2]. Deep learning techniques, gated recurrent unit (GRU), and convolutional neural network (CNN) are used to develop a car crash detection system. In addition, a weighted average ensemble set is used as an ensemble technique.

Our proposal is focused on detecting road speed anomalies, taking as reference the vehicle that is driving. Firstly, the information captured by the cameras installed in the car is processed to detect vehicles on the road. Each detected vehicle will be tracked over time, and its trajectory will be analyzed to determine the approach speed. Subsequently, statistical regression techniques are applied to determine whether or not an anomaly exists.

The following sections are structured as follows: Sect. 2 shows the proposed methodology, Sect. 3 on page 3 explains the experiments supporting our proposal and Sect. 4 on page 4 explains our conclusions.

2 Methodology

In this section the proposed methodology for the detection of dangerously approaching vehicles is detailed. First of all, an object detection deep neural network must be applied to each incoming video frame, in order to generate a list of vehicle detections given by their bounding boxes. After that, an object tracking algorithm must be employed to obtain vehicle trajectories. Each trajectory is a list of bounding boxes associated to the same vehicle in successive video frames, allowing for intermediate frames where the vehicle has not been detected.

Let t be the time index within the video sequence. The angular diameter δ (in radians) of an object, also called apparent diameter, is given by:

$$\delta = 2 \arctan \frac{d}{2D} \tag{1}$$

where d is the actual diameter of the object (in meters) and D is the distance from the camera to the object (also in meters). If $D \gg d$, then we can apply the small angle approximation $\alpha \approx \arctan \alpha$ to obtain:

$$\delta = \frac{d}{D} \tag{2}$$

We have experimentally found that the small angle approximation yields good results for approaching vehicles. If we further assume that the object moves at a constant speed v relative to the camera, then (1) can be rewritten as:

$$\delta = \frac{d}{e_0 + vt} \tag{3}$$

where e_0 is the distance of the object at time $t = 0$. Then we can invert (3) to obtain:

$$\frac{1}{\delta} = \frac{e_0 + vt}{d} \tag{4}$$

Since d and e_0 are constants for each vehicle, (4) means that the inverse of the apparent diameter $\frac{1}{\delta}$ has a linear dependence with respect to time t. For each time instant that a vehicle is detected, its apparent diameter δ can be approximated as the square root of the area (in pixels) of the bounding box associated to the vehicle at that time instant. This allows computing the slope of the linear function of (4), which is the relative velocity v of the vehicle with respect to the camera, by linear regression. Please note that a sample for the linear regression is obtained for each video frame where the vehicle is visible. In order to enhance the estimation of v, the RANSAC (RANdom Sampling And Consensus) algorithm is employed to filter out the outlying measurements of $\frac{1}{\delta}$ within the linear regression. Finally, a threshold on v is defined so that approaching vehicles that have a excessively large velocity are detected as dangerous.

3 Experiments

3.1 Implementation Details

Figure 1 shows a schematic of our proposal. All the implementation of the method has been done using python[1] for general-purpose programming and pytorch[2] for the use of artificial neural networks. To ignore spurious detections, all trace detections with less than 30 elements are ignored. Likewise, to detect the relative velocity of a vehicle at time t, the detections of that vehicle from time $t - 15$ are used. The RANSAC application has been performed using the Scikit-Learn[3] library with 1 as the residual threshold value. 7 is used as a speed threshold to mark an overtaking as dangerous.

3.2 Data

Due to the lack of a specific dataset for this problem, a set of videos have been obtained using a camera installed on the rear of a car. The dataset contains 4 150-s videos with shape 1920 × 1090, 30 frames per second, and 23 overtaking vehicles (both car and trucks). Overtakings have been manually annotated with their time window and labeled as dangerous or safe based on the relative speed of the recording car (Fig. 2).

[1] https://www.python.org/.

[2] https://pytorch.org/.

[3] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RANSACRegressor.html.

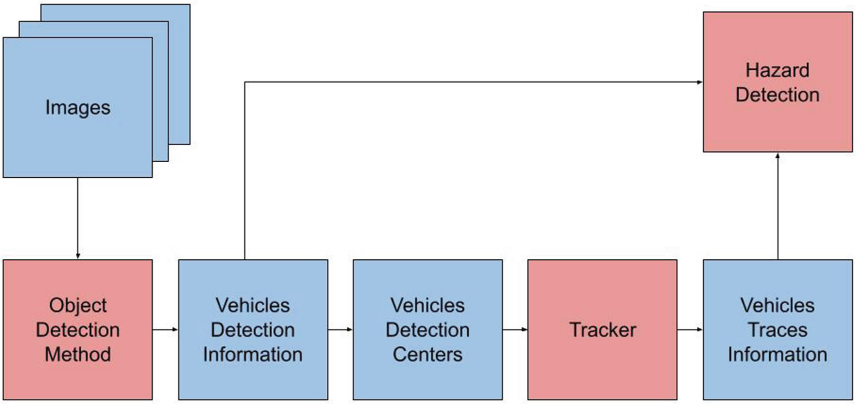

Images
Object Detection Method
Vehicles Detection Information
Vehicles Detection Centers
Tracker
Vehicles Traces Information
Hazard Detection

Fig. 1. Proposal scheme. Blue squares represent information while red squares represent sub-methods. Images from the sequence are provided to object detection method to get classes, bounding boxes, and confidences. Vehicles centers are given to the tracker to obtain a relation between vehicles appearing in different images. Both trace information and bounding boxes are then given to the proposed hazard detection method.

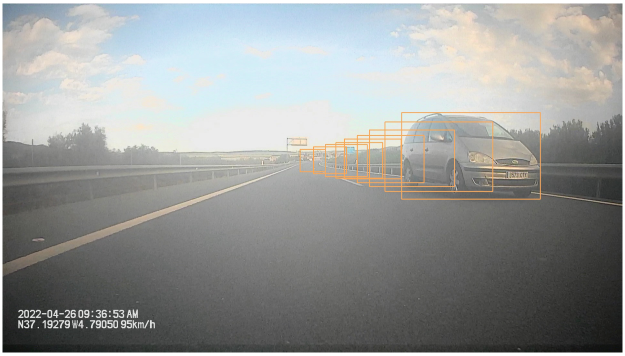
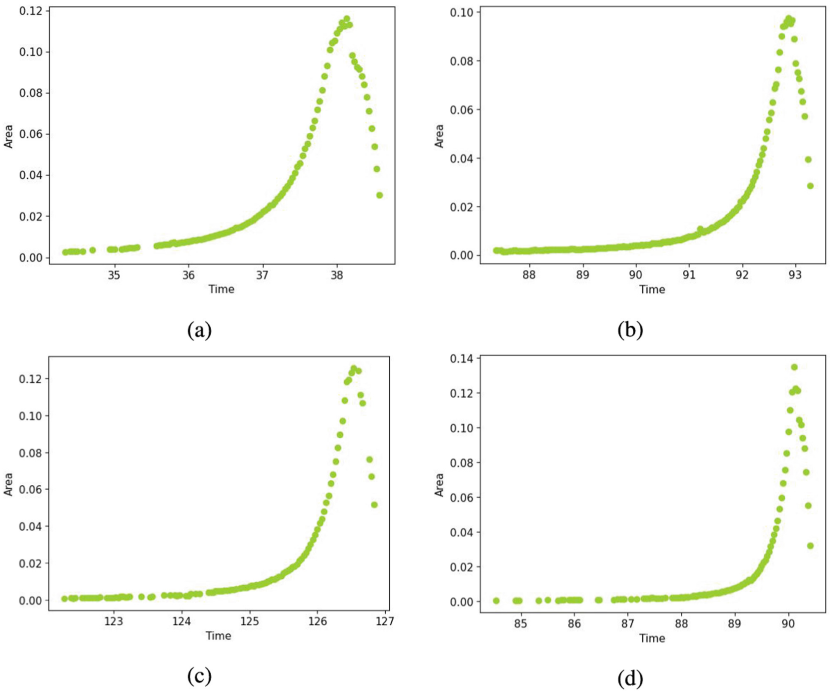


Fig. 2. Object detection bounding box evolution for a tracked overtaking car. Only one in ten bounding box is printed to allow a better readability.



(a) (b)

(c) (d)

Fig. 3. Ratio of vehicle bounding box to the total area of the image for four different overtaking vehicles.

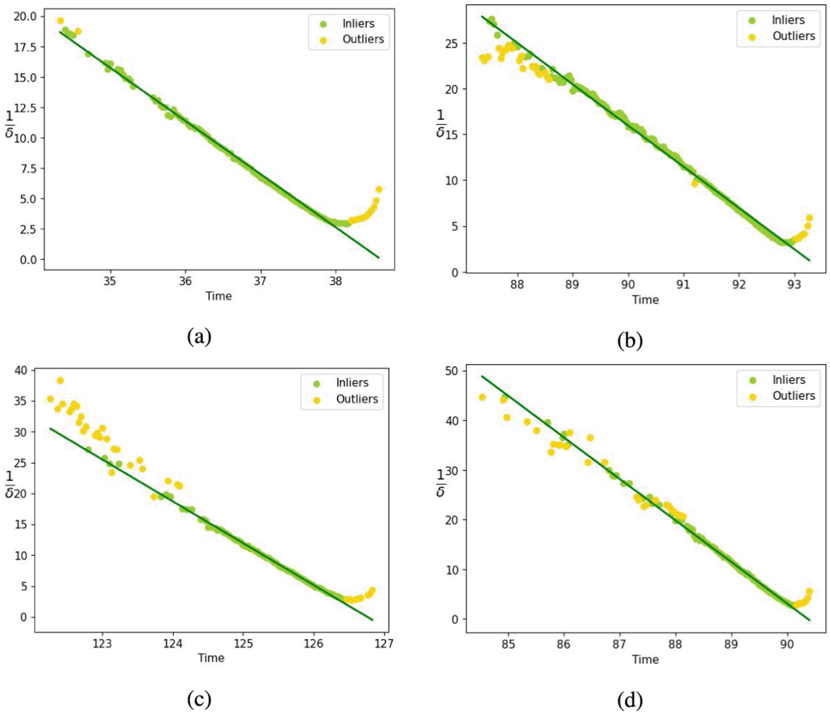
3.2.1 Tracker and Object Detection

As a tracker algorithm, incoming frames bounding box centers are related to the last tracked objects using a linear sum assignment[4] to get the minimum weight matching in bipartite graphs. Our tracker also contemplates occasional disappearances of objects due to object detection failures.

In order to perform the object detection task, YOLO v5 [1] model has been applied. YOLO is a convolutional layer-based method using a single detection pass to split the image into N regions with size $S \times S$ to contain objects and classes proposals later unified using Non Maximum Suppression. The well known *ultralytics*[5] implementation of this model has been used with its pre-trained weights.

[4] https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.linear_sum_assignment.html.

[5] https://github.com/ultralytics/yolov5.



(a)

(b)

(c)

(d)

Fig. 4. Figure 3 data converted to lines and RANSAC effect. Green dots are points taken into account by RANSAC, yellow ones are points taken as outliers by RANSAC algorithm and not used to get the linear regression. Green line is the predicted line by the RANSAC regressor for the same time.

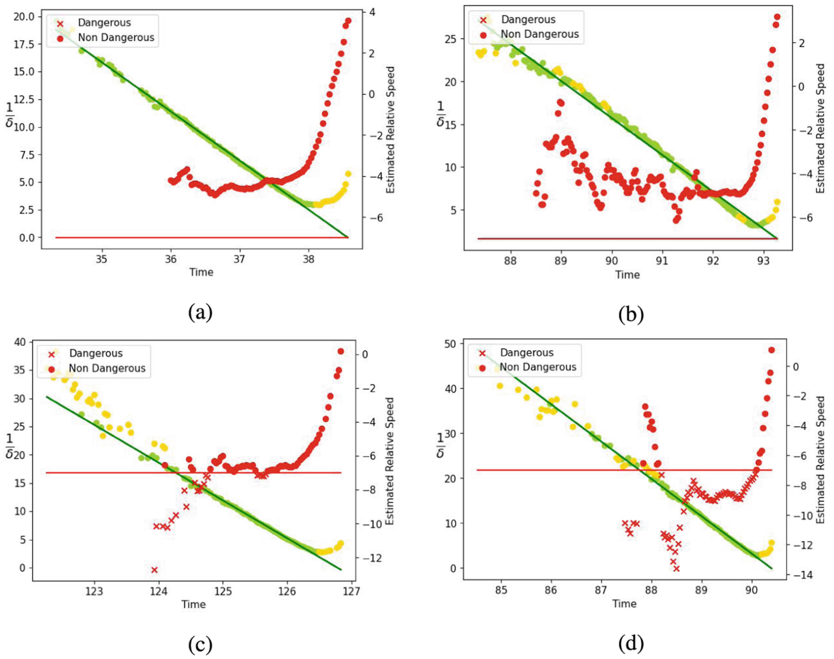
3.3 Evaluation

In order to study our method performance, two well known metrics as accuracy and precision have been selected.

$$ACC = \frac{TP + TN}{TP + FN + TN + FP} \tag{5}$$

$$PPV = \frac{TP}{TP + FP} \tag{6}$$

with TP (True Positive) as correct positive identification, TN (True Negative) as correct negative identification, FP (False Positive) as type I error, and FN (False Negative) as type II error. Accuracy provides an estimate of how close our predictions are to previously observed data while precision indicates the reliability of the system in identifying dangerous situations. Given a vehicle trace, if its estimated speed has been designated as hazardous at some point in the trace, the approach of that vehicle is considered hazardous.



(a)

(b)

(c)

(d)

Fig. 5. Figure 4 with the danger labels assigned to each time t by our method. Red lines are thresholds to consider a relative speed as dangerous. Red dots are non dangerous speeds while red X's are dangerous detected speeds. Two are approximations considered hazardous, and two are approximations considered non-hazardous. Red lines, red dots and red X's are to be measured with the right y-axis.

3.4 Results

Figure 3 shows the ratio of the vehicle bounding box to the total area of the image for four different overtaking vehicles. The bounding box area increases while the overtaking is starting and then drops as the vehicle is coming out of the camera angle. Figure 4 shows the data for same four vehicles converted to a line using equations described in Sect. 2. As can be observed, as the vehicle approaches the data is more stable and continuous until the vehicle begins to leave the visible region. The same figure also shows the effect of RANSAC on the data and its usefulness in preventing the slope from being affected by outliers.

Figure 5 shows the same data as Fig. 4 with the hazard output. As can be seen, even with RANSAC, older data tend to be less stable.

Table 1. Quantitative results. The data shown in the first four columns are the sum of the results for the four videos studied because the accuracy and precision are calculated on the complete dataset.

TP	TN	FP	FN	ACC	PPV
9	11	3	0	0.8695	0.75

Table 1 shows a summary of our results. The hyperparameter adjustment has taken into account the preference of type I errors over type II errors given the application of the system. This system is initially conceived as a possible aid to the driver, not as a substitute for him, so it would always be supervised and in the event of a hypothetical overtaking indicated as dangerous, the driver would be in charge of verifying and acting accordingly. This approach aligns with the absence of False Negatives, but the presence of false positives, decreasing precision but maintaining a better accuracy as shown in Fig. 1.

4 Conclusions

In this paper, a method to support driving by detecting hazards during vehicle overtaking has been proposed. The proposal takes advantage of the evolution of the bounding boxes provided by an object detection method based on convolutional neural networks to estimate the relative speed of another vehicle overtaking the recording vehicle. To test it, a set of overtaking data has been collected and manually annotated. Based on the results obtained, the proposal is considered promising as a possible future driver assistance system that will allow the driver to act in the prevention of a possible accident. As future work, a larger data set of overtaking data should be used to study and correct the weaknesses of the method.

Acknowledgment. This work is partially supported by the Ministry of Economy and Competitiveness of Spain under grants TIN2016-75097-P and PPIT.UMA.B1.2017. It is also partially supported by the Ministry of Science, Innovation and Universities of Spain under grant RTI2018-094645-B-I00, project name Automated detection with low-cost hardware of unusual activities in video sequences and by the Autonomous Government of Andalusia (Spain) under project UMA18-FEDERJA-084, project name Detection of anomalous behavior agents by deep learning in low-cost video surveillance intelligent systems. All of them include funds from the European Regional Development Fund (ERDF). The work has been also supported by the University of Málaga through its Research Plan (Plan Propio de Investigación UMA). The authors thankfully acknowledge the computer resources, technical expertise and assistance provided by the SCBI (Supercomputing and Bioinformatics) center of the University of Málaga. They also gratefully acknowledge the support of NVIDIA Corporation with the donation of two Titan X GPUs used for this research. Iván García-Aguilar is funded by a scholarship from the Autonomous Government of Andalusia (Spain) under the Young Employment operative program [grant number SNGJ5Y6-15]. The authors acknowledge the funding from the Universidad de Málaga.

References

1. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection (2020). https://arxiv.org/abs/2004.10934
2. Choi, J.G., Kong, C.W., Kim, G., Lim, S.: Car crash detection using ensemble deep learning and multimodal data from dashboard cameras. Expert Syst. Appl. **183**(C) (2021). https://doi.org/10.1016/j.eswa.2021.115400
3. García-Aguilar, I., Luque-Baena, R.M., López-Rubio, E.: Improved detection of small objects in road network sequences using CNN and super resolution. Exp. Syst. **39**(2) (2021). https://doi.org/10.1111/exsy.12930
4. Gomaa, A., Minematsu, T., Abdelwahab, M.M., Abo-Zahhad, M., ichiro Taniguchi, R.: Faster CNN-based vehicle detection and counting strategy for fixed camera scenes. Multimedia Tools Appl. (2022). https://doi.org/10.1007/s11042-022-12370-9
5. Ijjina, E.P., Chand, D., Gupta, S., Goutham, K.: Computer vision-based accident detection in traffic surveillance. In: 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1–6 (2019). https://doi.org/10.1109/ICCCNT45670.2019.8944469
6. Le, T.N., Ono, S., Sugimoto, A., Kawasaki, H.: Attention R-CNN for accident detection. In: 2020 IEEE Intelligent Vehicles Symposium (IV), pp. 313–320 (2020). https://doi.org/10.1109/IV47402.2020.9304730
7. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
8. Luque, R.M., Domínguez, E., Palomo, E.J., Muñoz, J.: A neural network approach for video object segmentation in traffic surveillance. In: Campilho, A., Kamel, M. (eds.) ICIAR 2008. LNCS, vol. 5112, pp. 151–158. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-69812-8_15
9. Molina-Cabello, M., Luque-Baena, R., López-Rubio, E., Thurnhofer-Hemsi, K.: Vehicle type detection by ensembles of convolutional neural networks operating on super resolved images. Integr. Comput.-Aided Eng. **25**(4), 321–333 (2018). https://doi.org/10.3233/ICA-180577
10. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks (2015). https://doi.org/10.48550/ARXIV.1506.01497
11. Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning (2019). https://arxiv.org/abs/1905.11946
12. Wang, C., Dai, Y., Zhou, W., Geng, Y.: A vision-based video crash detection framework for mixed traffic flow environment considering low-visibility condition. J. Adv. Transp. **2020**, 1–11 (2020). https://doi.org/10.1155/2020/9194028
13. Youssef, Y., Elshenawy, M.: Automatic vehicle counting and tracking in aerial video feeds using cascade region-based convolutional neural networks and feature pyramid networks. Transp. Res. Record: J. Transp. Res. Board **2675**(8), 304–317 (2021). https://doi.org/10.1177/0361198121997833
14. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points (2019). https://arxiv.org/abs/1904.07850

Image Classification Applied to the Problem of Conformity Check in Industry

Nour Islam Mokhtari[1], Igor Jovančević[1,2(✉)], Hamdi Ben Abdallah[3], and Jean-José Orteu[3]

[1] DIOTASOFT, 201 Pierre and Marie Curie Street, 31670 Labège, France
[2] Faculty of Natural Sciences and Mathematics, University of Montenegro, Cetinjska 2, 81000 Podgorica, Montenegro
igorj@ucg.ac.me
[3] Institut Clément Ader (ICA), Université de Toulouse, CNRS, IMT Mines Albi, INSA, UPS, ISAE, Campus Jarlard, 81013 Albi, France

Abstract. This paper shows the application of several learning-based image classification techniques to conformity check, which is a common problem in industrial visual inspection. The approaches are based on processing 2D images. First, a classification pipeline has been developed. An effort has been invested into choosing an appropriate classifier. First experiment was performed with *HoG* features (Histogram Of Gradient) and Support Vector Machine (*SVM*). Further, to improve accuracy, we employed a *bag of visual words* (BoVW) and *ORB* detector for extracting features that we further use to build our *dictionary of visual words*. The final solution uses features extracted by passing an image through a pre-trained deep convolutional neural network *Inception*. Using these features a *SVM* classifier was trained and high accuracy was obtained. To augment our image data set, different transformations such as zoom and shearing were applied. Promising results were obtained which shows that state-of-the-art deep learning classification techniques can be successfully employed in the visual industrial inspection field.

1 Introduction

Visual inspection of mechanical assemblies is an essential phase in the production process. The objective of this paper is to automate the inspection procedures by developing computer vision algorithms able to perform this task. The developed algorithms are exploiting 2D images and 3D point clouds and are running on a robotic platform or on a hand-held tablet. The Computer-Aided Design (CAD) model of the inspected mechanical assemblies is utilized as an input for in-house developed CAD-based 2D/3D localization module. This module provides a close enough estimate of the relative pose between the camera and the assembly being controlled. The pose estimate is a requirement imposed by the algorithms to properly observe an element of interest.

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 340–349, 2023.
https://doi.org/10.1007/978-3-031-18050-7_33

Treated inspection problem belongs to a large group of problems. The challenge is to verify the presence of certain elements at their pre-specified locations in an assembly. To validate the approach, four real use cases are solved. They concern metallic parts in the aerospace manufacturing domain and verifying their presence at predefined locations.

2 Related Work

In the previous papers, the authors are dealing with visual inspection challenges in the aerospace industry, by employing conventional image processing [1–3] and 3D point cloud processing techniques [4,5] or recent deep learning architectures on 3D point clouds [6]. In this work, a way has been found to provide enough 2D images in order to evaluate some known machine learning approaches and some deep learning architectures.

Inspection based on 2D image analysis has been in the focus of many works. Generally, the use of 2D image analysis for defect detection is preferred over 3D point clouds, because of its lower computation time. Leiva *et al.* [7] tackled a visual detection and verification of exterior aircraft elements. Authors in [8] developed a fully automated vision system to inspect the surface of crankshafts.

In vision-based inspection, machine learning models that perform classification and object detection can be very useful bricks. Park *et al.* [9] present a convolutional neural network (CNN) to detect dirties, scratches, burrs, and wears. Miranda *et al.* [10] presented a CNN-based method to detect and inspect screws on aircraft fuselage images acquired by a UAV. Jong-Chih *et al.* [11] developed a machine-learning method to classify 4 types of visible surface defects on semiconductor wafers.

In most cases, there are not enough images to train the models because assemblies are rarely available for data acquisition. This paper will present an approach to employ classification while having little data.

3 Methodology and Results

Following sections will demonstrate the experiments with multiple solutions and gradual development of the image classification pipeline to verify the presence of objects mounted on a mechanical assembly. It will also show how a small dataset has been augmented.

3.1 Problem Statement - Verifying Presence by Classification

The aim is to verify that a screw is present (as in Fig. 1a) in a predefined location on a mechanical assembly. The localization module enables cropping an input image and extracting a region of interest around the predefined location of the screw. Figure 1 shows 3 samples that the classification algorithm should be able to differentiate. The methodology and the results of three different evaluated approaches will be outlined. The gradual improvement in the results will be demonstrated. The approaches have been applied to six different tasks but here only four of them will be presented.

3.2 SVM with HoG

Gaussian Support Vector Machine (SVM) with Histogram of Gradients (HoG) descriptor has been a popular solution to similar tasks [12,13]. So this approach has been employed first. C and γ are parameters that can control the SVM decision boundary. To choose the best parameters C and γ for the classifier, *cross-validation* has been used. Namely, our dataset has been split into 2 subsets, 80% for training and 20% for testing. Then the classifier has been trained on the larger subset with the chosen parameters and then tested on the smaller set. This way, the parameters C and γ that maximize both the training accuracy and the testing accuracy have been chosen. Also, this way the over-fitting problem has been addressed, since the trained classifier has been tested on a subset that was not seen before. This will subsequently give an idea on how well the classifier will perform on new data in the future. The tests are listed in Tables 1 and 2. It can be noted that the results are not satisfactory.

3.3 SVM with BoVW

Further on, HoG was replaced by *bag of visual words* (BoVW) which is a feature representation technique that uses a partly unsupervised machine learning scheme based on *k-means* algorithm. First step is to construct a *visual dictionary* of BoVW descriptors in the points that were previously extracted by the ORB detector. The cross-validation approach has been used, as in Sect. 3.2, to choose the optimal parameters. Main focus was put on two parameters that have the most influence on the results: C for SVM and K for the size of the visual words dictionary. In Tables 3 and 4 a significant improvement in accuracy can be observed, compared to SVM+HoG approach.

The 2 parameters $\max_{\text{iter}}$ and ϵ work as a stopping criteria for the SVM algorithm. The SVM optimization will stop if one of the 2 criteria is reached: either the loss function value became lower than ϵ or a maximum number of iterations ($\max_{\text{iter}}$) was reached.

Table 1. Different tests using SVM with HoG (C is variable, γ is fixed)

SVM params	Accuracy (for train set and test set)	
$C = 0.1$	train set = 48.62%	test set = 56.52%
$C = 0.5$	train set = 50.00%	test set = 50.00%
$C = 1.0$	train set = 100.0%	test set = 50.00%
$\mathbf{C = 2.0}$	train set = 100.0%	test set = **60.00%**
$C = 4.0$	train set = 100.0%	test set = 43.48%
$C = 8.0$	train set = 100.0%	test set = 45.65%
$C = 16$	train set = 100.0%	test set = 50.00%

Table 2. Different tests using SVM with HoG (γ is variable, C is fixed)

SVM params	Accuracy (for train set and test set)	
$\boldsymbol{\gamma} = \mathbf{0.1}$	train set = 100%	test set = **58.69%**
$\gamma = 0.5$	train set = 100%	test set = 56.52%
$\gamma = 1.0$	train set = 100%	test set = 50.00%
$\gamma = 2.0$	train set = 100%	test set = 52.17%
$\gamma = 4.0$	train set = 100%	test set = 41.30%
$\gamma = 8.0$	train set = 100%	test set = 32.61%
$\gamma = 16$	train set = 100%	test set = 54.35%

Table 3. Accuracy when changing K and fixing SVM parameters $\gamma = 1$ and $C = 1$

k-means	Accuracy (80% train; 20% test)	
$K = 10$	train set = 83.88%	test set = 82.97%
K = 30	train set = 86.71%	test set = **90.33**%
$K = 50$	train set = 88.56%	test set = 87.81%
$K = 60$	train set = 86.16%	test set = 85.43%
$K = 80$	train set = 88.55%	test set = 86.81%
$K = 100$	train set = 88.41%	test set = 87.36%

Table 4. Accuracy when changing C and fixing $K = 50$, $\max_{\text{iter}} = 2000$, $\epsilon = 0.01$ and $\gamma = 1$

SVM	Accuracy (80% train; 20% test)	
$C = 1$	train set = 88.56%	test set = 87.81%
$C = 2$	train set = 90.34%	test set = 90.91%
$C = 4$	train set = 91.91%	test set = 89.83%
$C = 10$	train set = 92.19%	test set = 92.01%
C = 20	train set = 92.67%	test set = **92.54**%
$C = 30$	train set = 4.03%	test set = 90.93%

In the experiments with BoVW, the reached accuracy was around 92.54% on the augmented data set. This means that he system is still making false predictions 7.5% of the time, which is unacceptable, by industry standards. This led us to using a more informative descriptor such as deep learning architecture which will be presented in the next section.

3.4 Linear SVM with CNN Features

The final solution is based on deep learning feature extraction and SVM classification. For this approach, more data are required, hence they needed to be generated. In industry, there are usually two large problems: the lack of data and the classes being imbalanced. To tackle these two challenges, a data augmentation scheme was included, followed by some manual data cleaning. Starting from a dataset of 381 images, a dataset of 1826 images was obtained. This augmented data set was distributed by class as follows: 672 images that contain a screw (Fig. 1a), 673 images that do not contain a screw (Fig. 1b) and 517 images where it is not possible to decide (Fig. 1c). The affine transformations were applied to increase the dataset. Those transformations were mainly a combination of zooming (to account for scale), skewing (to account for different camera poses), and cropping. Three examples of augmented data are shown in Fig. 2.

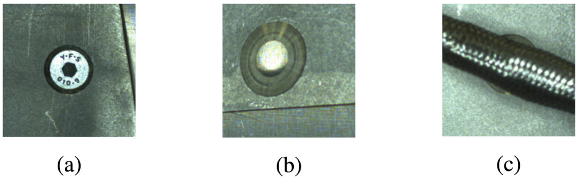

(a) (b) (c)

Fig. 1. Data before augmentation: (a) screw is present, (b) screw is absent, (c) unknown if screw is present

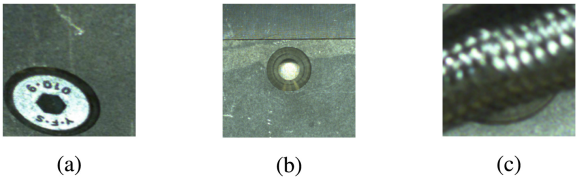


Fig. 2. Data after augmentation: (a) screw is present (skew, zoom, crop), (b) screw is absent (rotation, skew), (c) unknown if screw is present (zoom)

In paper [14], authors reported that, by using pre-trained networks such as VGG and Inception to extract features from images and then applying some traditional classifiers such as SVM, they were able to achieve the state of the art results in both classification and object detection. This approach is appropriate for the case treated in this paper, especially because not a lot of data is available for training a CNN from scratch, even with data augmentation. The working model of the proposed approach is demonstrated in Fig. 3.

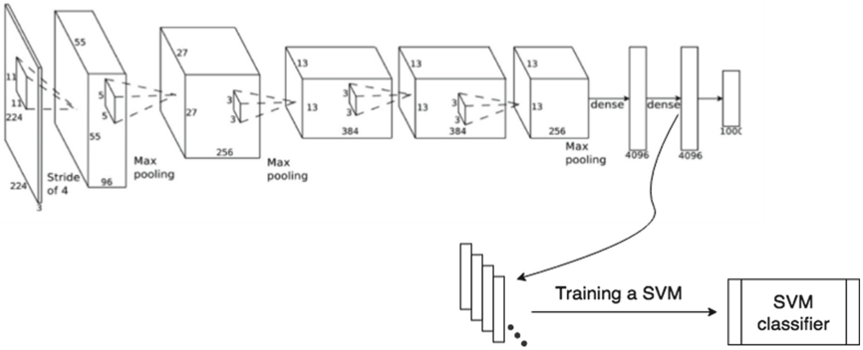


Fig. 3. Our scheme at training time

First, the images are passed through a pre-trained network such as Inception or VGG. Further, for each image, the output of the bottleneck layer is extracted, also called the transfer layer (referring to transfer learning). It has the size of 2048 in the Inception network. Finally, feature vectors from all the images are collected and used for training an SVM classifier.

Performed operations at test time are shown in Fig. 4. First, the test image is passed through the network, then the transfer layer output is extracted and passed through the trained SVM classifier. Finally the corresponding prediction is obtained.

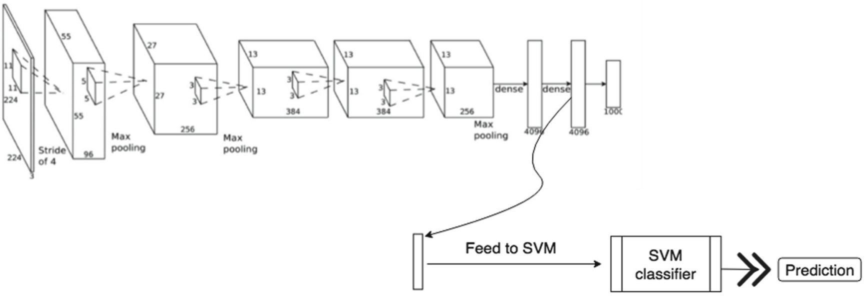


Fig. 4. Scheme at test time

To test this approach, a linear SVM with the extracted features is used. The results are shown in Table 5. It can be noted that the classification accuracies are very high. This is the best classifier used and the efficiency of this classifier comes first and foremost from the pre-trained CNN *Inception v1*. The reason why the extracted features are very discriminative is because the network has been trained on millions of images from ImageNet data set [15] and it has learned to extract the most relevant parts from the image; most relevant in the sense of distinguishing between different classes. Developed pipeline was evaluated on five other and even more challenging classification tasks and some very good results were obtained.

Table 5. Inception v1 and linear SVM

SVM	Accuracy
$C = 1$	training = 100%, testing = 99.73%
$C = 5$	training = 100%, testing = 100%

In Fig. 5 another inspection use case with 5 different classes can be observed. In this task, the objective is to build a system that can distinguish between 5 different types of screws that can be found in some airframes.

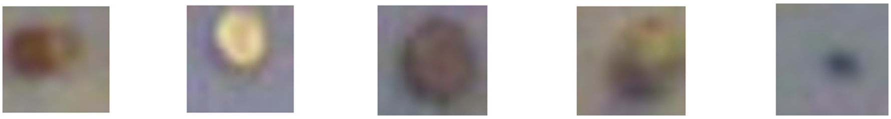


Fig. 5. Small support screws samples

Dataset contains 1227 images: 137 in class1, 116 in class2, 395 in class3, 310 in class4 and 269 in class5. The dataset has been split in an 80%/20% manner. Obtained accuracies are very high: 100% on the train set and 94.5% on the test set.

The same approach that consists of a linear SVM with InceptionV1 extractor was used in two other tasks that are described below.

Task 1: The goal is to develop a system that can recognize one of three possible scenarios: a red clump exists (Fig. 6), no red clump exists (Fig. 7) and unknown state (the clump might be hidden by another part) (Fig. 8).

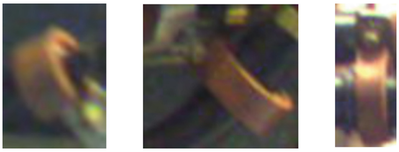


Fig. 6. Class 1: presence of red clumps (*valid* state) (Color figure online)

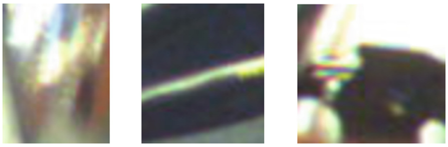


Fig. 7. Class2: absence of clumps (*invalid* state)

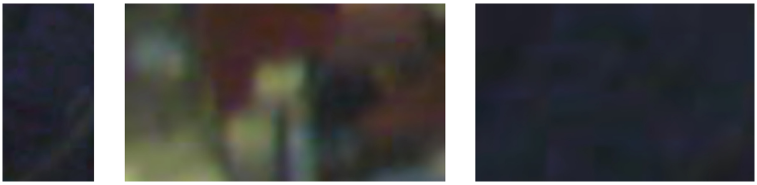


Fig. 8. Class 3: *Unknown* state (decision can not be made, often acquisition or localization problem)

Dataset contains 712 images distributed among 3 classes: 234 *valid* state, 164 *invalid* state, 37 *unknown* state). The set has been split as previous sets: 80%/20%. Obtained accuracy on training set was 100% and on testing set: 98.03%.

Task 2: The goal is to be able to distinguish between three different states on a mechanical engine. The first state: the metallic box is mounted on the engine (Fig. 9). The second state: the box is absent (Fig. 10). The third state: it is not possible even for a human to decide whether the box is mounted or not, for example when it is hidden by other parts of the engine (Fig. 11).

Dataset consisted of 1022 images, distributed among 3 classes: 633 images where the box is present, 201 images with no box and 188 images with unknown state. After splitting the dataset 80%/20%, the following accuracies were reached: 100% on training set, and 98.03% on testing set.

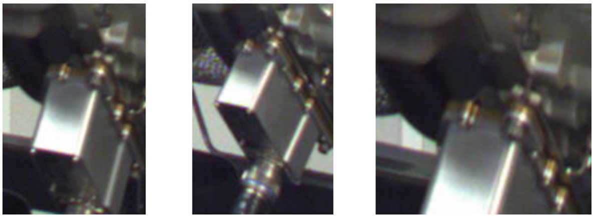


Fig. 9. Class 1: presence of the metallic box (valid state)

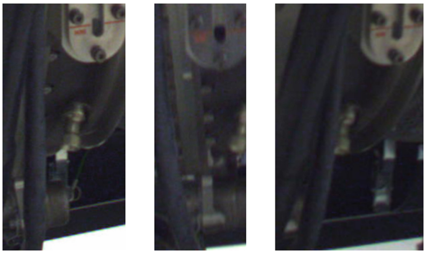


Fig. 10. Class 2: absence of the metallic box (invalid state)

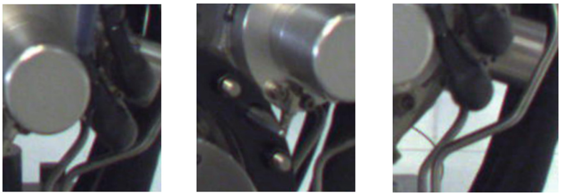


Fig. 11. Class 3: unknown state (decision can not be made, often acquisition or localization problem)

4 Conclusion

This work applied state-of-the-art machine learning to a known industrial visual inspection problem: conformity check. The main tool is classification for 2D images. Developed classification pipeline consists of a feature extractor and a supervised machine learning scheme, i.e. SVM classifier. It is designed to solve the problem of verifying the presence of different mechanical parts. Different feature extractors were evaluated and it has been found that a CNN architecture outperformed traditional feature extractors: HoG and BoVW. Namely, a pre-trained neural network *Inception v1* was employed to extract feature vectors from images. Further, these feature vectors were used to train a linear SVM. This approach generated very high accurracies. This work shows that machine learning can be employed for industrial applications even with small size datasets. The experiments that were done can easily be extended to other industries such as manufacturing or automotive industry.

Acknowledgements. This research has been carried out within the joint research laboratory "Inspection 4.0" between IMT Mines Albi/ICA and DIOTA. This laboratory is co-funded by the French "Région Occitanie". We would like to thank the Région Occitanie for its financial support.

References

1. Jovancevic, I., Larnier, S., Orteu, J.-J., Sentenac, T.: Automated exterior inspection of an aircraft with a pan-tilt-zoom camera mounted on a mobile robot. J. Electron. Imaging **24**(6), 1–15 (2015)
2. Ben Abdallah, H., Jovancevic, I., Orteu, J.-J., Brèthes, L.: Automatic inspection of aeronautical mechanical assemblies by matching the 3D CAD model and real 2D images. J. Imaging **5**, 81–108 (2019)
3. Viana, I., Orteu, J.-J., Cornille, N., Bugarin, F.: Inspection of aeronautical mechanical parts with a pan-tilt-zoom camera: an approach guided by the computer-aided design model. J. Electron. Imaging **24**, 061118 (2015)
4. Jovancevic, I., et al.: 3D point cloud analysis for detection and characterization of defects on airplane exterior surface. J. Non Destr. Eval. **36**, 74 (2017)
5. Ben Abdallah, H., Orteu, J.-J., Jovancevic, I., Brèthes, L., Dolives, B.: 3D point cloud analysis for automatic inspection of complex aeronautical mechanical assemblies. J. Electron. Imaging **29**(04), 1–22 (2020)
6. Mikhailov, I., Jovancevic, I., Mokhtari, N., Orteu, J.-J.: Classification using a three-dimensional sensor in a structured industrial environment. J. Electron. Imaging **29**(4), 1–14 (2020)
7. Leiva, J., Villemot, T., Dangoumeau, G., Bauda, M.-A., Larnier, S.: Automatic visual detection and verification of exterior aircraft elements. In: 2017 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM), pp. 1–5. IEEE, May 2017
8. Tout, K., Bouabdellah, M., Cudel, C., Urban, J.-P.: Automated vision system for crankshaft inspection using deep learning approaches. In: Fourteenth International Conference on Quality Control by Artificial Vision, vol. 78. SPIE, July 2019
9. Park, J.-K., Kwon, B.-K., Park, J.-H., Kang, D.-J.: Machine learning-based imaging system for surface defect inspection. Int. J. Precis. Eng. Manuf.-Green Technol. **3**(3), 303–310 (2016). https://doi.org/10.1007/s40684-016-0039-x
10. Miranda, J., Veith, J., Larnier, S., Herbulot, A., Devy, M.: Machine learning approaches for defect classification on aircraft fuselage images aquired by an UAV. In: Fourteenth International Conference on Quality Control by Artificial Vision, vol. 10. SPIE, July 2019
11. Chien, J.-C., Wu, M.-T., Lee, J.-D.: Inspection and classification of semiconductor wafer surface defects using CNN deep learning networks. Appl. Sci. **10**(15), 5340 (2020)
12. Navneet, D., Bill, T.: Histograms of oriented gradients for human detection. In: Schmid, C., Soatto, S., Tomasi, C. (eds.) International Conference on Computer Vision & Pattern Recognition (CVPR 2005), vol. 1, pp. 886–893. IEEE Computer Society, San Diego, United States, June 2005
13. Xiao, J., Hays, J., Ehinger, K., Oliva, A., Torralba, A.: Sun database: large-scale scene recognition from abbey to zoo. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3485–3492. IEEE, June 2010

14. Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: an astounding baseline for recognition. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, vol. 1403, pp. 512–519. IEEE, March 2014
15. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. **115**, 211–252 (2014)

A Virtual Sensor Approach to Estimate the Stainless Steel Final Chemical Characterisation

Damián Nimo[1(✉)], Javier González-Enrique[1], David Perez[2], Juan Almagro[2], Daniel Urda[1], and Ignacio J. Turias[1]

[1] Department of Computer Science, University of Cadiz, Cadiz, Spain
damian.nimojarquez@uca.es

[2] Dpto. Técnico, Polígono Industrial Los Barrios ACERINOX Europa, S.A.U., Los Barrios, Spain

Abstract. This paper presents an approach based on machine learning methods to solve a real industrial problem. During the manufacture of stainless steel with certain characteristics, due to the manufacturing process itself, the steel moves away from the ideal conditions and it is necessary to determine how far the final product is from the desired one. For this determination, a procedure for the development of a virtual sensor has been carried out to replace the current semi-manual procedure of the ACERINOX EUROPA, S.A.U. factory in Cadiz. The results obtained are very promising and it is planned to install an application in the factory to work initially in parallel to the human expert until it can be used as a stand-alone application.

Keywords: Machine learning · Artificial intelligence · Classification · Industry 4.0

1 Introduction

The manufacture of stainless steel [3] is a complex industrial process that is based on the rolling of a casting of material that is melted in a steel mill. This industrial process is influenced by many variables and process conditions from the beginning (scrap) to the end (rolled steel coils) [8]. With the advent of Industry 4.0, industries want to use smart technologies to combine these complex production techniques with the efficient use of people and assets [4]. In this context, Artificial Intelligence Techniques are being applied to solve different problems in manufacturing systems [6], also making use of the advantages of the Internet of Things [5].

Artificial Intelligence techniques have been traditionally used to solve problems that require, by their nature, intelligent strategies to be solved. However, most of these problems in the industry require ad-hoc solutions whose design and implementation are fixed on solving a specific problem in a specific case [9].

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 350–360, 2023.
https://doi.org/10.1007/978-3-031-18050-7_34

In concrete, the problem we are focusing on in this study consists of verifying whether the final characteristics of a material (stainless steel) coincide with its theoretical characteristics. This determination procedure requires human intervention by an expert who takes into account: the real chemical composition data of casting, the chemical composition data of a coil sample of the final rolled steel and the data of related castings (the previous and the following ones), as well as the historical data of confusion tests performed and the tolerances allowed in the tests. They also take into consideration the permitted standards for the steels and the expert knowledge of the company's technicians. This check is called a confusion test.

Depending on the degree of deviation from the real conditions, the material is classified as such. Thus, there are 1 to 5 kinds of final determinations: 1, the material is correct with its casting; 2, it does not match its casting, but matches the steel type; 3, it does not match its casting and does not match the steel type; 4, the material has a mixture, but matches the steel type; and 5, it has a mixture, but does not match the steel type. This process is currently very manual, requiring highly trained personnel, and is not exempt from possible deficiencies due to human error, from simple absent-mindedness to failures that cannot be detected by the personnel. Solving these problems involves a large investment in man-hours from the company's side, which translates into costs. This is why an AI-based approach is needed that matches the efficiency of trained technical staff, but does not fall into the same problems as using human operators and allows the company to divert the cost of solving this problem to other work.

In this paper, we are going to try to solve the problem of the classification of stainless steel using three different models, a Neural Network, a Naive Bayesian Classifier and a Support Vector Machine that replicates the work carried out by the expert technicians in charge of this task in the factory. For this purpose, the data set provided by the company we are working with, ACERINOX Europa S.A.U. a stainless steel factory located in Los Barrios (Spain), will be used. These data were collected from 2014 to 2019. They have some other approaches to machine learning to solve other problems in their factory as we can see in [10].

The rest of the paper is organised as follows: Sect. 2.1 describes the data used. Section 2.2 the machine learning methods tested in the analysis. Section 3 shows the results of the study. Finally, some concluding remarks are provided in Sect. 4.

2 Materials and Methods

2.1 DataSet

The data set is provided by the company ACERINOX Europa S.A.U. belonging to the Factory of Palmones, Los Barrios, Cádiz. This dataset is composed of different types of stainless steel: austenitic, duplex, ferritic and martensitic. The total data sample consists of 6961 samples belonging to the different types of steel. In addition, each sample is described by 35 variables. Each sample is divided into 19 elements which made a 665 variable for a sample of a steal. Some of these variables are associated with other auxiliary data sets such as the castings related to the sample described in the steel and the chemical composition of these castings. This auxiliary data set is composed of the different casts used in the main data set and has 18 variables, each casting is composed of 19 elements witch it makes 342 variables for one casting.

The list of total data can be seen in the Table 1. In it, we can see how the number of data for certain types of steel is very small as is the number of samples for classes 2, 3 and 5. For this reason, the study is going to be carried out on two different types of steel: steel type 150 and 240, as although steel type 160 has more elements than steel type 240, it is a very similar type of steel to steel type 150, for this reason, it was discarded for the study, opting for steel type 240.

Now that we have the dataset focused on two specific types of steels, we can see, as we have said, that the numbers of samples from classes 2, 3 and 5 are very small, so we will focus the study on the data from classes 1 and 4.

Table 1. Total data according to the type of Steel and the numbers of elements of each class within it

Steel[1]	Total	Class 1	Class 2	Class 3	Class 4	Class 5
60	24	24	0	0	0	0
120	65	47	0	0	18	0
140	105	82	2	0	19	2
150	**2833**	**2412**	**14**	**4**	**400**	**3**
160	2555	2190	8	6	348	3
180	6	6	0	0	0	0
240	**782**	**667**	**3**	**2**	**109**	**1**
260	5	5	0	0	0	0
280	113	102	0	2	9	0
315	261	228	0	0	33	0
350	166	152	0	0	14	0
380	3	3	0	0	0	0
390	8	4	0	0	2	2
500	6	4	0	0	2	0
800	12	11	0	0	1	0
900	17	14	3	0	0	0

The identification of steel types is an internal classification of the company.

Due to the large difference between samples of one class type and another it was decided to resample [7] the data as a classifier usually fails to classify in the small class due to the smaller number of class instances. For this resampling, the Safe-Level-SMOTE [2] method was used which generates samples of the minority class within the overlapping region. This is achieved by selecting minority instances with different weights, called safe levels, using the Nearest Neighbour technique on the minority samples. This leaves us with four datasets shown in the Table 2.

Table 2. Total data of the four datasets.

DataSet	Steel	Total^2	Class 1	Class 4
Without resampling	150	2812	2412	400
Without resampling	240	776	667	109
With resampling	150	12448	9648	2800
With resampling	240	3431	2668	763

Total data of Class 1 and 4

Each sample in the dataset is reduced by removing repeated features that are common to all the elements that make up the dataset. This reduces the number of characteristics of the sample to 53. And its representation consists of an alphanumeric vector with each of the characteristics in its indices.

In addition, the castings to which the steels belong have a specific chemical composition. As with the samples of the main dataset, the samples of the dataset of the chemical compositions of the castings are stripped of the repeated elements common to all the elements of the casting. This leaves the samples of the chemical compositions of the casts represented as an alphanumeric vector of 36 indices, representing each of the remaining characteristics of the samples.

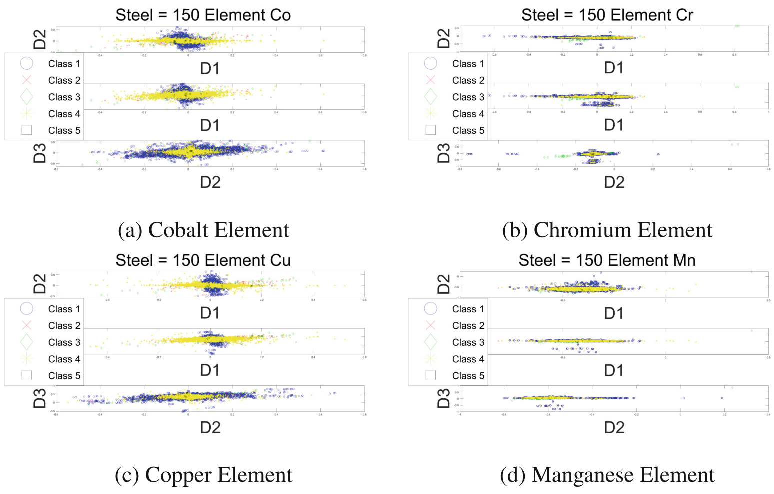
Each sample of a steel type is related to three different casts, the current cast to which it belongs and the casts before and after the current cast. Therefore, to represent a final sample from each dataset, the vector representing the steel sample is concatenated with the three vectors of the related casts; previous, current and following. With this, each sample of the datasets is represented as an alphanumeric vector whose indices represent the 164 characteristics of the sample.

Finally, data that are not necessary for the analysis, such as dates and the like, are eliminated, and a vector is created, and concatenated with the previous one, with the possible errors of the sample about the Adjudication Rules according to the type of steel in question. This leaves us at the end with a numerical vector of 135 positions representing the characteristics of the data for the representation of each sample in the datasets.

2.2 Methods

Given the high dimensionality of the data, a feature reduction was performed using a priori information from the data. This gave us the possibility to represent in two dimensions a point cloud representing the data of each class to see which elements of the steel type can be used to better discern between elements of one class and another. This can be seen in Figs. 1 and 2.

Figure 1 plots the steel type 150 data, using the a priori knowledge to represent its chemical elements in two dimensions plotted using the first a priori



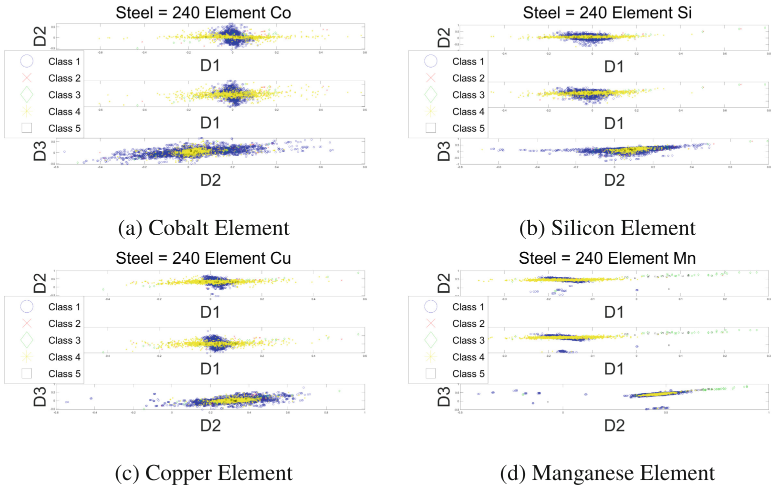
(a) Cobalt Element

(b) Chromium Element

(c) Copper Element

(d) Manganese Element

Fig. 1. Some elements represented in 2D their relationship using a priori knowledge for Steel of type 150



(a) Cobalt Element

(b) Silicon Element

(c) Copper Element

(d) Manganese Element

Fig. 2. Some elements represented in 2D their relationship using a priori knowledge for Steel of type 240

knowledge D1, Euclidean distance between the value of the element in the sample and the corresponding value of the element in the current casting, against the second a priori knowledge D2, between the sample and previous casting values, and the third, D3, between the sample and following casting values, as well as the second against the third. Thus in Fig. 1a we see the distribution of the data based on Cobalt in which many data belonging to Class 4 and Class 1 are overlapping, however, several dividing lines can be drawn on the sides of the point cloud to discern elements of Class 4 that are not matched in Class 1. Horizontal lines can likewise be drawn around the point cloud of Class 4 elements that separate Class 1 elements from Class 4 elements. This division, also present in Fig. 1c for the chemical element Copper, can be made using all three types of a priori knowledge.

While it is true that this separation can be observed in Fig. 1d for Manganese and in Fig. 1a for Chromium, in a less pronounced way, its use was determined by having this separation in different areas from the other elements, which can help to resolve disputes in the classification.

In Fig. 1b it can also be seen how Class 3 elements are separated from the rest, however, this separation only occurs for this class in this type of steel and this element, so it was decided to continue with the separation of the data into two classes, Class 1 and Class 4, and leave this for a later study.

This was done analogously for the subset of data belonging to steel type 240 in Fig. 2.

After this study, we managed to reduce the dimensionality of each steel to 15, five elements for each a priori knowledge, in the case of steel 150 and to 12, four elements for each a priori knowledge, in the case of steel 240. The elements used can be seen in the Table 3.

Table 3. Total dataset elements used

Steel	Elements
150	Co, Cr, Cu, Mn, Mo
240	Co, Cu, Mn, Si

For resampling, we use the Safe-Level SMOTE [2] method to increase the data of Class 1 by x3 and Class 4 by x6. This leaves us with the number of dataset elements reflected in the Table 2.

Three different models were used for classification: A Neural Network(NN) based model, for this the MatLab function *patternnet* was used which provides a feed-forward neural network with a hidden layer that can be trained for classification. The algorithm used to train this network was the Scaled Conjugate Gradient [11], determining the optimal number of neurons by 10-fold cross-validation with 75% for training and 25% for testing.

Another model used is a Naive Bayesian Classifier [13](NB) with the Kernel density estimation function to estimate the probability density function with a Normal distribution as Kernel smoother type and Kernel smoothing density support is all real values. The MatLab function *fitcnb* was used for its creation and training.

The final model used is a Support Vector Machine [12](SVM) with a polynomial kernel [1] of grade three which allows the use of learning for non-linear models. This model was created with *fitcsvm* function of MatLab.

The data was divided into 75% for training and 25% for testing the models and validated by 10-fold cross-validation.

3 Results

After splitting the data sets, experiments were performed on them. For the data sets without resampling, steel type 150 showed an Overall Accuracy of 95.88% in the Neural Network based model. This same model reported for this steel an accuracy of 99.46% for Class 1 and an accuracy of 74.25% for Class 4 samples. The Bayesian Classifier for steel type 150 without resampling gave an Overall Accuracy of 93.95%. For Class 1 samples it reported an accuracy of 96.06% and accuracy of 81.25% for Class 4. The SVM model reported an Overall Accuracy of 94.17% for steel type 150 without resampling. For Class 1 it reported accuracy of 98.41% and for Class 4 samples an accuracy of 68.63%.

For type 240 steel samples without resampling, the Neural Network-based model obtained an Overall Accuracy of 94.97% and an accuracy of 99.7% for Class 1 and an accuracy of 66.09% for Class 4 samples. On the other hand, the Bayesian Classifier obtained, for this dataset, an Overall Accuracy of 96.27% and in its division by classes, it reported an accuracy of 98.2% for Class 1 and accuracy of 84.45% for Class 4 samples. The SVM model reported an Overall Accuracy of 97.53%. An accuracy of 99.05% for samples of Class 1 and 86.51% accuracy for samples of Class 4. These results can be seen in Table 4.

These models report very high Overall Accuracy but very uneven accuracy between the class samples. As can be seen in Table 6 with their confusion matrices, and their measures in Table 8, this is due to the large difference in samples between Class 1 and Class 4, so we went on to use the models on the resampled datasets to test their performance and obtained the results shown in Table 5 and Table 7.

For the steel type 150 data set with resampling, the Neural Network-based model reported an Overall Accuracy of 95.7% and in its division by classes it achieved an accuracy of 97.76% for Class 1 samples and an accuracy of 83.25% for Class 4 samples. For the same dataset, the Bayesian Classifier-based model obtained an Overall Accuracy of 92.89% and an accuracy of 94.28% for Class 1 samples and an accuracy of 84.5% for Class 4 samples. The SVM model reported an Overall Accuracy of 93.8%, 96.47% accuracy for samples of Class 1 and accuracy of 77.67% for Class 4 samples.

For the Type 240 dataset with resampling, the Neural Network-based model obtained an Overall Accuracy of 94.46%, an accuracy of 97.6% for Class 1 samples and an accuracy of 75.27% for Class 4 samples. The Bayesian Classifier-based model reported an Overall Accuracy of 95.62%, an accuracy of 97.16% for the classification of Class 1 samples and an accuracy of 86.09% for Class 4 samples. The SVM model obtained an Overall Accuracy of 96.33%, an accuracy of 97.61% for Class 1 samples and 87.06% accuracy for samples of Class 4.

Although the Overall Accuracy is slightly lower in the models using the resampled datasets, the classification accuracy for the Class 4 samples is significantly higher.

Looking at the results we can see how the Neural Networks based model performs somewhat better in Overall Accuracy than the SVM model and the Bayesian Classifier-based model in the dataset without resampling, the latter having a vastly higher accuracy for Class 4 samples. This changes with the resampling where SVM reported the best results in both overall accuracy and accuracy for Class 4.

Since, as mentioned above, correct classification of the classes is more important in the problem at hand, the Support Vector Machine model is better suited to solve it with resample while the Naive Bayesian classifier-based model is better suited to solve it without resampling.

Table 4. Results of the models divided into overall accuracy and accuracy per class

Model	Steel	OA	Class 1 accuracy	Class 4 accuracy
NN	150	95.88	99.46	74.25
NB	150	93.95	96.06	81.25
SVM	150	94.17	98.41	68.63
NN	240	94.97	99.7	66.09
NB	240	96.27	98.2	84.45
SVM	240	97.53	99.05	86.51

Table 5. Model results divided into overall accuracy and accuracy per class using the DataSet with Resampling

Model	Steel	OA	Class 1 accuracy	Class 4 accuracy
NN	150	95.7	97.76	83.25
NB	150	92.89	94.28	84.5
SVM	150	93.80	96.47	77.67
NN	240	94.46	97.6	75.27
NB	240	95.62	97.16	86.09
SVM	240	96.33	97.61	87.06

Table 6. Confusion Matrices for Models without Resampling

Steel 150 Model NN

	Class 1	**Class 4**
Class 1	2399	13
Class 4	103	297

Steel 150 Model NB

	Class 1	**Class 4**
Class 1	2317	95
Class 4	75	325

Steel 150 Model SVM

	Class 1	**Class 4**
Class 1	2374	38
Class 4	125	275

Steel 240 Model NN

	Class 1	**Class 4**
Class 1	665	2
Class 4	36	73

Steel 240 Model NB

	Class 1	**Class 4**
Class 1	655	12
Class 4	16	93

Steel 240 Model SVM

	Class 1	**Class 4**
Class 1	661	6
Class 4	14	95

Table 7. Confusion Matrices for Models with Resampling

Steel 150 Model NN

	Class 1	**Class 4**
Class 1	7074	162
Class 4	402	1998

Steel 150 Model NB

	Class 1	**Class 4**
Class 1	6823	413
Class 4	372	2028

Steel 150 Model SVM

	Class 1	**Class 4**
Class 1	6981	255
Class 4	535	1865

Steel 240 Model NN

	Class 1	**Class 4**
Class 1	1953	48
Class 4	161	493

Steel 240 Model NB

	Class 1	**Class 4**
Class 1	1945	56
Class 4	90	564

Steel 240 Model SVM

	Class 1	**Class 4**
Class 1	1954	47
Class 4	84	570

Table 8. Confusion Matrices Measures

	Without Resampling						With Resampling					
	Steel 150			Steel 240			Steel 150			**Steel 240**		
Measure	NN,	NB	SVM	NN	NB	SVM	NN,	NB	SVM	NN	NB	SVM
Sensitivity	0.92	0.93	0.95	0.90	0.95	0.97	0.94	0.94	0.92	0.92	0.95	0.95
Specificity	0.97	0.87	0.87	0.98	0.93	0.94	0.92	0.83	0.87	0.91	0.90	0.92
Accuracy	0.93	0.92	0.94	0.91	0.94	0.97	0.94	0.91	0.91	0.92	0.94	0.95
Precision	0.99	0.96	0.98	0.99	0.98	0.99	0.97	0.942	0.96	0.97	0.97	0.97

4 Conclusions

The great complexity of stainless steel manufacturing lies in the many variables that influence it from the beginning of the process to its end. The problem studied is based on the comparison of the final characteristics of the steel produced with the real characteristics of this type of steel. Currently, this comparison is carried out by the company's expert staff, classifying the steel produced into five different classes: 1 - correct product, 2 - does not match its casting but matches the type of steel, 3 - does not match its casting or the type of steel, 4 - the product is a mixture but matches the type of steel, 5 - the product is a mixture and does not match the type of steel. This comparison is called a confusion test.

In the context of bringing this process to Industry 4.0, techniques based on Artificial Intelligence were used, specifically three models, one based on Neural Networks other on a Naive Bayesian Classifier and a Support Vector Machine. These models will be trained and validated on data provided by the company ACERINOX Europa S.A.U. on which a preprocessing was carried out to adjust them to the models used in the study.

The results obtained show how the use of the model based on the Bayesian Classifier provides better results in the datasets without resampling since, although its overall accuracy is not the highest, 93.95% compared to 95.88% of the model based on Neural Network and 94.17% of the Support Vector Machine model for steel type 150 and 96.27% compared to 94.97% of the NN and 97.53% of the SVM for steel 240, it has the highest accuracy for Class 4 for steel type

150, 81.25% compared to 74.25% of the NN and 68.63% of the SVM. And since the difference between the accuracy for Class 4 for steel type 240, 84.45% respect to 86.51%, is so much small concerning the difference in steel type 150, the use of the model reports the best results for the data without resampling.

Looking at the results of the resampled dataset we obtain the accuracy variation shown in Table 9 where it can be seen that the variation of the Overall Accuracy and Precision for Class 1 samples averaged from the models is negligible, a reduction of 0.167 in OA and 0.179 for Class 1, we can consider the use of appropriate resampling and use the models on this dataset.

With the resampled dataset the best performing model is the model based on the Bayesian Classifier because, although as with the data set without resampling it does not have the highest accuracy values for the overall accuracy and precision of Class 1 for either of the two steel types, its accuracy values for the Class 4 are constant and higher than the other two models, 84.5% for steel type 150 and 86.09% for steel type 240 while NB and SVM report 83.25% and 77.67% for steel type 150 and 75.27% and 87.06% for steel type 240.

Table 9. Accuracy Variation with resampling

	Overall accuracy	Class 1 accuracy	Class 4 accuracy
Accuracy Variation	–0.167	–0.179	0.1977

The improvements brought about by the resampling make us optimistic about taking this strategy to the samples of the other classes, for which there was much fewer data available, so the study will continue along this path that has been opened up.

Acknowledgement. This work is supported through grant RTI2018-098160-B-I00 from MICINN-SPAIN and OT2020/091 ACERINOX EUROPA, S.A.U. (A86327632)

References

1. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, pp. 144–152 (1992)
2. Bunkhumpornpat, C., Sinapiromsaran, K., Lursinsap, C.: Safe-Level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In: Theeramunkong, T., Kijsirikul, B., Cercone, N., Ho, T.-B. (eds.) PAKDD 2009. LNCS (LNAI), vol. 5476, pp. 475–482. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01307-2_43
3. Davis, J.R., et al.: Stainless steels. In: ASM International (1994)
4. Dilberoglu, U.M., Gharehpapagh, B., Yaman, U., Dolen, M.: The role of additive manufacturing in the era of industry 4.0. Proc. Manuf. **11**, 545–554 (2017)

5. Dopico, M., Gómez, A., De la Fuente, D., García, N., Rosillo, R., Puche, J.: A vision of industry 4.0 from an artificial intelligence point of view. In: Proceedings on the International Conference on Artificial Intelligence (ICAI), p. 407. The Steering Committee of The World Congress in Computer Science, Computer (2016)
6. Lee, J., Davari, H., Singh, J., Pandhare, V.: Industrial artificial intelligence for industry 4.0-based manufacturing systems. Manuf. Lett. **18**, 20–23 (2018)
7. Li, T., Bolic, M., Djuric, P.M.: Resampling methods for particle filtering: classification, implementation, and strategies. IEEE Signal Process. Mag. **32**(3), 70–86 (2015)
8. Lo, K.H., Shek, C.H., Lai, J.: Recent developments in stainless steels. Mater. Sci. Eng. R. Rep. **65**(4–6), 39–104 (2009)
9. Martínez-López, F.J., Casillas, J.: Artificial intelligence-based systems applied in industrial marketing: An historical overview, current and future insights. Ind. Mark. Manage. **42**(4), 489–495 (2013)
10. Mesa, H., et al.: A machine learning approach to determine abundance of inclusions in stainless steel. In: Pérez García, H., Sánchez González, L., Castejón Limas, M., Quintián Pardo, H., Corchado Rodríguez, E. (eds.) HAIS 2019. LNCS (LNAI), vol. 11734, pp. 504–513. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29859-3_43
11. Møller, M.F.: A scaled conjugate gradient algorithm for fast supervised learning. Neural Netw. **6**(4), 525–533 (1993)
12. Noble, W.S.: What is a support vector machine? Nat. Biotechnol. **24**(12), 1565–1567 (2006)
13. Saritas, M.M., Yasar, A.: Performance analysis of ann and naive bayes classification algorithm for data classification. Int. J. Intell. Syst. Appli. Eng. **7**(2), 88–91 (2019)

Convolutional Neural Networks for Structured Industrial Data

Luis Moles[1], Fernando Boto[2](✉), Goretti Echegaray[3], and Iván G. Torre[4]

[1] Department of Computer Sciences and Artificial Intelligence,
University of the Basque Country (UPV/EHU), Donostia-San Sebastián, Spain
luis.moles@ehu.eus

[2] TECNALIA Research and Innovation. Parque Científico y Tecnológico de
Gipuzkoa, Donostia - San Sebastián, Gipuzkoa, Spain
fernando.boto@tecnalia.com

[3] Department of Applied Mathematics, University of the Basque Country
(UPV/EHU), Donostia-San Sebastián, Spain
goretti.echegaray@ehu.eus

[4] Vicomtech Foundation, Basque Research and Technology Alliance (BRTA),
Donostia-San Sebastián, Spain
igonzalez@vicomtech.org

Abstract. Regression methods aim to predict a numerical value of a target variable given some input variables by building a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$. In Industry 4.0 regression tasks, tabular data-sets are especially frequent. Decision Trees, ensemble methods such as Gradient Boosting and Random Forest, or Support Vector Machines are widely used for regression tasks with tabular data. However, Deep Learning approaches are rarely used with this type of data, due to, among others, the lack of spatial correlation between features. Therefore, in this research, we propose two Deep Learning approaches for working with tabular data. Specifically, two Convolutional Neural Networks architectures are tested against different state of the art regression methods. We perform an hyper-parameter tuning of all the techniques and compare the model performance in different industrial tabular data-sets. Experimental results show that both Convolutional Neural Network approaches can outperform the commonly used methods for regression tasks.

1 Introduction

Industrial processes are strongly being affected by the increase of the digitisation of the manufacturing and production processes, consequence of the Industry 4.0 revolution.

These changes provide a host of advantages, such as waste, overproduction and the reduction of energy consumption [1], reduced production costs and lead times and enhanced quality of the product [2].

Nevertheless, the automation of the manufacturing processes results in an increase in the amount of data available, making it difficult to ensure the quality

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 361–370, 2023.
https://doi.org/10.1007/978-3-031-18050-7_35

and integrity of all of them [3]. In this scenario, the efficient analysis and evaluation of great amounts of data collected from many different sources becomes critical.

This challenge makes technologies such as Big Data, Machine Learning (ML) and Artificial Intelligence (AI) one of the pillars of Industry 4.0. In recent years, Deep Learning techniques are emerging as a really popular option, due to its integration of feature extraction in the training process and its ability to build complex and non-linear relationships between features. It has also proven to be really successful in areas such as computer vision, speech recognition and text generation, where unstructured data as images or text are mostly used.

However, most industrial problems structure data in tabular form, which consists of a set of samples with a collection of features. It is usual to face some issues when dealing with this kind of data, such as missing values, categorical features or lack of prior knowledge of the data set structure. This makes the application of Deep Learning techniques challenging, being decision tree-based algorithms such as Gradient Boosting (GB) the top choice for tabular problems [4].

As we can see, the potential and success of Deep Learning with certain type of data is beyond doubt. Being able to transfer all the benefits of Deep Learning with unstructured data to problems with tabular data, could change the way of approaching those problems and improve the performance of current techniques.

Among all existing Deep Learning approaches, Convolutional Neural Networks (CNN) emerge as one of the most popular and successful ones. These architectures, apart from being computationally efficient, have a great capacity to automatically detect important features of the input without any human supervision, which has made them stand out when working with images, but not with structured data.

Therefore, the main objective of this work is to study the benefits that arise from applying CNNs with structured data in industrial problems. Specifically, we compare the usual techniques in industry with tabular data, such as Random Forest and Gradient Boosting, with two convolutional approximations for this type of data. We perform hyper-parameter tuning for all the models, and we work with an embedding that adapts the data to an image form.

The remainder of the article is organised as follows. Section 2 presents the related work. Section 3 details the goals of the work and explains the followed methodology. Section 4 shows the specific contribution. The results and their analysis are presented in Sect. 5 and the conclusions are shown in Sect. 6.

2 Related Work

Machine Learning has been widely used for solving problems in Industry 4.0, in different areas such as fault detection and predictive maintenance.

Fault detection is a vital task in Industry 4.0, as it guarantees production efficiency by preventing unexpected machinery breakdowns [5].

There are several works that apply Machine Learning in fault detection. In [6] and [7] a board-level functional fault diagnosis is carried out, using different ML models such as Artificial Neural Networks (ANN) and Support Vector

Machines (SVM). [8] shows different ensemble models for concept drift detection with imbalanced industrial data, and [9] compares different machine learning algorithms for class-imbalanced process fault detection problems.

The use of predictive tools to detect the need of maintenance activities allowing early detection of breakdowns is called predictive maintenance. It is another huge mission in Industry 4.0, as we can see in [10]. Several ML approaches have been used for predictive maintenance, as shown in [11,12] and [13].

Prediction of the remaining useful life (RUL) of the machines is another topic that draws the attention of ML solutions. Examples of this are [14], where the RUL of a turbofan engine is predicted, [15] where they predict failure and remaining useful life of a wind turbine gearbox, and [16], a paper that predicts RUL of bearing acoustic emission signals.

We also find ML techniques in other industrial problems such as security and detection of cyber-physical attacks in 3-D printing processes using KNN and Random Forest [17], in prediction of energy consumption [18] and in the creation of data driven models for prediction in steel production processes [19].

Deep Learning approaches are also gaining popularity in Industry 4.0 solutions, as we see in [20] and [21], where they present different DL approaches for predicting the RUL of lithium-ion batteries and in [22], where a Long-Short-Term-Memory (LSTM) network for rotatory machine remaining useful life prediction is designed.

Deep Learning architectures are also used for fault prediction, as seen in [23] and [24].

CNNs have been used in various industrial tasks, such as in tool wear classification [25], predictive maintenance [26], or in fault detection in industrial images [27].

As we have mentioned before, DL approaches present some issues when they face problems involving tabular data, which can lead to a decrease in their performance. In the past years, some works such as TabNet [28], Neural Oblivious Decision Ensembles (NODE) [29] or DNF-Net [30] have appeared trying to solve this limitations.

1D-CNN [31] models are also found in industrial solutions, such as in [32], where they are used for bearing fault diagnosis, and in [33], used for predicting the RUL of roller bearings. In [34] an evolutionary LSTM-FCN network for pattern classification in industrial processes is presented.

The existence of categorical variables is one of the challenges when using DL with tabular data, as neural networks only accept numerical values.

Therefore, transformation methods are needed to convert the data into a valid format. Label encoding converts data by mapping every category to certain numbers. One-hot-encoding or binary encoding are also used to transform categorical data.

Other transformation methods change the tabular data into an image format by means of a 2D encoding, which allows to use 2D-CNN with tabular data.

[35] propose an approach called Image Generator for Tabular Data (IGTD). The main idea of IGTD is to associate features to pixel positions, making similar features close to each other in the image.

Our work is based on the approach presented in [36] and will be explained in more detail in the next section.

3 Materials and Methods

The main objective of this work is to compare the performance of CNN approaches for tabular data against other machine learning techniques. This will be tested with two different data-sets of different industrial processes.

For this task, the results of seven ML models are compared in this paper. A simple Linear Regression model (LR) [37], tree based models such as Decision Trees (DT) [38], Random Forest (RF) [39] and Gradient Boosting (GB) [40], a Support Vector Machine model (SVM) [41], and both 1D-CNN [31] and 2D-CNN approaches [42].

For each data-set, the available data is split into train, validation and test set. 80% of the data is used for training and validating the models, and the remaining 20% is used as new data to test models' performance. Also a calibration of the hyper-parameters of each technique is conducted with the training set, using Grid-Search and Random-Search.

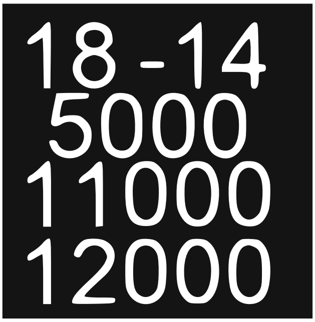
For the training of the convolutional networks, it was observed that the results using K-Fold Cross Validation were similar to those obtained with the train/test strategy, so it was decided not to use K-Fold for computational reasons. In order to find the best architecture, we have used Autokeras [43], an AutoML system based on keras, and we have tested 50 different architectures for each network.

Additionally, we have used the mean squared error as the loss function, a learning rate of 0.001, and the Adam optimizer.

As mentioned in Sect. 2, in order to work with 2D-CNN for tabular data, it is necessary to transform tabular data into a 2D format data, by means of an embedding.

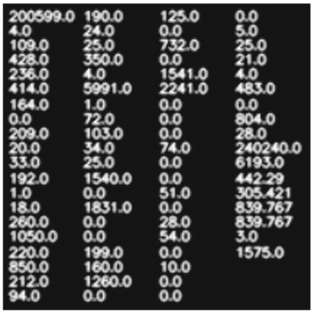
SuperTML [36] performs a two-dimensional embedding, where each instance of existing tabular data is converted into an image, by drawing each value of the features in a different area of the image, and ensuring the values occupy the largest possible part of it. After the process is finished, we have the tabular data embedded into an image format, which will be the input data for our CNN.

For both data-sets, we have generate 170 × 170 size images. Figure 1 shows the result of the embedding applied to the CVT data-set, where we have five predictive variables. Secondary metallurgy data-set have 77 predictive variables, and the result of performing the embedding can be seen in Fig. 2.



18 -14
5000
11000
12000

Fig. 1. SuperTML image for CVT data-set



200599.0	190.0	125.0	0.0
4.0	24.0	0.0	5.0
109.0	25.0	732.0	25.0
428.0	350.0	0.0	21.0
236.0	4.0	1541.0	4.0
414.0	5991.0	2241.0	483.0
164.0	1.0	0.0	0.0
0.0	72.0	0.0	804.0
209.0	103.0	0.0	28.0
20.0	34.0	74.0	240240.0
33.0	25.0	0.0	6193.0
192.0	1540.0	0.0	442.29
1.0	0.0	51.0	305.421
18.0	1831.0	0.0	839.767
260.0	0.0	28.0	839.767
1050.0	0.0	54.0	3.0
220.0	199.0	0.0	1575.0
850.0	160.0	10.0	
212.0	1260.0	0.0	
94.0	0.0	0.0	

Fig. 2. SuperTML image for secondary Metallurgy data-set

Finally, once the different models are generated, we compare the results obtained, using different error metrics, shown in Table 1.

Table 1. Evaluation metrics

Metric	Definition
Root mean squared error	$\sqrt{1/n \sum_{i=1}^{n} (y_i - \hat{y_i})^2}$
Mean absolute error	$1/n \sum_{i=1}^{n} \lvert y_i - \hat{y_i} \rvert$
R^2 Score	$1 - \frac{\sum (y_i - \hat{y_i})^2}{\sum (y_i - \bar{y})^2}$

4 Contribution

4.1 Continuous Variable Transmission (CVT)

CVT is a type of automatic transmission that can change the gear ratio to any value within its limits and according to the needs of the gear.

A CVT is made up of a roller in the cavity of the input and output discs. The transmission ratio is controlled by changing the roll inclination angle, thereby controlling the lead-in and lead-out radius.

Depending on the energy needed, the angle and the clamping force vary so that the energy coming out of the CVT is sufficient. These clamping forces generate friction in discs, which affects their "life".

The objective is to predict the efficiency of a continuous variable transmission, in order to define points of efficiency and "life" in different working conditions, which allow the transmission to be carried out efficiently.

The data-set of the CVT transmission consists of 6556 data instances with 5 predictive variables.

- Predictive variables:
  - Power: power working conditions.
  - Speed: speed working conditions.
  - Clamping force input: force applied to the transmission, responsible for pressing the input disc and rollers between them.
  - Clamping force output: force applied to the transmission, responsible for pressing the output disc and rollers between them.
  - Angle: angle of inclination of the rollers.
- Dependent variable:
  - Efficiency: efficiency of the CVT.

1D-CNN architecture consists of a convolutional layer followed by four fully-connected layers, with a dropout value of 0.5 before the layer in charge of the final prediction.

For the 2D-CNN approach, we have a deeper architecture, consisting of three convolutional layers, with max-pooling layers after the first and the second one, followed by a flatten layer and six fully-connected layers. This time, dropout value is set to 0.2 between the last two layers.

As mentioned in Sect. 3, a tuning of the optimal hyper-parameters have been conducted for every model.

4.2 Secondary Metallurgy Process

Secondary metallurgy consists of a group of processes carried out after the steel refining, by the conditioning of the liquid steel in order to obtain the required quality of the steel. Several tasks are performed during the secondary metallurgy step. Some of them are the homogenisation of the temperature and the chemical composition of the steel melt, the removal of trace elements or the improvement of cleanness [44].

After the secondary metallurgy, the obtained molten metal is transferred into various containers, where it will cool and solidify. This is called continuous casting. Finally, the generated strand is cut into predetermined lengths.

In this work, we aim to predict the quality of the casting of a secondary metallurgy process, in which molten metal is mixed with different chemicals in order to give it specific properties.

The data-set of the secondary metallurgy process consists of 2909 instances and 76 variables. During the descriptive analysis, we find the principal difficulty of the data. This process performs well in the majority of cases, which ends in a highly unbalanced data-set. This makes the prediction of low efficiency cases challenging. A description of the variables can be found in the following lines.

- Predictive variables:
  - Composition variables: a group of 12 variables which refers to the original chemical composition of the steel in the casting.
  - Overturning composition variables: a group of 12 variables of chemicals representing how much of these chemical elements exists in the steel after overturning.

  - Addition variables: 32 variables representing different chemical additions.
  - Temperature variables: group of 3 variables that measure the casting temperature and the temperature at the beginning and end of the process.
  - Gas variables: 4 variables representing different gas consumption during the process.
  - Time variables: 5 variables measuring the duration of different processes.
  - Energy variable: energy consumption in secondary metallurgy.
  - Information variables: 7 variables giving information of the castings (length, size, usages...).
- Dependent variable:
  - Efficiency: continuous variable representing the quality of the casting.

The structure of the 1D-CNN architecture consists of three convolutional layers, each one followed by max-pooling layers. After it, there is a dropout value of 0.5, followed by three fully connected layers.

For the 2D-CNN approach, we have used an existing neural network architecture named Xception [45].

5 Results and Discussion

Table 2. Results of CVT efficiency prediction

Model	MAE	$RMSE$	R^2
Linear regression	8.5e-03	4.6e-03	0.25
Decision trees	7.6e-03	3.3e-03	0.33
Random forest	6.9e-03	2.2e-03	0.44
Gradient boosting	6.2e-03	2.1e-03	0.44
SVM	5.3e-03	7.5e-03	0.42
1D-CNN	3.3e-03	**5.9e-04**	**0.82**
2D-CNN	**3.2e-03**	1.3e-03	0.66

Table 3. Results of secondary metallurgy efficiency prediction

Model	MAE	$RMSE$	R^2
Linear regression	0.77	1.06	0.24
Decision trees	0.69	1	0.31
Random forest	0.69	0.94	0.4
Gradient boosting	**0.61**	**0.88**	**0.41**
SVM	0.75	1	0.25
1D-CNN	0.69	0.98	0.34
2D-CNN	0.72	0.9	0.39

Table 4. Computational time

Model	Time
Linear regression	**0.91s**
Decision trees	76 s
Random forest	28 s
Gradient boosting	10 s
SVM	31 s
1D-CNN	200 s
2D-CNN	32.25 h

Results of the CVT efficiency prediction in Table 2, show how the traditional techniques do not obtain a good performance in terms of R^2 score, being the Random Forest the one with the best result.

Both convolutional approaches outperform the usual techniques, being the 1D-CNN approximation the one with best performance.

Regarding the values of the mean absolute error and the root mean square error, we observe how the results are good with all the techniques. However, convolutional approximations again offer better results.

Observing the results of the Secondary Metallurgy process shown in Table 3 we see that all of the approaches have a poor performance. As we mentioned before, this could be due to the complexity of the data. Among all techniques, Gradient Boosting performs better than the other ones.

Between the convolutional approaches, we see that, unlike in the CVT prediction, 2D-CNN approach performs better than 1D-CNN, nearly reaching the same performance than the Gradient Boosting.

Finally, the algorithms have also been analyzed based on their computational time during the training phase. For comparing purposes, all of them are measured using a CPU. As we see in Table 4, the 2D-CNN approach needs much more computational time and power than the rest of the techniques.

6 Conclusions and Future Work

In this work we have compared usual approaches for regression tasks in industrial processes with tabular data, and we have tested them against two convolutional approaches. This work shows promising results in the possibility of using convolutional networks with tabular data. The conclusions of the work are enumerated below.

1. Ensemble techniques such as Random Forest or Gradient Boosting are the most commonly used techniques for industrial problems with tabular data, and their performance is usually successful.
2. Both convolutional approaches can improve the performance of techniques commonly used in industrial processes.
3. Convolutional approaches achieve similar performance to state of the art techniques even in hard data-sets with unbalanced data and great number of features.
4. Both convolutional approaches perform better in the CVT data-set, where the number of variables were lower. This could show a direct relation between the complexity of the generated image and the performance of the convolutional approaches. However, more experiments will be needed to test this hypothesis.
5. Based on the results and the computational time needed by each technique, we conclude that the 2D-CNN approach is not the best option for working with the tested data-sets.

We have also identified a line of possible future work that could be investigated about SuperTML embedding. This embedding is not generalized, as it needs a modification of the coordinates and the font of the features depending on the used data-set. This makes the use of this embedding highly time consuming.

For this reason, we think that looking for a way of generalizing the embedding, no matter the used data-set would be an important step forward to this work. We are thinking about an approach based on the following ideas:

- Normalize the data and round it to the same decimals, so the size of every feature will be the same.
- Depending of the size of the image, test different font sizes and take the one that makes possible to write down all the features.
- Interactively tell the embedding the distribution of the features in the image (lines, columns).

References

1. Waibel, M., Oosthuizen, G., Du Toit, D.: Proc. Manuf. **21**, 774 (2018)
2. Oesterreich, T.D., Teuteberg, F.: Comput. Ind. **83**, 121 (2016)
3. Vaidya, S., Ambad, P., Bhosle, S.: Proc. Manuf. **20**, 233 (2018)
4. Shwartz-Ziv, R., Armon, A.: (2021). https://openreview.net/forum?id=vdgtepS1pV
5. Saufi, S.R., Ahmad, Z.A.B., Leong, M.S., Lim, M.H.: IEEE Access **7**, 122644 (2019)
6. Ye, F., Zhang, Z., Chakrabarty, K., Gu, X.: IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **32**(5), 723 (2013)
7. Jin, S., Ye, F., Zhang, Z., Chakrabarty, K., Gu, X.: IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **35**(6), 985 (2015)
8. Lin, C.C., Deng, D.J., Kuo, C.H., Chen, L.: IEEE Access **7**, 56198 (2019)
9. Lee, T., Lee, K.B., Kim, C.O.: IEEE Trans. Semicond. Manuf. **29**(4), 436 (2016)
10. Carvalho, T.P., Soares, F.A., Vita, R., Francisco, Rd.P., Basto, J.P., Alcalá, S.G.: Comput. Ind. Eng. **137**, 106024 (2019)
11. Susto, G.A., Schirru, A., Pampuri, S., McLoone, S., Beghi, A.: IEEE Trans. Industr. Inf. **11**(3), 812 (2014)
12. Kanawaday, A., Sane, A.: In: 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), pp. 87–90. IEEE (2017)
13. Paolanti, M., Romeo, L., Felicetti, A., Mancini, A., Frontoni, E., Loncarski, J.: In: 2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA), pp. 1–6. IEEE (2018)
14. Mathew, V., Toby, T., Singh, V., Rao, B.M., Kumar, M.G.: In: 2017 IEEE International Conference on Circuits and Systems (ICCS), pp. 306–311. IEEE (2017)
15. Carroll, J., Koukoura, S., McDonald, A., Charalambous, A., Weiss, S., McArthur, S.: Wind Energy **22**(3), 360 (2019)
16. Elforjani, M., Shanbr, S.: IEEE Trans. Industr. Electron. **65**(7), 5864 (2017)
17. Wu, M., Song, Z., Moon, Y.B.: J. Intell. Manuf. **30**(3), 1111 (2019)
18. Touzani, S., Granderson, J., Fernandes, S.: Energy Buildings **158**, 1533 (2018)
19. Boto, F., Murua, M., Gutierrez, T., Casado, S., Carrillo, A., Arteaga, A.: Metals **12**(2), 172 (2022)
20. Tong, Z., Miao, J., Tong, S., Lu, Y.: J. Cleaner Prod. **317**, 128265 (2021)
21. Zhang, Y., Xiong, R., He, H., Pecht, M.G.: IEEE Trans. Veh. Technol. **67**(7), 5695 (2018)

22. Zhang, H., Zhang, Q., Shao, S., Niu, T., Yang, X.: IEEE Access **8**, 132188 (2020)
23. Sohaib, M., Kim, C.H., Kim, J.M.: Sensors **17**(12), 2876 (2017)
24. Luo, B., Wang, H., Liu, H., Li, B., Peng, F.: IEEE Trans. Industr. Electron. **66**(1), 509 (2018)
25. Terrazas, G., Martínez-Arellano, G., Benardos, P., Ratchev, S.: J. Manuf. Mater. Proc. **2**(4), 72 (2018)
26. Kiangala, K.S., Wang, Z.: IEEE Access **8**, 121033 (2020)
27. Saeed, F., Paul, A., Rho, S.: In: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, pp. 280–287. Springer (2020)
28. Arık, S.O., Pfister, T.: arXiv (2020)
29. Popov, S., Morozov, S., Babenko, A.: arXiv preprint arXiv:1909.06312 (2019)
30. Abutbul, A., Elidan, G., Katzir, L., El-Yaniv, R.: arXiv preprint arXiv:2006.06465 (2020)
31. baosenguo. Kaggle-MoA 2nd Place Solution (2021). https://github.com/baosenguo/Kaggle-MoA-2nd-Place-Solution. Original-date: 2020-12-09T02:24:45Z
32. Eren, L., Ince, T., Kiranyaz, S.: J. Signal Process. Syst. **91**(2), 179 (2019)
33. Yao, D., Li, B., Liu, H., Yang, J., Jia, L.: Measurement **175**, 109166 (2021)
34. Ortego, P., Diez-Olivan, A., Del Ser, J., Veiga, F., Penalva, M., Sierra, B.: Swarm Evol. Comput. **54**, 100650 (2020)
35. Zhu, Y., et al.: Sci. Rep. **11**(1), 1 (2021)
36. Sun, B.: In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (2019)
37. Seber, G.A., Lee, A.J.: Linear Regression Analysis, vol. 329. John Wiley & Sons (2012)
38. Rokach, L., Maimon, O.: In: Data Mining and Knowledge Discovery Handbook, pp. 165–192, Springer (2005). https://doi.org/10.1007/b107408
39. Breiman, L.: Mach. Learn. **45**(1), 5 (2001)
40. Friedman, J.H.: North **1**(3), 1 (1999)
41. Cortes, C., Vapnik, V.: Mach. Learn. **20**(3), 273 (1995)
42. O'Shea, K., Nash, R.: arXiv preprint arXiv:1511.08458 (2015)
43. Jin, H., Song, Q., Hu, X.: In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1946–1956. ACM (2019)
44. Fandrich, R., Lüngen, H.B., Wuppermann, C.D.: Metall. Res. Technol. **105**(7–8), 364 (2008)
45. Chollet, F.: In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1251–1258 (2017)

Classification of Polymers Based on the Degree of Their Transparency in SWIR Spectrum

Dominik Stursa[1](✉), Dusan Kopecky[1], Jiri Rolecek[1], Petr Dolezel[1], and Bruno Baruque Zanon[2]

[1] Faculty of Electrical Engineering and Informatics, University of Pardubice, Studentská 95, 532 10 Pardubice, Czech Republic
{dominik.stursa,dusan.kopecky,jiri.rolecek,petr.dolezel}@upce.cz

[2] E. Politécnica Superior, Universidad de Burgos, Biblioteca General. Pza. Infanta Doña Elena, s/n, 09001 Burgos, Spain
bbaruque@ubu.es

Abstract. Detection, classification and sorting of polymeric particles is a common task required in recycling industry. In the proposed work, an innovative method for detection of polymeric particles and their classification is introduced. The method is based on evaluation of images of polymeric particles, obtained from short-wavelength infrared (SWIR) camera, by convolutional neural network (CNN). Compared to conventionally used spectroscopes or hyper-spectral imaging, this method utilizes single wavelength (1 050 nm) and a degree of polymer transparency serves as the main descriptor. Five different polymers (ABS, ABS-T, Nylon, PETG, PLA) in form of regular blocks (size 15 × 15 × 0.3 mm) were used in the experiment. In total 203 images (size 288 × 288 px) were prepared for CNN training and 67 for testing. Scalable ASP U-Net was tested in 6 combinations and their outputs were compared. According to used intersection over union metrics over all outputs, the topology with 64 filters and depth of 3 exhibited the best results.

1 Introduction

Constantly growing mass production of plastic products increases pressure on the quality of product recycling at the end of its lifecycle in order to save valuable resources, lessen the impact on environment and strengthen the circular economy [1]. Recycling processes must be able to process quickly and precisely high volumes of polymer waste (mostly thermoplastics) containing various additives (e.g., colorants, flame retardants, plasticizers, cross-linkers, binders) or unwanted contaminants (e.g., wood, metal, or paper particles) [2]. The quality of the recycled output strongly depends of the ways of recyclate sorting. Nowadays sorting processes involve several steps including raw separation (e.g., sieving, magnetic and triboelectric separation or flotation) and final fine removal of foreign particles from recyclate (detected by e.g., molecular and atomic spectroscopies or

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 371–382, 2023.
https://doi.org/10.1007/978-3-031-18050-7_36

X-ray fluorescence and subsequently handled by e.g., robotic arm) [3,4]. All these methods must be fast, reliable, and withstand harsh industrial environment. As a result, they are complex and very expensive.

Herein, a novel method for classification and subsequently fine and efficient removal of foreign particles from roughly separated recyclate is proposed. The method employs short wave infrared camera (SWIR) working in the near-infrared region of electromagnetic spectrum (900–1700 nm). Contrary to the commonly used hyperspectral imaging in the infrared region [5], this method utilizes only single wavelengths on which tested plastics (PLA, ABS, ABS-T, Nylon, PETG) exhibit different degree of transparency. The relative simplicity of hardware is balanced by sophisticated software which utilizes convolutional neural network (CNN) called ASP U-Net [6]. ASP U-Net is responsible for real-time decision making and its output could be subsequently used by grasping systems (e.g., robotic arms equipped by grippers) for particle removal. The proposed methods is “a proof of concept” on which a more sophisticated and robust sorting method could be build.

ASP U-net neural network has been thoroughly compared with other state-of-the-art competitors in recent Ref. [6]. Here, we propose an innovative approach to the use of this network, which takes advance of different degree of transparency of polymer particles of similar size to classify them. Indeed, we also plan to compare the performance of other competitors in this particular application for our upcoming regular research paper.

2 Problem Formulation

The aim of this work is to propose and test an innovative method for detection, classification and sorting of various polymer particles in images provided by a SWIR camera at single wavelength.

The aspect of the work is the application of a suitable detection approach by a CNN. These detection methods work on several principles. The most commonly used one is the region proposal-based method, where the visual feature extraction is used to generate the proposed regions. The principle of this method is described in Refs. [7,8].

However, when generalizing the object detection, many detection regions can occur. This higher number of suggested regions slows down the speed of the overall evaluation significantly [9]. Due to the way of region generation and classification, these methods resemble sliding window-based methods [10], which generally perform insufficiently in this task [11].

By separate object localization and classification, these methods achieve higher accuracy at the cost of passing more detection windows, which leads to slower processing. It is impossible to use these methods when localizing a moving object (e.g., on conveyor). To solve this problem, nowadays CNNs are used for image segmentation, which leads to less accurate but fast detection [12]. These methods based on image segmentation, where each pixel in the image is assigned into a class, can find an object by finding a group of pixels with the same label.

3 Method Statement

Herein, an innovative method of detection and classification of various polymers is presented. It utilizes an image segmentation using CNN. Classification of a polymer type from SWIR camera images is possible due to the different degree of transparency of polymers at a certain wavelength in infrared region. The differences in degree of transparency of each polymer block can be used for polymer classification (considering some limiting conditions as similarity of dimensions of blocks).

In order to use the properties of polymers only at a given wavelength, it is necessary to prevent the access of ambient light, e.g. by a lightproof chamber. Example of the proper arrangement for block detection is illustrated on Fig. 1.

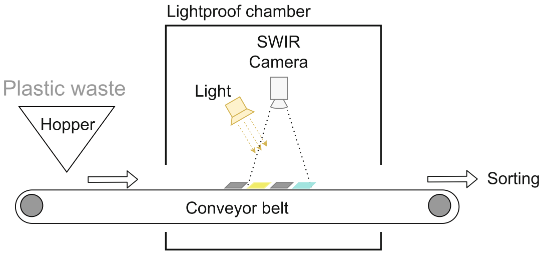


Fig. 1. Demonstration of the sorting line for polymers.

In a real situation, where polymer particles are of dissimilar size (although in narrow range due to the mechanical pre-sieving), the Lambert Beer's law of absorption will work against these settings. Therefore, a simplification in the form of a well-defined polymer height was used to validate the method.

3.1 Polymer Types

For the purpose of performing the experiments, blocks with the same thickness (0.3 mm) were made from different polymers. These blocks were printed from five types of polymeric materials by the Renkforce RF2000 3D printer. Specifically, polylactic acid (PLA); acrylonitrile butadiene styrene (ABS); nylon; polyethylene terephthalate glycol (PETG); acrylonitrile butadiene styrene with methyl methacrylate (ABS-T) were used. Images of the individual polymer blocks captured by RGB camera in the visible spectrum together with the image captured by SWIR camera are shown in Fig. 2.

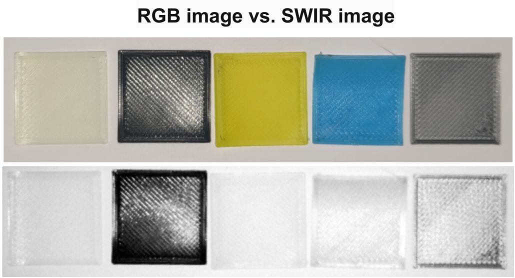


Fig. 2. Sample images of polymer blocks captured from a standard RGB camera and SWIR camera (1 050 nm); from the left: ABS-T, PETG, ABS, Nylon, PLA.

3.2 SWIR Camera and Lights

Standard SWIR cameras (e.g., Nit WiDySWIR camera [13]) operate in a short-wave infrared spectrum, which covers the range from 900–1700 nm. In addition to the SWIR sensor, a lens must be attached to the camera to ensure that the area is properly imaged and sharp. Furthermore, the SWIR camera requires either ambient light (in which the camera normally operates) or artificial illumination with light of a particular wavelength.

The possible outputs of SWIR camera are bounded with the camera type and manufacturer. Usually, cameras have a several different color schemes, e.g., “hot”, “metal” or “cold”, which are provided to describe the camera output image. Nevertheless, SWIR camera provides also a raw data format or 8-bit grayscale image as a typical sensor output for further processing.

3.2.1 Hardware Specification

Nit WiDySWIR 640G-S-DS-V3 camera together with a Computar 25 mm f2.8 lens were used to capture images of polymer blocks. The used light source was a circular light ORD80 (Smart Vision) manufactured by Smart Vision Lights operating at a wavelength of 1050 nm nm. This light source achieved the largest differences in degree of transparency among polymer blocks.

3.3 Background

As can be observed from Fig. 2, in order to distinguish the degree of transparency of each polymer type from the SWIR camera, images with a white background are not sufficient. Therefore, a mesh printed on a material that does not transmit light at a given wavelength (in this case paper) was chosen as the background.

3.4 Object Detection Method

The object detection based on the image segmentation is typically composed of two consecutive steps. First is the transformation of an input image to a segmented image, where objects are highlighted. After that, the search for same labelled pixel groups is performed.

In our case, there are some differences compared to the typical approach. The input image is transformed into several schematic images, where each image represents just single class only (polymer type) and the corresponding objects in it are marked with a gradient circle. The illustration of this process is on Fig. 3.

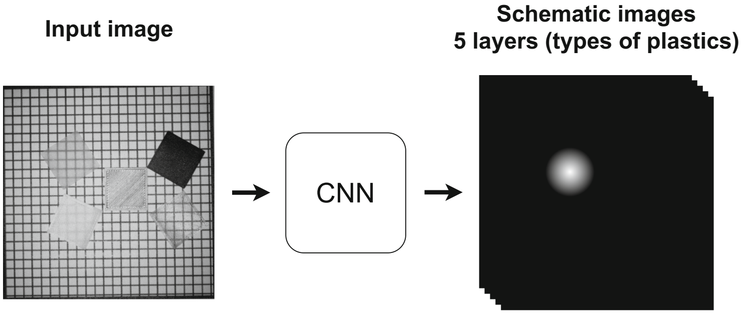


Fig. 3. Proposed object detection approach.

Based on previous experience with U-Net [14] and its modifications, the authors proposed to use an approach based on a CNN encoder-decoder that provides a well-transformed image.

3.5 CNN for Image Transformation

Many CNN architectures have been designed to solve such problems. These include, for example, SegNet [15], ResNet [16], PSPNet [17], as well as the ASP U-Net presented by the authors in previous publication [6].

ASP U-Net is based on the U-Net architecture, which was originally designed for biological and medical image segmentation tasks. U-Net is a typical encoder-decoder architecture with a bottleneck. In addition to the encoding and decoding itself, it contains skip signals from the encoder part to the decoder part at the same depth, which allow the network to propagate contextual information to higher layers.

Moreover, the topology is inspired by SqueezeNet [18], where the classical convolutional and transposed convolutional layers are replaced by modules called Down sampling module and Up sampling module. According to [19], these modules can provide more than 10× parameter reduction while maintaining the same accuracy. In addition, an attention gate [20] is implemented in the decoding part of the architecture to filter the key features propagated through the skip signals. The last improvement is the mirroring architecture, where by replicating the U-Net part, the network is expected to adjust its parameters to handle edge-relevant elements and plane-relevant elements separately.

Since the size of the extracted features naturally doubles and the image size halves with the depth of the architecture, the topology can be scalable. The scaling is then defined by the number of features in the first layer and the depth of the network. A simple layout of the architecture for feature size 16 and depth 2 is shown in the Fig. 4.

4 Experiments Procedure

In this section, the method of positioning the plastic blocks is first described along with the creation of the dataset for training the neural network. Then the process of training the ASP U-net topology is described along with its key parameters.

4.1 Hardware Conditions

Ensuring a suitable environment for scanning and observing the transparency of the individual polymer blocks was a key task, which led to the creation of a stand with a shield against external light. For the actual imaging, the SWIR

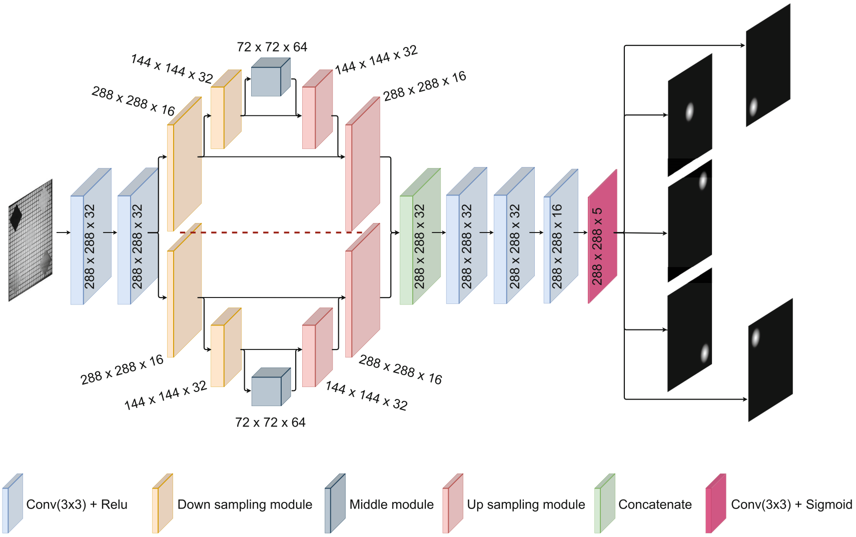


Fig. 4. Neural network topology. It all starts from the first convolutional layers containing twice as many filters as the parameter number of scalable features whose output enters the encoder. Every encoding layer then halves the width and height of its input, but doubles the feature size. Encoding is performed only to the depth defined by the scalable parameter. The opposite function is performed by a decoder layer that doubles the width and height size from its inputs and halves the feature size of previous decoding layer or bottleneck. Finally, concatenate and further processing occurs using a series of convolutional layers reducing the number of filters up to the output layer.

camera along with its lens was placed on a given stand over a defined area that was illuminated by a circular light placed at the very end of the lens. To ensure the best image quality, the focus and lens luminosity were adjusted along with a square crop of the imaged area to only the area containing the cross pattern.

4.2 Dataset Creation

For the purpose of learning and testing the neural network, a total of 270 separate SWIR camera images were taken, which included:

- 40 images containing one polymer block placed differently in the image area for each type of polymer (5×40 - totally 200 images);
- 40 images containing all polymer types one sample at a time;
- 30 images containing all types of polymers in two samples.

Depending on needs of convolutional neural network, the original 8-bit grayscale input images, captured by SWIR camera were resized to 288×288 px. After that, this group was split into a training set (203 samples) and a testing set (67 samples). Totally, 750 polymer samples were included in the images, of which 155 were included in images for testing purposes.

Next, the images had to be manually tagged. This labelling was done using a special application where the center points of each type of polymer block were marked. Subsequently, an algorithm was used to create a gradient sphere with a defined radius around the centre point. This process was used to create the target schematic images.

Examples of one input-targets pair is demonstrated in Fig. 5.

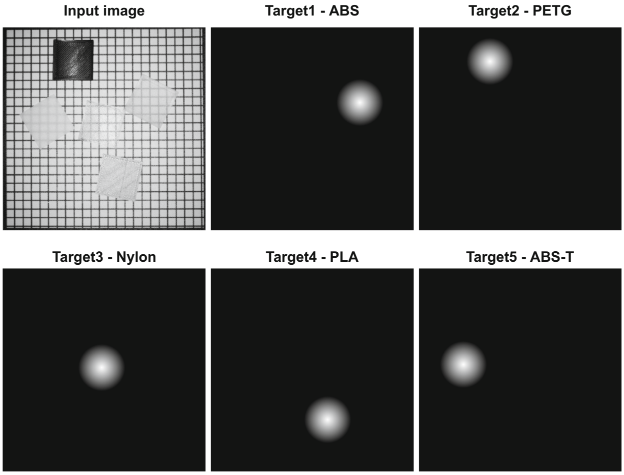


Fig. 5. Examples of input-targets pair.

Due to the size of the dataset, further augmentation was performed where each new sample was added before entering the training batch. In our case, this involved the use of brightness change (±20%), rotation (±20%°), and horizontal and vertical shift (±10%).

4.3 ASP U-Net Training

As a last step, the neural networks training was realized. A set of six scaled variants of the ASP U-Net architecture was used for training, leading to a suitable balance between accuracy and model complexity. For training purposes, all combinations of depths 3, 4 with filter counts of 16, 32, 64 were implemented.

Each architecture was trained using Adam's optimizer with initial weights randomized using a Gaussian distribution and using binary cross entropy as the loss function. For training data validation, 15% of the training set was used. Experiments were performed three times for each variant to reduce the stochastic nature of training (preventing the loss function from getting stuck in a local minimum).

The learning stopping criterion was defined by the total number of epochs, set to 500, along with the evaluation of the trend of the error function on the validation set. Throughout training, the model that achieved the best value of the loss function on the validation set was stored. Fitting the model in this way ensures selection of the best learned model that is not overtrained.

5 Results

In order to strictly evaluate performance of architectures, two metrics were used. As first the generalized intersection over union (gIoU) was performed for each image in testing set. Since five output images were available, the gIoU was calculated for each output separately. The average gIoU of all outputs for a given architecture was then calculated for the overall evaluation. This gIoU involves the predicted position dependency on the pixel intensity, which is suitable for full image evaluation. The values of gIoU over all testing images for best trained model of the tested architectures of scalable ASP U-Net is summarized in the Table 1.

Table 1. Results of gIoU evaluated for best tested topologies.

Filters	Depth	gIoU(Out 1)	gIoU(Out 2)	gIoU(Out 3)	gIoU(Out 4)	gIoU(Out 5)	Total gIoU
16	3	0.647	0.738	0.693	0.816	0.812	0.741
16	4	0.779	0.851	0.748	0.849	0.725	0.790
32	3	0.827	0.837	0.780	0.845	0.827	0.823
32	4	0.771	0.932	0.793	0.872	0.751	0.824
64	3	0.864	0.927	0.805	0.900	0.884	0.876
64	4	0.861	0.841	0.843	0.935	0.916	0.869

Best performing models were models with 64 filters, which reaches up to 0.876 of total gIoU value. The Table 1 also shows that models with fewer filters and lower depths do not have a high overlap of the ground truth with the predicted output. This is mostly due to the effect of light reflection from the surface of polymers located in the centre of the sensing zone, where sometimes the object is assigned to the wrong class. It is the place with higher reflections that cause creation of phantoms (the addition of a part of the output from its own to another class). The example covering phantoms generating in outputs together with outputs of the best performing model is shown in Fig. 6.

The overlap between predicted and ground truth bounding boxes (generated around the entire gradient shape) was thresholded at 0.75 and used as supporting information in each test image to obtain specific true positive, false positive, and false negative results in order to calculate precision and recall of all best tested models.

Based on numbers of total true positives (TP), false positives (FP), and false negatives (FN), the precision (Eq. 1) and recall (Eq. 2) was calculated for all models.

$$Precision = \frac{TP}{TP + FP}. \tag{1}$$

$$Recall = \frac{TP}{TP + FN}. \tag{2}$$

The precision expresses the ability of the model to correctly classify polymers relative to all predicted ones. The ability of a model to find all the relevant cases is called recall. It represents the percentage of true positive detection among all relevant ground truths. Both metrics for all tested models are summarized in the Table 2.

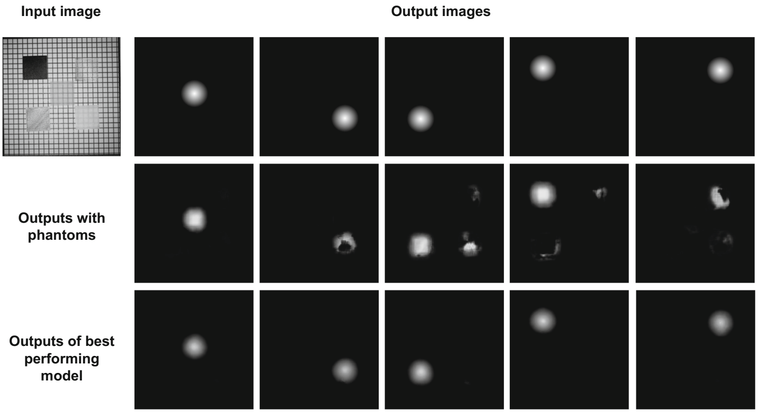


Fig. 6. Examples of outputs provided by model generating phantoms and by the best performing model.

Table 2. Results of precision and recall for best tested topologies.

Parameters	Filters	Depth	Filters	Depth	Filters	Depth	Filters	Depth	Filters	Depth	Filters	Depth
Topology	16	3	16	4	32	3	32	4	64	3	64	4
Precision	0,929		0,968		0,981		0,981		0,987		0,994	
Recall	0,831		0,842		0,873		0,895		0,965		0,965	

According to Table 2 the best model was the one with the most parameters (64 filters and depth 4), with precision 0.994 and recall 0.965. This means that the model was unable to find a single polymer sample in all classes of test images, and incorrectly found only 3 samples that were not part of the ground truth.

6 Conclusions

A solution for polymer block detection and polymer type classification using SWIR camera and light operating on a given wavelength along with the application of a sophisticated software method based on convolutional neural network has been proposed.

The proposed method was based on transformation of the original image into different schematic images each corresponding to a single class (polymer type). For the purpose of solving this problem, a neural network for semantic segmentation was used, namely scalable ASP U-Net.

Using the scalable ASP U-Net, experiments were performed for all combinations of 16, 32 and 64 numbers of filters with depths of 3 and 4 consecutive encoder and decoder modules. These tested architectures showed that with insufficient parameters, the network is more sensitive to light reflections in different places of the sensed area. In such cases, phantoms often occur along with output activation in another output layer (to wrong classification).

Surprisingly, the topology with 64 filters and a depth of 3 was the best performing, giving a total IOU equal to 0.876. The second was the topology with 64 filters and a depth of 4, which provided an IOU equal to 0.869. This result can probably be attributed to the stochastic behavior of the neural networks, where each topology was trained only three times.

One of the challenges for the future is to use ASP U-net neural network on real particles of recycled material and evaluate its performance in relation to the statistical variance of polymer particle size.

The degree of material transparency is not limited to recyclate sorting only. So another challenge is to use the approach to detect plastic moldings on conveyor belt that do not meet predetermined dimensional criteria or contrary, it can be used for height estimation of plastic moldings.

In future work, it would be interesting to observe the effect of different wavelengths on the degree of transparency of polymers in an attempt to replace expensive spectrometer solutions with a combination of cheaper SWIR cameras and lights together with sophisticated software method.

Acknowledgment. This publication is the outcome of the project SGS_2022_014 realized at the University of Pardubice, Faculty of Electrical Engineering and Informatics.

References

1. Serranti, S., Fiore, L., Bonifazi, G., Takeshima, A., Takeuchi, H., Kashiwada, S.: Microplastics characterization by hyperspectral imaging in the SWIR range, vol. 11197 (2019)
2. Bonifazi, G., Fiore, L., Gasbarrone, R., Hennebert, P., Serranti, S.: Detection of brominated plastics from e-waste by short-wave infrared spectroscopy. Recycling **6**(3), 54 (2021)
3. Bonifazi, G., Capobianco, G., Cucuzza, P., Serranti, S., Uzzo, A.: Recycling-oriented characterization of pet waste stream by SWIR hyperspectral imaging and variable selection methods. Detritus **18**, 42–49 (2022)
4. Araujo-Andrade, C., et al.: Review on the photonic techniques suitable for automatic monitoring of the composition of multi-materials wastes in view of their posterior recycling. Waste Manage. Res. **39**(5), 631–651 (2021)
5. Caballero, D., Bevilacqua, M., Amigo, J.M.: Application of hyperspectral imaging and chemometrics for classifying plastics with brominated flame retardants. J. Spectral Imaging **8** (2019)
6. Dolezel, P., Stursa, D., Kopecky, D., Jecha, J.: Memory efficient grasping point detection of nontrivial objects. IEEE Access **9**, 82130–82145 (2021)
7. Nguyen, N.-D., Do, T., Ngo, T.D., Le, D.-D.: An evaluation of deep learning methods for small object detection (2020)
8. Ju, M., Luo, H., Wang, Z., Hui, B., Chang, Z.: The application of improved yolo v3 in multi-scale target detection **9**(18) (2019)
9. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation, pp. 580–587 (2014)
10. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Mach. Intell. **32**(9), 1627–1645 (2010)
11. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection, vol. 2016-December, pp. 779–788 (2016)
12. Pi, Y., Nath, N.D., Behzadan, A.H.: Detection and semantic segmentation of disaster damage in UAV footage. J. Comput. Civil Eng. **35**(2), 04020063 (2021)
13. HDR SWIR camera: Accessed 6 May 2022. https://new-imaging-technologies.com/swir-products/widy-swir/
14. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9351, pp. 234–241 (2015)
15. Badrinarayanan, V., Kendall, A., Cipolla, R.: SegNet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(12), 2481–2495 (2017)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition, vol. 2016-December, pp. 770–778 (2016)
17. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network, vol. 2017-January, pp. 6230–6239 (2017)

18. Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K.: SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 mb model size (2016)
19. Beheshti, N., Johnsson, L.: Squeeze U-Net: a memory and energy efficient image segmentation network. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1495–1504 (2020)
20. Schlemper, J., et al.: Attention gated networks: learning to leverage salient regions in medical images. Med. Image Anal. **53**, 197–207 (2019)

Deep Learning Based Baynat Foam Classification for Headliners Manufacturing

Revanth Shankar Muthuselvam[1], Ramón Moreno[1](✉), Mario Guemes[1], Miguel Del Río Cristobal[1], Ignacio de Rodrigo Tobías[2], and Alvaro Jesús López López[2]

[1] Department of Advanced Manufacturing 4.0, Grupo Antolín, Burgos, Spain
ramon.moreno@grupoantolin.com

[2] University of Comillas, Cátedra de Industria Conectada, ICAI, Madrid, Spain

Abstract. This paper shows the performance of four deep learning algorithms on Baynat foam classification (Resnet, Mobilenet, Inception and Xception). One of the key components on headliner manufacturing is the foam. It provides acoustic isolation, lightness and robustness. Together with foam, other components are added such as textile fabrics and fiber components. Depending on the foam cell-size distribution, right amount of glue to be applied is determined correspondingly. This paper introduce AI algorithms on foam classification. The experiments are carried out using a dataset of 3000 images of foam cuts obtained from a single foam block.

1 Introduction

In headliner manufacturing unit, foam polymer [1,2] is one of the key components which provide excellent properties such as acoustic isolation, thermal insulation and more over it is light weight suitably used as a base material in headliner.

There are several providers in the market, one of them is Bayer, with their patented product Baynat polyurethane [3] which is a semi-rigid foam. As aforementioned, one of its property is thermally insulating nature and highly effective noise absorber that is dimensionally stable and resistant to high temperatures. It is used as the base material for composite sandwich headliner in vehicles. Baynat is optimized for interior usage with low organic emissions and odour for increased safety and comfort. When it comes to production, the thermoformable buckle and rupture-resistant nature of the material make headliners easy to work with and simple to install.

In short, the polyurethane is made by the combination of two components: polyol and isocyanate. In chemical terms, Polyol formulations consists of: Polyols, predominantly long and short-chained polyethers, Crosslinkers and/or Chain extenders, Surfactants, Catalysts and Water. In average, a block of foam is around 8 cubic meters depending on the region, the size and shape the cells varies. Figure 1 shows a microscopic view of two different regions of a block of foam.

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 383–390, 2023.
https://doi.org/10.1007/978-3-031-18050-7_37

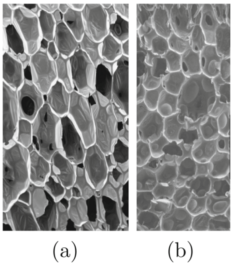


Fig. 1. On the left an example of 4 mm cells, on the right 2 mm cells

The objective of this work is to classify the foams according with the cells shape and distribution. On the block of foam, the size and shape is linearly different. On the bottom, the cells are smaller whereas the top, the cells are bigger. Figure 2 shows a couple of pictures taken from different slides of foam. Experimental r

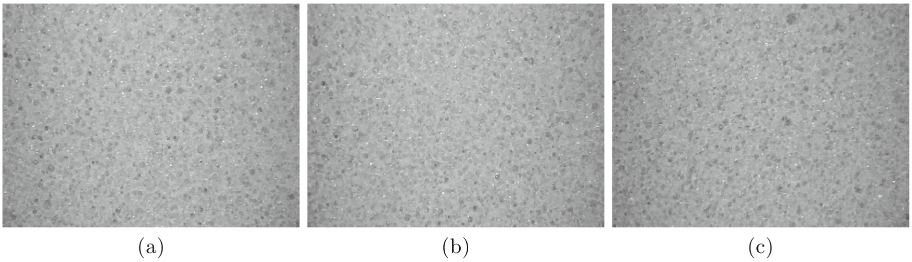


Fig. 2. Examples of images of foam. On the left (a) a sample of the bottom side, in center (b) a sample of the middle region, on the right (c) a sample of the top.

The overall manufacturing process can be summarized on Fig. 3: Starting with the raw chemical components (Polyls and so on), the foaming process requires 20 min. After that, 36 h for chemical stabilization is required. Once the foam block is fully solidified and cooled, it is ready for cutting. The outcoming foam slides are stacked, ready for the headliner manufacturing.

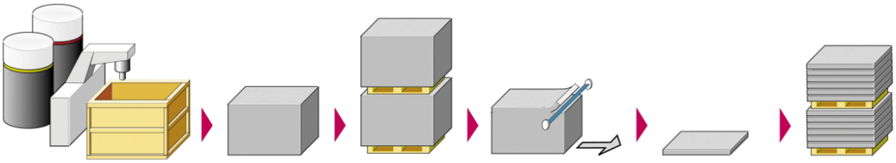


Fig. 3. Summary of foam manufacturing process

2 Deep Learning Methods

In the state of the art, there are several methods for image classification. It is well known that as bigger is the model, better learning capabilities, whereas on the other side it takes more time on inference of the details. Therefore, it is very important to find out an accurate model that learns the classes properly and inferences the images as fast as possible. In this work we have compared the following four models: Xception, Resnet, Mobilenet and Inception.

2.1 Xception

The Xception model [4] is an improved version of CNN's Inception model [5]. It has many applications on the state of the art, where it is used as image classifer, e.g. medical image processing and clothing image classification [6,7]. The model has deep convolutional layers and wider convolutional layers that work in parallel. It has two levels, where one of them has a single layer. This layer slices the output into three segments and passes it on to the next set of filters. The first level has a single convolutional level of 1 * 1 filter, while the next level has three convolutional levels of a 3 * 3 filter. The aspect that defines the Xception model is the Depthwise Separable Convolution. A general deep CNN model takes care of spatial and channel distribution, but the Xception model involves depthwise and pointwise convolution. Its architecture is designed with 71 stacked and depthwise separable convolutional layers. The model can classify hundreds of object categories and has rich representations of its utilities for a wide range of pictures.

2.2 Resnet50

Resnet50 is a model weel optimized for many platforms. The Residual Network (ResNet) [8,9] overcomes the phenomenon of vanishing or exploding gradient. ResNet's basic architecture is inspired by the VGG network [10], where the convolutional layers use 3×3 filters. The architecture for the ResNet model consists in 152 layers. Every subsequent segment follows the same pattern: a three-by-three convolution is performed with a constant dimension: 64, 128, 256, and 512. This model converges fast and generates good results. Resnet network is widely used for image classification [11,12].

2.3 MobilenetV2

As the name applied, the MobileNet model [13] is designed to be used in mobile applications, and it is TensorFlow's first mobile computer vision model. MobileNet uses depthwise separable convolutions. It significantly reduces the number of parameters when compared to the network with regular convolutions with the same depth in the nets. This results in lightweight deep neural networks. A depthwise separable convolution is made from two operations: depthwise convolution and pointwise convolution. When MobileNets Applied to Real Life The speed and power consumption of the network is proportional to the number of MACs (Multiply-Accumulates) which is a measure of the number of fused Multiplication and Addition operations. The best property of this method is the lightweight and its application on devices with low computing capabilities. Mobilenet it is widely used for image classification [14,15] where it is providing outstanding performance.

2.4 InceptionV3

Inception V3 is a well-known model, that demonstrated its wonderful performance on image recognition with the benchmarking Imagenet dataset, providing a performance up to 78.1% acccuracy. This model gathers a many ideas of several researches, based on the ideas shared by Szegedy et al. [5]. The model has several symmetric and asymmetric layers, including convolutions, pooling layers, concatenations, drop outs and fully connected layers. One of the highlighting features is the wide use of batch normalization as well as the loss function is computed by Softmax. Liwise aforementioned networks, Inception is also widely used for image classification [16,17] where it is proving very good results.

3 Experimental Results

In order to train the models we have collected a big set of images from a block of foam. The capturing procedure has been as follow.

A block of foam has been sliced in 120 layer of 5mm width. From every layer we have collected 25 images around the center of the block, it turned out in a dataset of 3000 images.

This dataset has been splited in three basic classes (bottom, middle and top), taking apart 10% of the samples for validation. Figure 4 illustrates the idea.

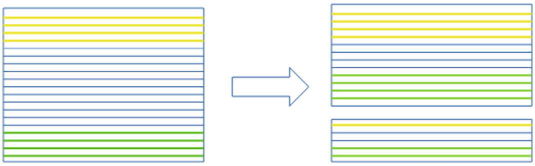
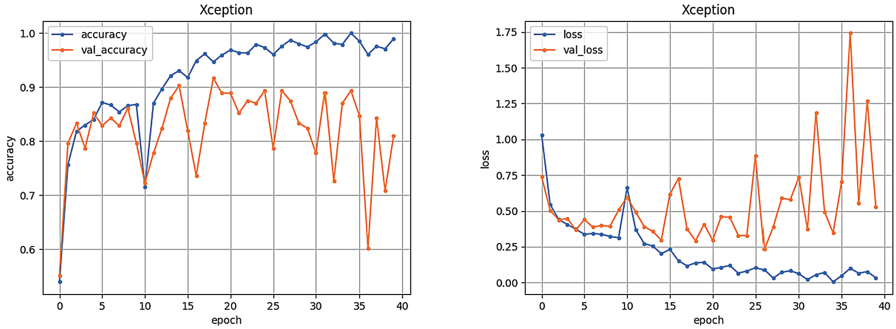


Fig. 4. Creating balanced training and testing datasets from scratch. On the left is the raw data, where yellow region is the collection of top samples, the green region is the collection of bottom samples, and the middle region in blue is the collection of middle samples. On the right are the balanced datasets, both for training and validation.

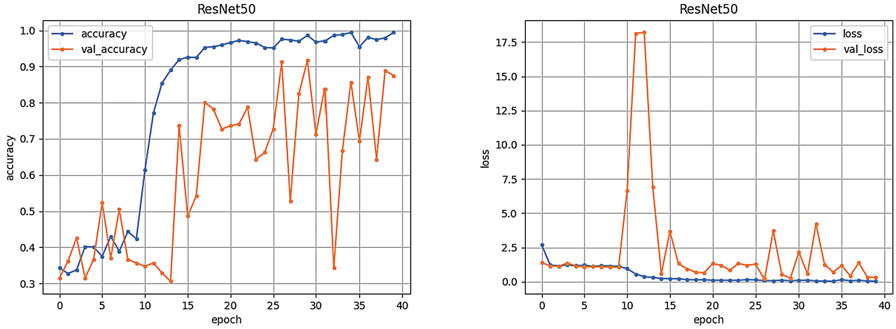
Once the datasets are ready for training, the experiments has been carreid out. Onwards, Fig. 5 shows the performance of Xception model, Fig. 6 Resnet Model, Fig. 7 Mobilenet model and Fig. 8 the Inception Model.

Xception model reaches a high accuracy after 30 epoch training whereas the loss validation is 1.75 maxima. Resnet50 model shows high accuracy after 25 epochs, whereas the validation loss in the range 2 to 4. MobilenetV2 reaches a high accuracy after 30 epochs training, with an unstable loss validation most of the time. Apart from this, we can appreciate a poor validation accuracy on training. Finally InceptionV3 reaches a good training accuracy after 30 epochs whereas the validation accuracy shows unstable performance. On the other side, loss validation is in the range 1 to 7. Finally, we can see that in average, all the models have overfitting from the 10th epoch forward.



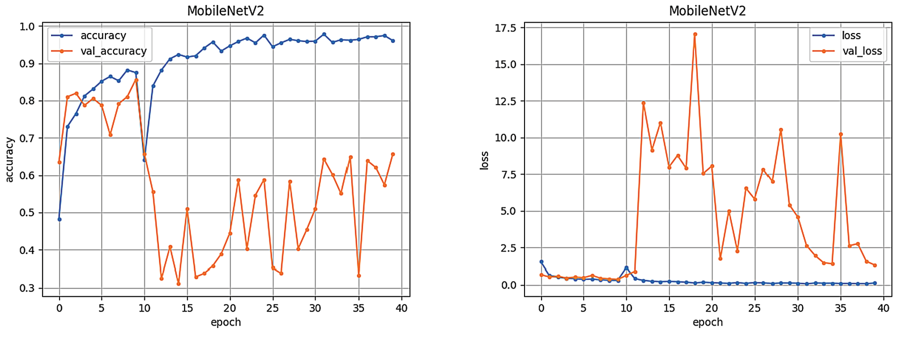
(a) (b)

Fig. 5. Xception Model. On the left the accuracy, on the right the loss. In both cases, on training and validation datasets.



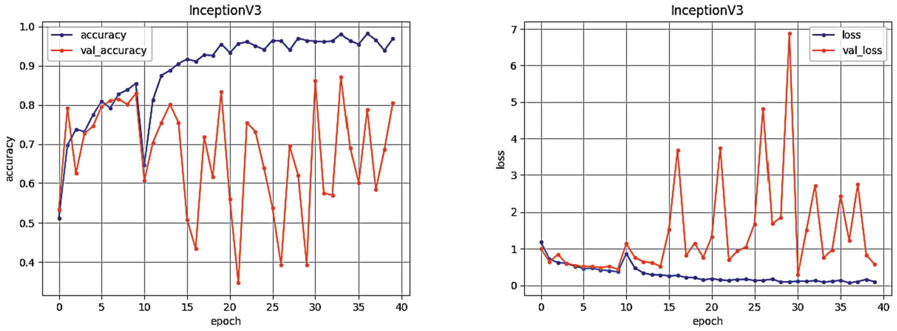
(a) (b)

Fig. 6. Resnet Model. On the left the accuracy, on the right the loss. In both cases, on training and validation datasets



(a) (b)

Fig. 7. Mobilenet Model. On the left the accuracy, on the right the loss. In both cases, on training and validation datasets.



(a) (b)

Fig. 8. Inception V3 Model. On the left the accuracy, on the right the loss. In both cases, on training and validation datasets.

4 Conclusions and Further Work

This paper has presented an industrial use case of computer vision in headliners manufacturing. At this moment, the most important issue in the project is to demostrate that it is possible to model this 3-class classification with proper accuracy. After the round of experiments explained in the experimental section, we have seen that these methods exist and show good performace, where the best one at first insight is the Xception model, which in fact is reaching the best accuracy with low overfitting.

We are aware that it is quite difficult to improve the training results of a model which already has excellent figure. In order to improve the loss on validation, we will continue with this project merging the image classification approach with other learning methodologies such as time series analysis. Our next steps we will create a stationary regression algorithm that identifies the class of the foam, having into account the last predictions. We are confident that with this approach, the validation loss will be close to zero.

References

1. Khemani, K.: Polymeric Foams: An Overview (1997)
2. Akindoyo, J.O., Beg, M.D.H., Ghazali, S., Islam, M.R., Jeyaratnam, N., Yuvaraj, A.R.: Polyurethane types, synthesis and applications a review. RSC Adv. **6**(115), 114 453–114 482, December 2016, publisher: The Royal Society of Chemistry. https://pubs.rsc.org/en/content/articlelanding/2016/ra/c6ra14525f
3. Baynat®|Polyurethane semi-rigid foams by Covestro. https://solutions.covestro.com/sk/brands/baynat
4. Chollet, F.: Xception: Deep Learning with Depthwise Separable Convolutions. arXiv:1610.02357 [cs], April 2017. arXiv: 1610.02357. http://arxiv.org/abs/1610.02357
5. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. arXiv:1512.00567 [cs], December 2015. arXiv: 1512.00567. http://arxiv.org/abs/1512.00567
6. Jinsakul, N., Tsai, C.-F., Tsai, C.-E., Wu, P.: Enhancement of deep learning in image classification performance using Xception with the swish activation function for colorectal polyp preliminary screening. Mathematics **7**(12), 1170, December 2019, number: 12 Publisher: Multidisciplinary Digital Publishing Institute. https://www.mdpi.com/2227-7390/7/12/1170
7. The clothing image classification algorithm based on the improved Xception model. Int. J. Comput. Sci. Eng. https://doi.org/10.1504/IJCSE.2020.111426
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv:1512.03385 [cs], December 2015. arXiv: 1512.03385. http://arxiv.org/abs/1512.03385
9. Goyal, P., et al.: Accurate, large Minibatch SGD: training ImageNet in 1 Hour. arXiv, Technical Report. arXiv:1706.02677, April 2018. arXiv:1706.02677 [cs] type: article. http://arxiv.org/abs/1706.02677
10. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 [cs], April 2015. arXiv: 1409.1556. http://arxiv.org/abs/1409.1556
11. Mahajan, A., Chaudhary, S.: Categorical image classification based on representational deep network (RESNET). In: 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), pp. 327–330, June 2019
12. Reddy, A.S.B., Juliet, D.S.: Transfer learning with ResNet-50 for Malaria Cell-image classification. In: 2019 International Conference on Communication and Signal Processing (ICCSP), pp. 0945–0949, April 2019
13. Howard, A.G., et al.: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, April 2017. arXiv:1704.04861 [cs]. http://arxiv.org/abs/1704.04861
14. Food Image Classification with Improved MobileNet Architecture and Data Augmentation. In: Proceedings of the 2020 The 3rd International Conference on Information Science and System (2020). https://doi.org/10.1145/3388176.3388179
15. Wang, W., Li, Y., Zou, T., Wang, X., You, J., Luo, Y.: A novel image classification approach via dense-MobileNet models. In: Mobile Information Systems, vol. 2020, p. e7602384, January 2020, publisher: Hindawi. https://www.hindawi.com/journals/misy/2020/7602384/

16. Wang, C., et al.: Pulmonary image classification based on Inception-v3 transfer learning model. IEEE Access **7**, 146 533–146 541, conference Name: IEEE Access (2019)
17. Xia, X., Xu, C., Nan, B.: Inception-v3 for flower classification. In: 2017 2nd International Conference on Image, Vision and Computing (ICIVC), pp. 783–787, June 2017

Special Session on Time Series Forecasting in Industrial and Environmental Applications

A GAN Approach for Anomaly Detection in Spacecraft Telemetries

Carlo Ciancarelli[1], Giorgio De Magistris[2(✉)], Salvatore Cognetta[3], Daniele Appetito[3], Christian Napoli[2], and Daniele Nardi[2]

[1] Thales Alenia Space Italia S.p.A, Via Saccomuro, 24, Rome 00131, Italy
carlo.ciancarelli@thalesaleniaspace.com

[2] Department of Computer, Automation and Management Engineering,
Sapienza University of Rome, Via Ariosto, 25, Rome 00185, Italy
{demagistris,cnapoli,nardi}@diag.uniroma1.it

[3] Sapienza University of Rome, Via Ariosto, 25, Rome 00185, Italy
{cognetta.1874383,appetito.1916560}@studenti.uniroma1.it

Abstract. In spacecraft health management a large number of time series is acquired and used for on-board units surveillance and for historical data analysis. The early detection of abnormal behaviors in telemetry data can prevent failures in the spacecraft equipment. In this paper we present an advanced monitoring system that was carried out in partnership with Thales Alenia Space Italia S.p.A, a leading industry in the field of spacecraft manufacturing. In particular, we developed an anomaly detection algorithm based on Generative Adversarial Networks, that thanks to their ability to model arbitrary distributions in high dimensional spaces, allow to capture complex anomalies avoiding the burden of hand crafted feature extraction. We applied this method to detect anomalies in telemetry data collected from a simulator of a Low Earth Orbit satellite. One of the strengths of the proposed approach is that it does not require any previous knowledge on the signal. This is particular useful in the context of anomaly detection where we do not have a model of the anomaly. Hence the only assumption we made is that an anomaly is a pattern that lives in a lower probability region of the data space.

1 Introduction

Satellites in orbit are monitored by a network of sensors which produce a huge stream of telemetry data. When the amount of data is huge and it needs to be processed in very short time, a solution to extract meaningful information (that in our case consists in anomalous patterns) must be based on fully automatized processes. Anomaly detection in time series data is a well studied problem in both the data mining and machine learning communities. Since it always happens that normal data samples outnumber the anomalous ones, anomaly detection is considered a semi-supervised (where anomalous data is used only for testing) or unsupervised (with no information about normal or anomalous data) problem. An exhaustive review of the most common approaches can be found in [1]. In

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 393–402, 2023.
https://doi.org/10.1007/978-3-031-18050-7_38

particular a general approach consists in transforming the sequences into a feature space and then use a point anomaly detection technique in the new space to detect anomalies. However this approach depends both on the anomaly detection technique and the properties of the feature space. For example clustering based methods [10,13,19] require that the anomalies do not aggregate into clusters; nearest neighbour and density based methods [6] require that the anomalies do not form dense regions in the feature space; spectral methods [3,7] assume that a projection into a different space exists such that normal and anomalous points can be clearly distinguished. Another approach consists in training a model to predict the signal in the future and then compare the predicted and observed signals to detect anomalies, like in [14]. In this paper we propose the application of the Generative Adversarial Networks (GANs) for the anomaly detection in spacecraft telemetry data. The GANs provide a deep latent representation of data that can be used directly for the assessment. In particular they implicitly extract meaningful features that can be exploited to discriminate between normal and abnormal samples through the assignment of an anomaly score. The rest of the paper is structured as follows: Sect. 2 briefly describes the GANs framework focusing on its application on anomaly detection; Sect. 3 reviews the approaches used so far to tackle the problem of anomaly detection in spacecraft telemetries; Sect. 4 describes our dataset while Sect. 5 illustrates our method along with all the implementation details; Sect. 6 discusses the results and the conclusion is drawn in Sect. 7.

2 GANs and Anomaly Detection

The GANs Framework, firstly introduced in [5], is composed of two networks: a generator and a discriminator, often referred as G and D. The generator learns a mapping from the latent space Θ_z, usually the set of k-dimensional standard normal vectors, to the data space Θ_{data} and the discriminator learns a categorical probability distribution over the generated and real samples to discriminate between real and fake samples. G and D optimize the same criterion in opposite directions, following a two player minimax game. The general idea behind the application of GANs to anomaly detection consists in using the output of the discriminator and the reconstruction error to assign an anomaly score to a data sample x. The output of the discriminator is the probability that a sample is "real", hence its inverse can be directly interpreted as an anomaly score. The second part is less obvious and its justification relies on the structure of the latent space. It was shown [17] that the space learned by the generator has smooth transitions, because walking on the learned manifold results in semantic changes to the generated image. This encouraged the usage of the GANs framework as an unsupervised features extractor through an inverse mapping from data space to the latent space. The reconstruction error is the distance between a data point x and its reconstruction $G(z_\gamma)$ where z_γ is the inverse mapping of x into the latent space.

3 Related Works

Still nowadays the most widely used approaches for anomaly detection in spacecraft telemetries are based on simple Out of Limit (OOL) checks [11], meaning that, when the signal exceeded some predefined upper and lower bounds, an alarm is triggered. More advanced solutions introduced clustering techniques on multidimensional vectors obtained by manually extracted features, like the *Inductive System Health Monitoring* developed at NASA [9] and the *Automated Telemetry Health Monitoring System* (ATHMoS) developed at the German Space Operation Center (GSOC) [15]. Recent works propose the introduction of deep learning. For example [16] proposes the introduction of deep autoencoders to extend the manually extracted features, [12] proposes a deep autoencoder to model the normal behavior of the telemetries and a thresholding technique on the reconstruction errors to detect anomalies and it suggests the introduction of a recurrent architecture to take into account the temporal evolution. In this direction [8] proposes a Long Short-Term Memory (LSTM) recurrent network to predict the future signal under normal conditions, then, at test time, the predicted values are compared with the observed values and anomalies are computed using thresholding techniques. In this paper we propose the application of the GANs framework for the spacecraft telemetries anomaly detection. The GANs have already been used for finding anomalies in complex data. In particular [18] introduced the AnoGAN architecture for the marker discovery in tomography images of the retina. A more efficient version of AnoGAN called Efficient GAN-Based Anomaly Detection (EGBAD) was introduced in [20]. It uses the Bidirectional GANs architecture [4] to learn at training time an inverse mapping from data space to latent space. In this paper we present our implementation of the EGBAD framework along with an analysis of the anomaly scores that employs a histogram for the detection and the temporal evolution of the anomaly score for the localization of the anomaly.

4 Data

We have at our disposal simulated data that emulate the operation of a LEO (Low Earth Orbit) satellite. In particular we studied the sensors monitoring a *Reaction Wheel* (RW), which is a type of flywheel used primarily by spacecrafts for three-axis attitude control. The RW has a high pointing accuracy and it is particularly useful when the spacecraft must be rotated by very small amounts, for example for keeping a telescope pointed at a star. The Reaction Wheel was equipped with four sensors monitoring: the current absorbed, the temperature, the velocity and the commanded torque (see Fig. 1). TAS-Italia industry[1] provided us the simulation of four months of observation data, one of those with an anomalous behavior.

[1] https://www.thalesgroup.com/it/global/activities/space.

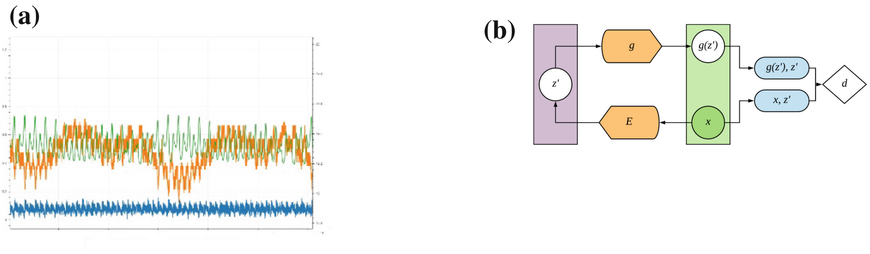


Fig. 1. (a) A sample of telemetry data. Each line corresponds to a sensor: motor current (blue line), temperature (orange line) and angular speed (green line). (b) Graphical representation of the Bi- GAN architecture. g and d are respectively the generator and the discriminator, while E is the encoder (the novel component that characterizes the Bi- GAN model). x is a data sample from the data distribution and z0 is a sample from the known latent distribution.

5 Method

In this paper we propose an implementation of the EGBAD framework (represented in Fig. 1b). In particular we split the stream of telemetries into fixed length sequences and implemented both the Generator, the Encoder and the Discriminator as multilayer perceptrons. The networks are trained adversarially with the original BiGAN loss. The generator G learns the mapping from samples from the latent distribution $p_z(z)$ to samples from the data distribution $p_x(x)$ while the encoder E learns the inverse mapping. The discriminator D discriminates jointly in data and latent space. Even though it is not explicit, the Encoder and the Generator are proven to be one the inverse of the other at the optimum [4]. The three networks were implemented as reported in Table 1. Both in the univariate and multivariate case we split the input into fixed length windows. The input shape of the Encoder changes in the univariate and multivariate cases. In the former it equals the sequence length while in the latter it is the sequence length multiplied by four, that is the number of channels, since the input is flattened.

We trained the model with Adam optimizer with learning rate of $1e-5$ and betas equal to 0.5. We used a batch size of 512 samples, while the latent dimension was equal to 32. All the weights and biases of the *encoder* and *discriminator* layers are initialized with the $Xavier$ initializer, while the weights and biases of the *generator* are initialized with the He initializer. We trained the network for 5 epochs for each experiments.

Once the model is trained, the anomaly score $A(x)$ is computed as a convex combination of the reconstruction loss L_G and the discriminator-based loss L_D:

$$A(x) = \alpha L_G(x) + (1 - \alpha) L_D(x) \tag{1}$$

where L_G and L_D are defined as follows:

$$L_G(x) = |x - G(E(x))|_1 \tag{2}$$

$$L_D(x) = |f_D(x, E(x)) - f_D(G(E(x)), E(x))|_1 \tag{3}$$

Table 1. BiGAN architecture.

Network	Layer	Units	Non linearity	Dropout
$E(x)$	Dense	64	Leaky ReLU	0.0
	Dense	64	Linear	0.0
$G(x)$	Dense	64	ReLU	0.0
	Dense	128	ReLU	0.0
	Dense	121	Linear	0.0
$D(x)$	Dense	128	Leaky ReLU	0.2
$D(z)$	Dense	128	Leaky ReLU	0.2
$D(x, z)$	Dense	128	Leaky ReLU	0.2
	Dense	1	Linear	0.0

In particular the L_D term is called feature matching loss [21] and f_D is the output of the discriminator layer that precedes the final classification layer. The anomaly score $A(x)$ is not bounded, hence, in order to be interpreted, it must be compared with some reference values, that in our case are the anomalies scores computed on the normal data.

6 Results

The proposed method was introduced to solve a complex anomaly detection task. Figure 2 shows some samples taken from the normal data distribution and from the anomalous data distribution. Also from a human analysis the anomaly is not easily identifiable. In such a complex scenario, approaches based on clustering and k nearest neighbour, that are the most frequently mentioned in literature [2], failed to identify the known anomaly. Figure 3 shows the principal components of some points sampled from the normal and the anomalous distributions and it shows that normal and anomalous points are not separable in the feature space. Despite points in Fig. 3 are just represented with the two principal components we also tried to cluster normal versus anomalous points using more than two principal components, but the results remained almost unchanged.

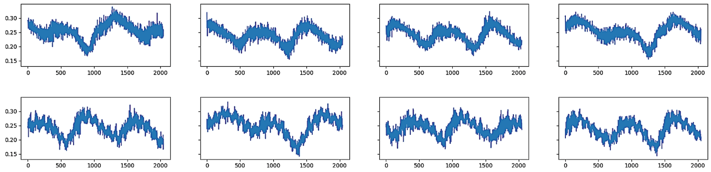


Fig. 2. Sequences sampled from the current sensor, in particular the sequences in the first row are sampled from the normal distribution, while the ones in the second row comes from the anomalous distribution.

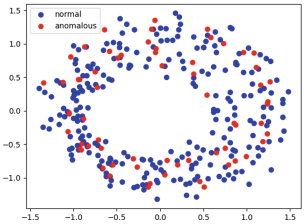


Fig. 3. First two principal components of the data samples (from the current sensor). The blue and red points represent respectively the normal and anomalous data points.

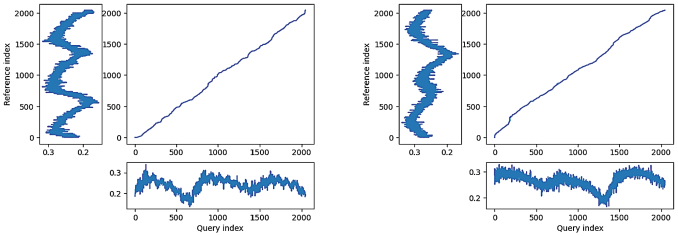
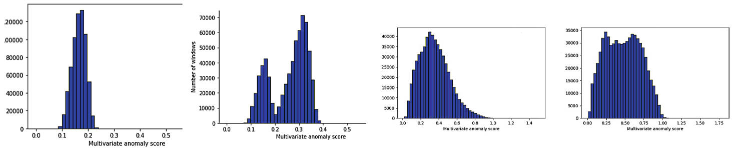


Fig. 4. Warping curve between the query sample (at the bottom of the curve) and its closes neighbours in the normal samples (at the left of the curve). In the left figure the query is a normal sample, while in the right figure the query is an anomalous sample. In both cases the query refers to the current sensor.

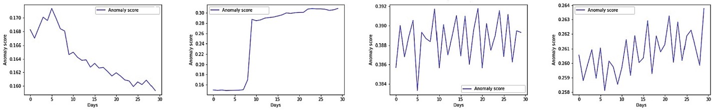
We also tried density estimation with Dynamic Time Warp (DTW): we fixed a threshold for the DTW distance and for each sample we counted the number of normal neighbours (with the DTW below the threshold). We tried different thresholds but the expected number of neighbours was approximately the same both for normal and anomalous samples. Figure 4 shows the 1-NN (taken from the normal samples) of a normal and an anomalous sample and the respective warping curve. We can see that they are very similar and the anomalous vs normal nearest neighbour warping curve is actually closer to a straight line with respect to the normal vs normal nearest neighbour warping curve. The failures of the classical approaches led us to the proposed method based on Generative Adversarial Networks, which results will be detailed in the rest of this section. In particular we carried both a univariate and multivariate analysis. In the former case the network is trained independently on each sensor, therefore the correlation between sensors is not considered. In the latter the network is trained once for all sensors, hence the input of the network is a multivariate time series in which each channel corresponds to a particular sensor. In both configurations we analyzed the anomaly scores varying the temporal granularity. We compared two observation periods that were not used for training. The first one, referred in the following as *Normal Period*, contains only normal data while the second, referred as *Anomalous Period* contains a known anomaly. For the sake of clarity let us specify that not every sample in the anomalous period is affected by an



(a) (b) (c) (d)

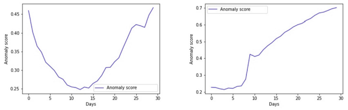
Fig. 5. Histograms counting the frequencies of the anomaly scores in a given observation period. In particular the first column (Figs. 5a and 5c) refers to the period without anomalies while the second column (Figs. 5b and 5d) to the one with the known anomaly. The First row (Figs. 5a and 5b) shows the histograms for the univariate case and in particular they refer to the current sensor, while the second row (Figs. 5c and 5d) illustrates the histogram for the multivariate case.

anomaly but rather there is just a single anomaly and its time localization is known. First we plotted an histogram of the anomaly score for each period: the range of the anomaly score is split into bins on the horizontal axis and the bars on the vertical axis are proportional to the frequency of each bin. The histograms of the two periods for the univariate and multivariate case are depicted in Fig. 5. It is important to notice that the anomaly is detected by both the univariate and multivariate approaches even though the plots are significantly different. In particular in the univariate case (top row in the Fig. 5) the curves outlined by the histograms have the shape of a unimodal distribution for the normal period and a bimodal distribution for the anomalous period. The peaks correspond to the anomaly scores with higher frequency. We can observe that the smaller peak in the anomalous period and the one in the normal period correspond to the same amount of anomaly score, hence they can be interpreted as the normal samples, while the bigger peak in the anomalous period matches a larger anomaly score, therefore it is probably caused by the anomalous sequences. The picture is different for the multivariate case in which we have unimodal distributions in both the normal and anomalous observation period. However the latter has higher mean and standard deviation, which are both indications of abnormal behavior. Although these graphs allow to individualize an anomaly they do not allow to individualize the instant, within the observation period, in which the anomaly has occurred. At this purpose we also plotted the evolution of the anomaly score with respect to time for both the normal and anomalous periods. It worth noting that the plots corresponding to different sensors do not always agree. For example Fig. 6 represents the evolution of the anomaly score for the current and speed sensors. In the former plot we can see a pulse that can be associated with the known anomaly while the latter is very noisy and it does not allow to identify an anomalous event. In the multivariate case (Fig. 7) the plot is more noisy, since some sensors were affected by the anomaly while others do not. Also in this case however there is a sudden increment in the anomaly score that corresponds to the known anomaly.



(a) (b) (c) (d)

Fig. 6. The figures show the temporal evolution of the anomaly score in different observation periods and different sensors. In particular Figs. 6a and 6b refers to the current sensor while Figs. 6c and 6d to the speed sensor. Figures 6a and 6c refers to the normal period while Figs. 6b and 6d to the anomalous one.



(a) (b)

Fig. 7. The figures show the temporal evolution of the anomaly score in the multivariate case for the normal (Fig. 7a) and anomalous (Fig. 7b) periods.

7 Conclusion

In this paper we applied the Generative Adversarial Networks, and in particular the BiGAN architecture, to the problem of anomaly detection in spacecraft telemetry data and compared the results with those of some classical algorithms. In our experiment the proposed method was the only one capable of identifying the known anomaly. Moreover the complexity of the model with respect to the dataset is drastically reduce with respect to methods based on KNN, since once the model is trained, the execution is very fast and do not depends on the size of the dataset. From the univariate and multivariate analisys emerged that there is no strong correlation between sensors and in particular some sensors seem not to be affected by the anomaly, while others showed a sudden increment of the anomaly score when the anomaly is happening. This fact is positive from an implementation point of view because the model scales well with the input. In particular we considered only four sensors connected to a reaction wheel but the system could be easily expanded for the monitoring of all the sensors in the spacecraft. This is not the case in the multivariate case, because the number of parameters is quadratic with respect to the input size. It could be the case however that the identification of more complex anomalies could depend on the complex relations between different sensor readings. At this purpose it would be interesting for a future work to test the model with more anomalies and different configurations of the input sources.

References

1. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. In: ACM Comput. Surv. **41**(3) (2009), pp. 1–72. https://doi.org/10.1145/1541880.1541882. ISSN: 0360–0300
2. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. ACM Comput. Surv. (CSUR) **41**(3), 1–58 (2009)
3. Ding, M., Tian, H.: PCA-based network traffic anomaly detection. Tsinghua Sci. Technol. **21**(5), 500–509 (2016)
4. Donahue, J., Krähenbühl, P., Darrell, T.: Adversarial feature learning. arXiv preprint arXiv:1605.09782 (2016)
5. Goodfellow, I., et al.: Generative adversarial networks. Commun. ACM **63**(11), 139–144 (2020). https://doi.org/10.1145/3422622 ISSN: 0001–0782
6. Gu, X., Akoglu, L., Rinaldo, A.: Statistical analysis of nearest neighbor methods for anomaly detection. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
7. Huang, L., et al.: In-network PCA and anomaly detection. In: Advances in Neural Information Processing Systems, vol. 19 (2006)
8. Hundman, K., et al.: Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In: Proceedings of the 24th ACMSIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 387–395 (2018)
9. Iverson, D.L.: Inductive System Health Monitoring. In: IC-AI, pp. 605–611 (2004)
10. Kiss, I., et al.: Data clustering-based anomaly detection in industrial control systems. In: 2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP), pp. 275–281. IEEE (2014)
11. Martinez, J.: New telemetry monitoring paradigm with novelty detection. In: SpaceOps 2012, p. 1275123 (2012)
12. Martinez, J., Donati, A.: Novelty Detection with Deep Learning. In: 2018 SpaceOps Conference, p. 2560 (2018)
13. Münz, G., Li, S., Carle, G.: Traffic anomaly detection using k-means clustering. In: GI/ITG Workshop MMBnet, vol. 7, p. 9 (2007)
14. Napoli, C., et al.: Exploiting wavelet recurrent neural networks for satellite telemetry data modeling, prediction and control. In: Expert Systems with Applications, p. 117831 (2022)
15. OMeara, C., Schlag, L., Wickler, M.: Applications of deep learning neural networks to satellite telemetry monitoring. In: 2018 SpaceOps Conference, p. 2558 (2018)
16. OMeara, C., et al.: ATHMoS: automated telemetry health monitoring system at GSOC using outlier detection and supervised machine learning. In: 14th International Conference on Space Operations, p. 2347 (2016)
17. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
18. Schlegl, T., et al.: Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. arXiv: 1703.05921 [cs.CV] (2017)
19. Syarif, I., Prugel-Bennett, A., Wills, G.: Unsupervised clustering approach for network anomaly detection. In: Benlamri, R. (ed.) NDT 2012. CCIS, vol. 293, pp. 135–145. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30507-8_13

20. Zenati, H., et al.: Efficient gan-based anomaly detection. arXiv preprint arXiv:1802.06222 (2018)
21. Zhang, Y., et al.: Adversarial feature matching for text generation. In: International Conference on Machine Learning, pp. 4006–4015, PMLR (2017)

Management and Forecasting of the Demand for Caskets in the Funeral Sector. Study Before and During the Covid-19 Pandemic

Cristina Martínez González[1], Athénaïs Sauvée[2], Santiago Porras Alfonso[1](✉), and Julio César Puche Regaliza[1]

[1] Department of Applied Economics, University of Burgos, Plaza Infanta Doña Elena, s/n, 09001 Burgos, Burgos, Spain
sporras@ubu.es

[2] Department of Private Law, University of Burgos, Plaza Infanta Doña Elena, s/n, 09001 Burgos, Burgos, Spain

Abstract. Businesses usually categorize their products according to an inventory plan in order to minimize storage costs. In this regard, the specific local funeral service company selected for the study needs new methods to reduce costs due to strong fluctuations in demand. The prime aim of this paper is to determine the most accurate model to forecast future values in customer demand. A demand time series approach and statistical, linear and neural network methods are used for prediction. The study is divided in two scenarios, the first one presents a normal situation in which the techniques applied can predict the demand with an acceptable error. The second one is the period during the Covid 19, in this case, the predictions are more imprecise due to the large spikes in demand that occurred during the pandemic and that could not be predicted by any method.

Keywords: Funeral service industry · Covid-19 · Customer demand prediction · Time series analysis · Forecasting

1 Introduction

Nowadays, forecasting techniques are applied to numerous fields of the economic world such as sales, warehouse management, identification of customer profiles, market research, stock portfolio analysis and much more. However, the economic sectors and the activities of each company are very varied. This paper presents an analysis in order to predict sales in a very particular sector, the funeral sector. Specifically, in a funeral company in Burgos, Spain. The objective is to make a prediction as accurate as possible of the demand for coffins in the following months. This is important since it allows optimal management of warehouses and the programming of orders to suppliers. To carry out this objective, the starting

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 403–412, 2023.
https://doi.org/10.1007/978-3-031-18050-7_39

data will be the monthly sales of previous years. The data will be treated as a time series, first they will be studied to see the type of demand in this sector, and then the prediction techniques will be applied to evaluate which is the best. The particularities of the sector must be considered since it is very difficult to know the demand a priori. Two scenarios will be considered, one pre-Covid 19 and the other during the pandemic to see the behavior of the techniques used in the face of unpredictable demand and in which there was a lack of supplies. Forecasting caskets sales in funeral sector, has never been studied previously. However, there are other articles about this sector within the field of marketing [3] or business [2] among others.

2 The Funeral Services Industry

It is important for the study to clarify the specific scheme in which the research is going to be tested. The prime reason to select this industry relies on the fact that usually, the service business does not have adequate inventory management policies implemented [9]. The market of the funeral services sector has experienced a great transformation in the past decades, becoming more liberalized and more transparent. And it is formed by much more professional and technical businesses. The sector has evolved to try to break this gap between some logistic areas such as inventory management, accrual accounting and control that must be followed especially by small and medium businesses. Even if funeral homes are included in a service sector, these businesses must put inventory control policies into effect as they have to manage the number of caskets sold in a certain period, for instance. In recent years, the sector has implemented modern systems in terms of their logistics and the way the businesses are run with the aim to provide a whole integrated service to the society while being economically efficient. The industry is trying to break the distance between this particular kind of business and society by searching and deploying new initiatives. This transformation has become much more important as a consequence of the demand fluctuations along the past years and especially over the year 2020 with the global pandemic situation.

The number of funeral services in a year is dependent on the mortality rate in the country and therefore the funeral service sector has a relatively constant demand, except for unusual situations such as the global pandemic of Covid-19 in 2020.

2.1 Funeral Services Sector in Data

In 2019 in Spain there were 417,625 deaths, approximately 2.4% less than the year 2018, which was the year with more deaths according to official data provided by national institutions until the global pandemic situation arose, as represented in Fig. 1. Moreover, 2020 is now the year with more fatalities since there is official data, approximately 20.4% more than the previous year [1].

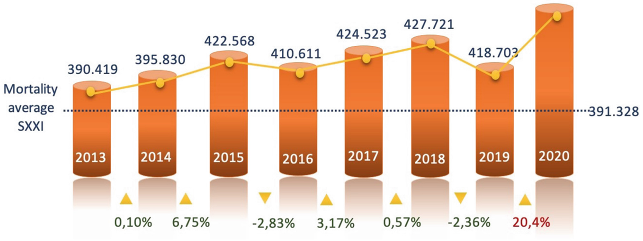
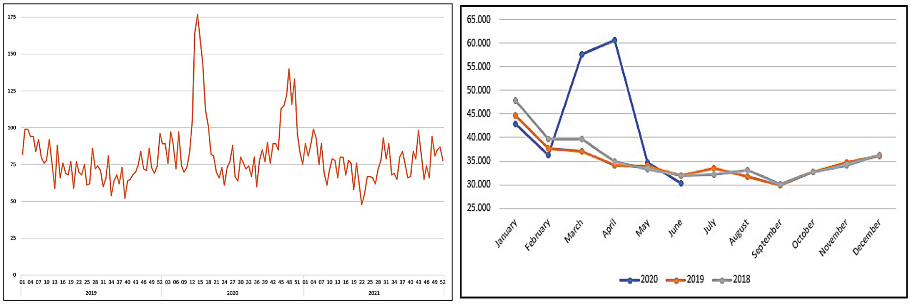


Fig. 1. Mortality evolution in Spain 2013–2020

Turning to the local scheme, in Fig. 2(a), the lector can contemplate a time series representation of the total number of fatalities in Burgos disclosed by weeks in a specific frame time from 2019 till 2021. As it can be clearly seen there is a huge spike during the weeks of March, April and May of 2020 due to the Covid-19 situation.



(a) Burgos by weeks (2019-2021)

(b) Spain by month (January 2018- June 2020)

Fig. 2. Fatalities time series

According to the official data provided by the National Institute of Statistics in Spain, the increase of fatalities during the first semester of 2020 due to Covid-19 was 44651. In other words, there was more than a 20% increase in fatalities in comparison with the same period of the previous year, as represented in Fig. 2(b).

3 Dataset Description

The analysis is going to be performed in two main products of the same products category in the company -lower and medium classes of coffins-. The specific time frame to conduct the analysis is from January 2016 until March 2021 with a monthly frequency and it is executed in two main schemes, -a pre-Covid 19 scenario and a Covid 19 context-. The objective is to find out the best model to forecast the demand of the lower and medium coffins in the short-term phase in order to design adequate supply policies and a proper control and management of the stock materials and inventory assets [13]. The company works with other kinds of coffins, children's, high-end and premium range. But the first two have very low demand and the last one is only served on request, which is why they are excluded from the study since they do not have a great impact on inventory management.

4 Methodology

The data is treated as a time series, the analysis is conducted by using some techniques like lagged predictors [6] and rolling forecasting [15]. This approach has been tested successfully in other works like in [14]. For this research, the database has been spliced into two categories; one specific time frame to be used as a training set to prove the different models; and another specific time frame used as a test to check model's accuracy in forecasting. The problem is constructed around two main time frame scenarios, one pre-Covid 19 and another during Covid 19. For the first scheme, the original data set to perform the test is the information from the caskets sales of the year 2019, while the parallel information of the years 2016–2018 is used to train the data. For the second scenario, the methodology is similar, but the data set used as the test is the information from the year 2020 and the train data set from the years 2016–2018. The general objective of the study is to find the best model which adjusts the data to forecast future value. To evaluate the reliability and accuracy of the models, their performances are going to be compared using Mean Absolute Percentage Error (MAPE) and Relative Absolute Error (RAE).

The specific software selected to conduct the analysis are the R Caret package [10] and the R Forecast package [7]. Both of them provide different techniques for forecasting.

To analyze the performance of several techniques we have selected three different approaches. Firstly, a group of models based on statistics such us Seasonal Naive, Arima, or TBATS; secondly, some linear models such as Linear Regression, Boosted Generalized Linear Model, Support Vector Machines with Linear Kernel model and Generalized Linear model with elastic net; and thirdly, the data is going to be process by neural networks - NNAR.

The Autoregressive Integrated Moving Average Model - Arima
Is a statistical analysis model quite used in forecasting future values related to internal consumption [8]. This model is one of the most general and precise ones for time series prediction analysis. On this occasion, the model is aimed at predicting future values by analyzing the differences between original values of the time series. For that reason, this model is based on regression analysis of the original values.

Autoregressive Neural Network Model - NNAR
Neural networks are "universal approximators" [17] used to model temporal series data as inputs and successfully applied in short term predictions. This

specific model is known as neural network autoregression or NNAR model. In this paper, NNAR model uses the historical data provided by the company as inputs to forecast future outputs. This model forecasts future values by establishing relationships between historical sequences of independent values.

Seasonal Naïve Model - Snaive
This specific model is widely used in time series estimations with seasonal data. This model allows to make forecasts by analyzing specific observed values of the same time period of the year and, on this basis, it predicts future ones [16]. In other words, the forecast of future values of any specific month is equal to the last observed values of that month of past years. In a simplified manner, it forecasts future periods based on past ones.

TBATS Model
Is one of the most advanced models for accounting seasonality in time series analysis which allows the user to create long term predictions. Its nomenclature states for the main key characteristics of the model - "Trigonometric seasonality, Box-Cox transformation, ARMA errors, Trend and Seasonal components" [4]. It is adequate for dealing with complicated seasonal time series data changing over time.

Linear Regression model
Is a tool used to forecast future values on the basis of past ones. It establishes relationships between the original values and the predicted ones by fitting a linear equation. In other words, it determines the underlying trend in the data set by establishing a linear relationship between the original values and the predicted ones [6].

Boosted Generalized Linear Model - BGLM
This model optimizes forecast accuracy by establishing relationships between the original values and the predictive ones. It estimates the corresponding prediction function by determining the specific number of interactions that maximizes the time series data likelihood [5].

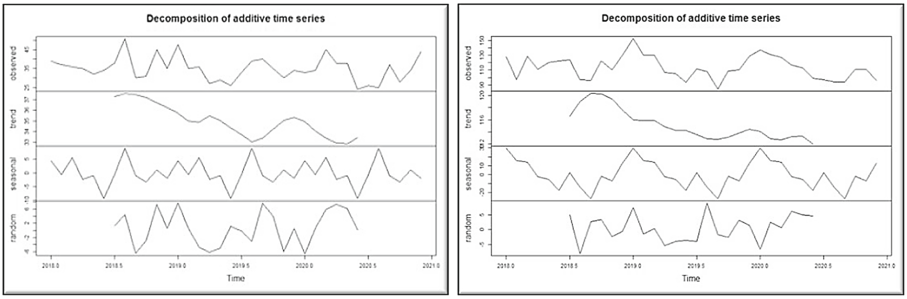
Support Vector Machines with Linear Kernel model- SVMLK
This model was originally created for pattern recognition purposes but, then, it was extended for predictive estimations. Nowadays, it is a powerful model used to estimate a function based on the original data set and to forecast accurately future values [19].

Generalized Linear Model with Elastic Net - GLM
Generalized linear models are an extension of the classical linear regression models. This model is based on establishing a function which links the original data with the predictive ones. In other words, it builds the relationship between the historical data set values and the predictive ones in a linear manner [11].

5 Experiments and Results

5.1 Pre-covid 19 Model: Training Data Set 2016-2018 and Test Data Set 2019

Firstly, the original database has been treated to perform the trend analysis and to examine its cyclical and seasonal variations. According the Fig. 3, the lower category of caskets shows a long term decrease trend which indicates a trend decline in the general direction of the time series data base. Regarding its seasonal variations, it can be seen that there are nearly similar patterns that the specific time series seems to describe along the years. As for the random movements, in this case, it shows the occasional and casual changes due to unpredicted events such as Covid-19. For the medium category of sold caskets the long term trend of the original data is descending. With respect to its seasonal variation, there are long term oscillations showing identical patterns.



(a) Lower caskets

(b) Medium caskets

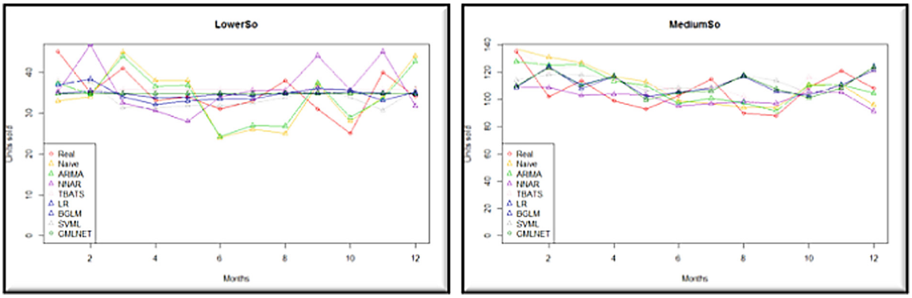
Fig. 3. Lower and Medium caskets series decomposition

The time period chosen for the analysis needs to be considered as it contributes to improve accuracy. Generally, the larger the prediction the more inaccurate it is likely to be [18]. The errors of the method have been compiled in Table 1. Analyzing the numeric outcomes obtained with each of the models we can state that the most accurate models for the lower category in this specific data set are either the Boosted Generalized Linear Model or the Generalized Linear model with elastic net as represented in Fig. 4.

According to [12] MAPE values gives a good prediction for the low and medium categories. Values for the medium category of the ARIMA, TBATS and NNAR methods give a prediction with a high precision. On the other hand, analyzing the RAE, we look for the values closest to zero. For the lower category, we highlight the values of BGLM, SVMLK and GLM that are less than one and would be good predictors. In the case of the middle category, the three models with values below one are ARIMA, TBATS and the best of them is NNAR which, seeing the graphical representation, is the one that best fits the data.

Table 1. Pre-Covid19

Method	Lower category		Medium category	
	MAPE	RAE	MAPE	RAE
Naive	0.183	1.625	0.110	1.086
ARIMA	0.160	1.394	0.095	0.945
NNAR	0.157	1.298	**0.079**	**0.834**
TBATS	0.127	1.064	0.098	0.960
Linear regression	0.122	1.044	0.136	1.336
BGLM	**0.112**	**0.956**	0.133	1.303
SVMLK	0.114	0.996	0.136	1.315
GLM	0.116	0.992	0.131	1.278



(a) Lower caskets prediction

(b) Medium caskets prediction

Fig. 4. Lower and Medium caskets Pre-Covid 19 prediction

5.2 Covid 19 Model: Training Data Set 2016–2018 and Test Data Set 2020

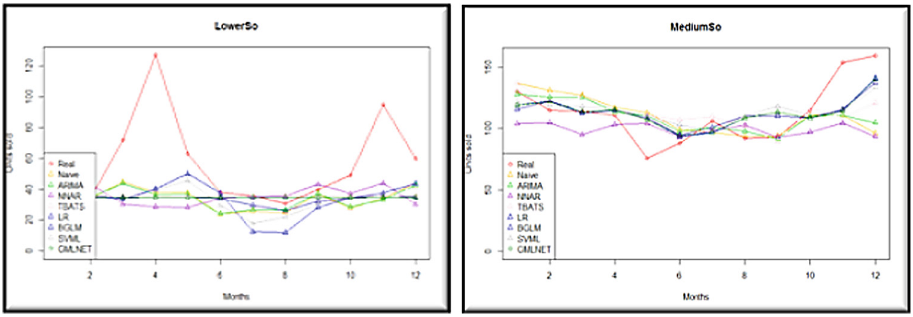
In this scenario the prime objective is to find the most accurate model when the product demand is influenced by an externality like Covid-19. The numeric results are expressed in Table 2.

Comparing both scenarios, it highlights that the value of the errors has increased exponentially. This is due to the fluctuation because of the Covid 19 pandemic situation occurred in 2020. From Table 2 it is identified that the best criterion are TBATS and NNAR for the lower category. On the other hand, for the medium category the best models to make predictions are Linear Regresion and NNAR.

Table 2. Covid19

Method	Lower category		Medium category	
	MAPE	RAE	MAPE	RAE
Naive	0.312	1.018	0.150	0.959
ARIMA	0.316	1.033	0.131	0.837
NNAR	0.306	**0.987**	**0.126**	0.843
TBATS	**0.281**	0.993	0.137	0.833
Linear regression	0.359	1.065	0.131	**0.767**
BGLM	0.296	1.031	0.133	0.785
SVMLK	0.358	1.045	0.148	0.851
GLM	0.287	1.014	0.137	0.808

It is highly important to scrutinize the graphical results in this case. Figure 5(a) shows that for the lower category none of the models can predict such a colossal increase of the demand of coffins due to Covid 19. The medium category has the same concern regarding the pandemic situation. There are two peaks of demand that the models cannot predict accurately. On this occasion, it is more complicate to interpret the graphical results and to conclude which one is the most accurate. Across the board, Linear Regression is the most suitable model for this research, as represented in Fig. 5(b).



(a) Lower caskets prediction

(b) Medium caskets prediction

Fig. 5. Lower and Medium caskets Covid 19 prediction

6 Conclusions and Future Work

Based on the findings of the study, the sales volume of coffins shows a slight seasonal tendency where detected patterns change with a certain regularity in a certain time lapse. Moreover, many predictive models based on historical data can be used to detect trends and predict future variables, but in this study the best model to forecast the demand in coffins in a normal period in the medium category is the NNAR model. Conversely, the most accurate model to forecast the future trend of lower category coffins is either the Boosted Generalized Linear. Furthermore, for a time period influenced by an externality like Covid 19 the best model to forecast future value changes. We can say that Autoregresive Neural Networks gives a decent approximation in both categories. Although it is true that none of the models is capable of predicting the largest peaks of the pandemic, especially in the model of low caskets, which were the most demanded.

It is important to say that in a sector that is really sensitive, this study has been limited by the confidentiality of the data, perhaps with more information more precise predictions could be made.

As future work, it is proposed to obtain post-Covid 19 data to check if the models adapt to the new situation. Also implement this prediction system in the company to improve the inventory management.

Acknowledgements. We are very grateful to the company San Jose Funeral Services for having authorized this study. We thank all the authority chiefs for having collaborated with us.

References

1. Asociación Nacional de Servicios Funerarios (PANASEF): Radiografía del sector funerario 2020 (2020). https://www.panasef.com/wp-content/uploads/2020/10/PANASEF-Radiograf~Ana-2020.pdf
2. Beard, V.R., Burger, W.C.: Change and innovation in the funeral industry: a typology of motivations. OMEGA J. Death Dying **75**(1), 47–68 (2017). https://doi.org/10.1177/0030222815612605. PMID: 28395641
3. Beard, V.R., Burger, W.C.: Selling in a dying business: an analysis of trends during a period of major market transition in the funeral industry. OMEGA J. Death Dying **80**(4), 544–567 (2020). https://doi.org/10.1177/0030222817745430. PMID: 29235385
4. De Livera, A.M., Hyndman, R.J., Snyder, R.D.: Forecasting time series with complex seasonal patterns using exponential smoothing. J. Am. Stat. Assoc. **106**(496), 1513–1527 (2011)
5. Hofner, B., Mayr, A., Robinzonov, N., Schmid, M.: Model-based boosting in R: a hands-on tutorial using the R package mboost. Comput. Stat. **29**(1), 3–35 (2014)
6. Hyndman, R.J., Athanasopoulos, G.: Forecasting: Principles and Practice. OTexts (2018)
7. Hyndman, R.J., Khandakar, Y.: Automatic time series forecasting: the forecast package for R. J. Stat. Softw. **27**, 1–22 (2008)
8. Iqbal, N., Bakhsh, K., Maqbool, A., Ahmad, A.S.: Use of the ARIMA model for forecasting wheat area and production in Pakistan. J. Agric. Soc. Stud. **1**(2), 120–122 (2005)
9. Johnson, C.H.: Small business inventory expensing. Tax Notes (2010)
10. Kuhn, M.: Building predictive models in R using the caret package. J. Stat. Softw. **28**, 1–26 (2008)
11. McCullagh, P., Nelder, J.A.: Generalized Linear Models. Routledge, New York (2019)

12. Meade, N.: Industrial and business forecasting methods - Lewis, CD. J. Forecast. **2**(2), 194–196 (1983). https://doi.org/10.1002/for.3980020210
13. Miranda, P.A., Garrido, R.A.: Incorporating inventory control decisions into a strategic distribution network design model with stochastic demand. Transp. Res. Part E Logist. Transp. Rev. **40**(3), 183–207 (2004)
14. Porras, S., Jove, E., Baruque, B., Calvo-Rolle, J.L.: A comparative analysis of intelligent techniques to predict energy generated by a small wind turbine from atmospheric variables. Logic J. IGPL (2022)
15. Series, T.: Forecasting. J. Forecast. **8**, 141–155 (2017)
16. Sultana, N., Sharma, N.: Statistical models for predicting swine F1u incidences in India. In: 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), pp. 134–138. IEEE (2018)
17. Taskaya-Temizel, T., Casey, M.C.: A comparative study of autoregressive neural network hybrids. Neural Netw. **18**(5–6), 781–789 (2005)
18. Turner, L.W., Witt, S.F.: Forecasting tourism using univariate and multivariate structural time series models. Tour. Econ. **7**(2), 135–147 (2001)
19. Zhong, Y., Zhao, L., Liu, Z., Xu, Y., Li, R.: Using a support vector machine method to predict the development indices of very high water cut oilfields. Pet. Sci. **7**(3), 379–384 (2010). https://doi.org/10.1007/s12182-010-0081-1

Explainable Artificial Intelligence for the Electric Vehicle Load Demand Forecasting Problem

Juan Alberto Gallardo-Gómez, Federico Divina, Alicia Troncoso,
and Francisco Martínez-Álvarez(✉)

Data Science and Big Data Lab, Pablo de Olavide University, 41013 Seville, Spain
{jagalgom,fdivina,atrolor,fmaralv}@upo.es

Abstract. In this work we will address the short-term electricity consumption forecasting problem related to the electric vehicle load demand. In particular we will focus on the explainability of the model obtained. These are important aspects of this problem, since it would help gaining insight on the most important features involved in the forecasts. For the purpose of forecasting, we will use linear regression and three machine learning methods: random forest, gradient boosting and long short-term memory artificial neural network. Later, We add an explainability layer to the models generated, to get a better understanding of the predictions. As far the predictions are concerned, results obtained by the long short-term memory neural network are more accurate than those obtained by random forest and gradient boost, having used linear regression as baseline. The features that most contribute to the predictions are the 25 closest to the present but also a set of features with 30 to 60 unit lag.

1 Introduction

The vast majority of vehicles still rely on fossil fuels nowadays. It follows that these vehicles are raising serious public concerns about energy sustainability and environmental friendliness. As a consequence, electric vehicles (EVs) are becoming more and more popular and widespread as EVs have been recognized as a mean to reduce CO_2 emissions.

Accurate real-time prediction of energy consumption of EVs is a critical factor for improving the usability of EVs [5]. However, even if accurate predictions may help in the spread of the use of EVs, since they will help the drivers to estimate the remaining driving range, it will not be enough for policy-makers. In fact, many predictions systems rely on the so called black-box methods, such as deep learning [15]. This would imply that the managers of ITS will not be able to understand the reasons behind the predictions. This implies that estimating the load demand will be possible, but that the reasons that lead to that demand will not be available, and so no further actions that could be taken for optimizing the infrastructures may not be taken.

Inspired by such observations, the aim of this work is to give insight into the explainability of the short-term electricity consumption forecasting machine

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 413–422, 2023.
https://doi.org/10.1007/978-3-031-18050-7_40

learning models, focusing on machine learning models. In order to obtain a better understanding of the predictions we use the LIME Python package [9], which provides us a feature relevance ranking for each snapshot in time of the predictions.

In order to investigate the effectiveness of our proposal, we have used a dataset containing data of the electricity consumption at the recharge of EVs from many areas of the UK. The data was generated within the My Electric Avenue project [13]. We have then generated some prediction models, based on machine learning (ML) and used the LIME package in order to explain the model.

The rest of the paper is structured as follows. Section 2 presents an overview of the state of the art in the field of deep learning, explainable artificial intelligence (XAI) and its application to time series. Section 3 describes the proposed methodology, while Sect. 4 reports and discusses the results achieved. Finally, Sect. 5 draws the main conclusions and discusses some possible further developments.

2 Related Works

In 2015, Tjoa and Guan [14] summarized the current scenario with regards to explainability and interpretability in ML models, and applied categorization techniques to the medical domain. The main conclusions of this work was that there was the need to improve the reliability of those explanations.

Sahakyan et al. [10] covered in a survey the available techniques for explaining tabular data, being the LIME package the most common among the techniques designed for outcome explanations. There is another very remarkable survey in [3] in which concepts, taxonomies and challenges are addressed.

Tosun et al. [16] proposed a platform to assist pathologist in breast lesions diagnosis called HistoMapr-Breast. This software recognizes the key areas for diagnosis to provide pathologist with a visual guide and it is the first medical XAI application.

Schlegel et al. [11] evaluated XAI techniques on image and text time series, claiming that there is a lack of interpretable visual explanations for time series and very poor XAI methods in general for Recurrent Neural Networks (RNN).

In [6] SHapley Additive exPlanation (SHAP) was used to remove redundant features in a classification problem of earthquake collapsed or non-collapsed buildings. The authors were able to identify that the dataset was biased, suggesting the usage of XAI with that purpose. In [1], SHAP was also used to extract the most relevant features for vegetation cover mapping.

Arras et al. [2] proposed a framework to evaluate computer vision explanations. Basically, the proposed method consisted in a ground truth dataset called CLEVR-XAI with pixel heatmaps and two new metrics for evaluation that aim to improve the trustworthiness of XAI techniques in computer vision.

Muddamsetty et al. [7] presented an XAI method called SIDU. This algorithm over performs the former computer vision explainers, being able to provide more precise areas of the image that explain the prediction.

In view of the works reviewed, XAI applied to time series is still an open problem and this work intends to apply a currently available XAI methods to explain predictions for electricity demand obtained with ML.

3 Methodology

This section describes the methodology proposed. Thus, a general flowchart can be found in Fig. 1, illustrating the steps followed in this study. The next subsections details all the steps involved in such general flowchart.

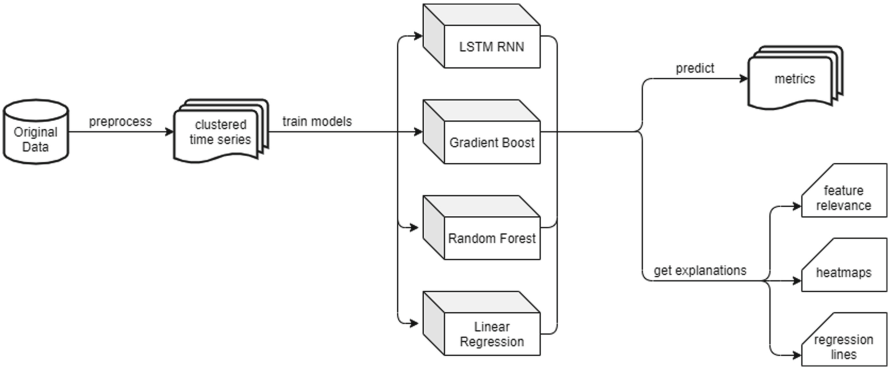


Fig. 1. Flowchart of the proposed methodology.

3.1 Data Acquisition

The first step consists in obtaining the time series we will be using. From one of the source files of My Electric Avenue data, the load demand is extracted and shaped into different time series, one for each group of population. The time series were transformed into a matrix with the structure shown in Fig. 2, where there is an historical window of values to forecast a prediction horizon. In this case the window is set to 168 and the prediction horizon to 24, that is, seven days are used as pattern sequence to predict the following day.

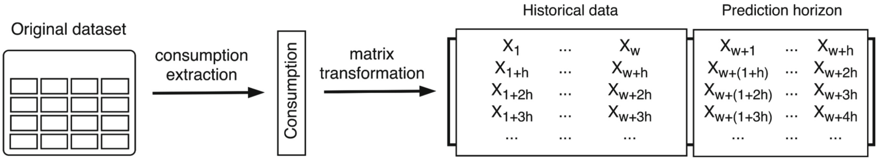


Fig. 2. Data transformation: historical data and prediction horizon.

3.2 Machine Learning Methods

The second step consists in comparing the performance of four ML methods: Linear Regression (LR), Random Forest (RF), Gradient Boost (GB) and Long-Short Term Memory Recurrent Neural Network (LSTM RNN). The LR and both decision tree methods were implemented in Python using the scikit-learn library [8], and the LSTM is built under Keras library [4]. The reason behind using these machine learning methods is that they offer different learning paradigms.

3.3 Models Explainability

In order to obtain a better understanding of the LSTM predictions we will be using the Local Interpretable Model-agnostic Explanations Python package (LIME) [9] which provides us a feature importance ranking for each snapshot in time of the prediction. The LIME explainer generates a fake permuted dataset for each sample and calculates the distance within the original data. It then makes a prediction on the fake data and picks the features that best describe the original model. Finally it fits a simple model on the fake data with the selected features and uses weights derived from it to explain the feature contribution for the original sample prediction.

To help the understanding of how the top features can change throughout the prediction horizon, we will generate a heatmap plot, where the more highlighted areas of the heatmap correspond to those features which have contributed the most to the predictions. In order to plot the heatmap, we define a zeros matrix with h rows and W columns (prediction horizon and historical window). We count over that matrix for each sample of the data how many times the W variables have appeared on the rankings of the H explanations.

3.4 Quality Parameters

Three metrics have been calculated to measure the performance of the models: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and Mean Squared Error (MSE). The formulas can be found below:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - x_i| \tag{1}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n} \frac{|y_i - x_i|}{x_i} \tag{2}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n} \left(y_i - x_i\right)^2 \tag{3}$$

4 Results

4.1 Dataset Description

The original data was obtained on request from My Electric Avenue project [13]. The project took place in the United Kingdom, where electric vehicles were given to various groups of population in different areas in order to measure the electricity consumption at the recharge of the vehicle. The reason behind this was the growing tendency of the EV and sustainability in personal transport in general, which could cause a serious impact to the grid due to the load demand.

The data obtained contain measures for each population group, identified by a prefix. The measures were filtered in clusters [12], creating the time series. The distance between measures is 10 min. Each prefix denoting a cluster reflects the respective urban area, these are Lyndhurst (BL), Chiswick (CC), Chineham (CRG), Shouth Gosforth (GC), Whiteley (JD), Marlow (MC), South Shields (SS and SS2) and Your Homes (YH).

All clusters except for YH follow a similar distribution, but some are closer to each other like MC to SS or BL to CRG, JD and SS2. Due to page limitations, we only present and discuss results regarding clusters SS, CRG, SS2 and YH.

4.2 Results Discussion

In this section, we report the results obtained by the machine learning algorithms applied and discuss them. First, we include the configuration selected for each method, which have been found following a grid search strategy, for the most relevant parameters (the remaining parameter values were left as default):

1. RF: estimators = 500, bootstrap = true.
2. LR: fit intercept = true.
3. GB: estimators = 500, maximum depth = 4, learning rate = 0.01, loss function = squared error.
4. LSTM: units = 64, dropout = 0.1, dense layers = 24, optimizer = Adam, learning rate = 0.001, loss function = squared error.

Table 1 summarizes the errors reported for the CRG cluster. It can be seen that RF, GB and LSTM reached quite similar errors, being that of LR significantly greater. It is worth noting that the lowest MAPE was reached by LSTM but with higher MSE, which can be explained as a wide variety of errors in its prediction. LSTM obtained the smallest errors but also some much bigger than RF and GB.

Table 1. Results for the CRG cluster.

Method	MAE	MAPE	MSE
RF	19.01	15.28	684.88
GB	19.09	15.32	691.42
LSTM	19.92	15.04	807.12
LR	21.69	29.07	844.87

Analogously, Table 2 reports the errors obtained for the SS cluster. In this case, RF, GB, and LSTM obtained quite similar errors but, again, LSTM reached the lowest error in terms of MAPE. MSE for LSTM, on the contrary, was greater than those of RF and GB, indicating that there exists a larger error interval.

Table 2. Results for the SS cluster.

Method	MAE	MAPE	MSE
RF	20.14	22.31	733.06
GB	20.25	22.31	745.84
LSTM	20.14	21.07	754.04
LR	23.21	26.56	931.22

In Table 3 the errors found for the SS2 cluster can be found. As in the previously commented clusters, RF, GB, and LSTM obtained quite similar errors but, again, LSTM reached the lowest error in terms of MAPE. Again, MSE for LSTM was greater than that of RF and GB. This situation can be explained with the same reasoning as aforementioned. LR obtained larger errors and seems not to be competitive when compared with the rest of machine learning algorithms.

Table 3. Results for the SS2 cluster.

Method	MAE	MAPE	MSE
RF	21.06	18.83	872.81
GB	21.49	19.46	886.87
LSTM	21.52	18.47	944.30
LR	25.13	23.62	1104.63

Finally, Table 4 shows the errors obtained for the YH cluster. For this cluster, LSTM obtained the best results in terms of all metrics considered, even for MSE. RF was the second best algorithm and GB the third one. Results reported by LR, once again, were quite worse than those of the other methods.

Table 4. Results for the YH cluster.

Method	MAE	MAPE	MSE
RF	18.68	13.33	734.20
GB	19.31	13.75	776.38
LSTM	18.06	12.98	689.49
LR	24.15	17.71	1043.68

From the analysis of the previous tables, it can be concluded that there exist non-linear relations in the dataset that cannot be captured by LR. Consequently, RF, GB and LSTM achieve better results, being LSTM slightly superior than RF and GB for both MAE and MAPE, but exhibiting higher MSE, which suggests that LSTM generates the lowest but also the highest errors.

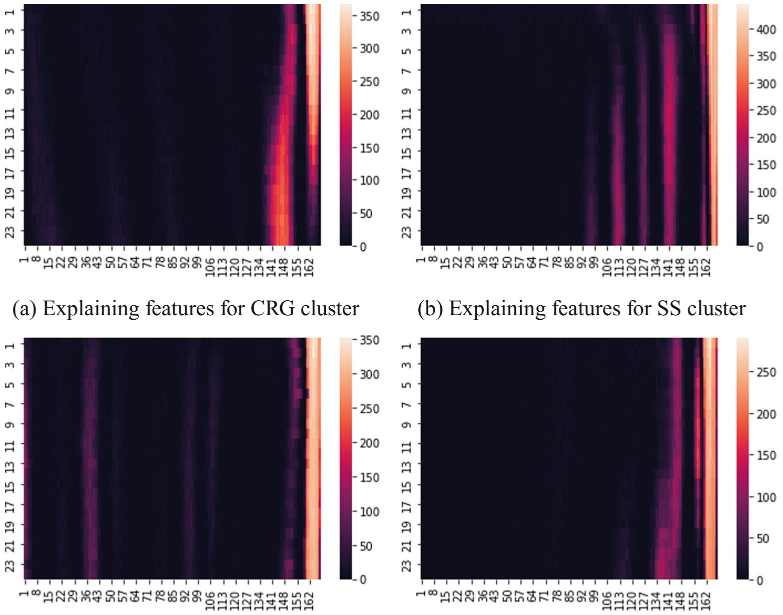
4.3 Explainability

We now explore how the models can be explained in order to better understand the features that influence the most the outputs. Since LR models were used as a baseline and for pages limitation, we will not consider it.

Then, a comparison of the top features obtained from the decision tree models and the heatmaps produced with the LIME package for the LSTM models seems a better approach. Observing the heatmaps in Fig. 3, the x-axis identifies the time whereas the y-axis identifies the features. Thus, the vertical stripes intensify at the furthest values for most of the clusters. This is a clear indicator that the most important values taken into account for the predictions are the closest to the present moment, the last 25 values of the historical window. These patterns are repeated for all the clusters.

If we observe Figs. 4 and 5 for the RF and GB feature relevance plots, respectively, there is a different scenario. The blue bars are the feature importances of the forest, along with their inter-trees variability represented by the error bars. Feature importances are computed as the mean and standard deviation of accumulation of the impurity decrease within each tree. That is, with these figures, we expect to find the most informative features. Most of the charts contain one or several features from the last 25 values of the historical window, like the LSTM heatmaps, but there is another pattern that repeats as well: 30 to 60 values of the historical windows. Some heatmaps have stripes around these values, but they are not clearly differentiated.

These coincidences are so meaningful, there is no doubt that the lasts values of the historical window have a huge contribution to the prediction on both decision trees and the artificial neural network. Thus, this fact should be taken into consideration for further studies related to the short-term electricity consumption forecasting problem.



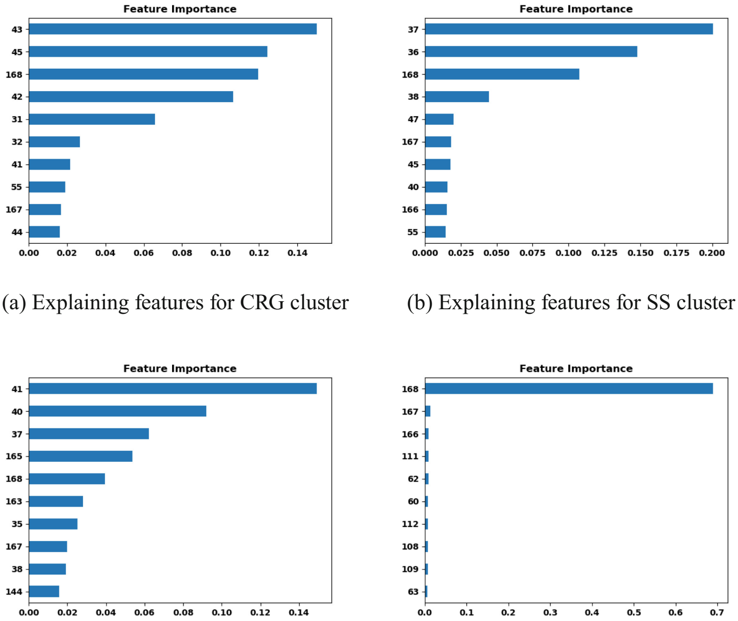
(a) Explaining features for CRG cluster

(b) Explaining features for SS cluster

(c) Explaining features for SS2 cluster

(d) Explaining features for YH cluster

Fig. 3. Explaining feature relevance for LSTM in the four target clusters



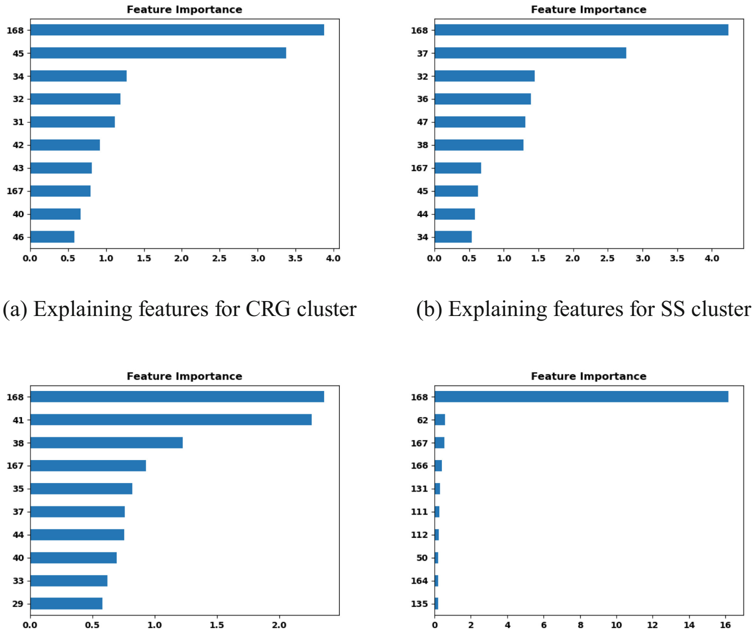
(a) Explaining features for CRG cluster

(b) Explaining features for SS cluster

(c) Explaining features for SS2 cluster

(d) Explaining features for YH cluster

Fig. 4. Feature importance RF charts.



(a) Explaining features for CRG cluster

(b) Explaining features for SS cluster

(c) Explaining features for SS2 cluster

(d) Explaining features for YH cluster

Fig. 5. Feature importance GB charts.

5 Conclusions

In this work we have addressed the short-term electricity consumption forecasting problem related to the electric vehicle load demand, using the data from the project My Electric Avenue. For this purpose, there has been used four machine learning methods: LR, RF, GBT and LSTM. We have later added an explainability and interpretability layer to the models generated, to get a better understanding of the predictions. The results have shown that LSTM performs better in terms of MAE and MAPE than RF and GB, but not in terms of MSE. LR, as expected, obtained the worst results. In terms of the XAI, the features that most contribute to the predictions are the 25 closest to the present. However, the analysis of the features that most contributed in RF and GB were not that close to the present. With these results in mind, we suggest a deeper optimization of the models using metaheuristics.

Acknowledgements. We would like to thank the Spanish Ministry of Economy and Competitiveness for the support under projects TIN2017-88209-C2-1-R and PID2020-11795RB-C21.

References

1. Abdollahi, A., Pradhan, B.: Urban vegetation mapping from aerial imagery using explainable AI (XAI). Sensors **21**(14), 4738 (2021)
2. Arras, L., Osman, A., Samek, W.: CLEVR-XAI: a benchmark dataset for the ground truth evaluation of neural network explanations. Inf. Fus. **81**, 14–40 (2022)
3. Arrieta, A.B., Díaz-Rodríguez, N., del Ser, J., et al.: Explainable Artificial Intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. Inf. Fus. **58**, 82–115 (2020)
4. Chollet, F., et al.: Keras (2015). https://github.com/fchollet/keras
5. Gómez-Quiles, C., Asencio-Cortés, G., Gastalver-Rubio, A., et al.: A novel ensemble method for electric vehicle power consumption forecasting: application to the Spanish system. IEEE Access **7**, 120840–120856 (2019)
6. Martin, S.S., Pradhan, B.: Earthquake-induced building-damage mapping using explainable AI (XAI). Sensors **21**(13), 4489 (2021)
7. Muddamsetty, S.M., Jahromi, M.N.S., Ciontos, A.E., Fenoy, L.M., Moeslund, T.B.: Introducing and assessing the explainable AI (XAI) method: SIDU. CoRR, abs/2101.10710:1–35 (2021)
8. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
9. Ribeiro, M.T., Singh, S., Guestrin, C.: Why should I trust you?: Explaining the predictions of any classifier. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135–1144 (2016)
10. Sahakyan, M., Aung, Z., Rahwan, T.: Explainable artificial intelligence for tabular data: a survey. IEEE Access **9**, 135392–135422 (2021)
11. Schelgel, U., Arnout, H., El-Assady, M., Oelke, D., Keim, D.A.: Towards a rigorous evaluation of XAI methods on time series. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop, pp. 4197–4201 (2019)
12. Scitovski, R., Sabo, K., Martínez-Álvarez, F., Ungar, S.: Cluster Analysis and Applications. Springer (2021)
13. The EA Team. My Electric Avenue. https://eatechnology.com/resources/projects/my-electric-avenue/. Accessed 17 Nov 2021
14. Tjoa, E., Guan, C.: A survey on explainable artificial intelligence (XAI): towards Medical XAI. CoRR, abs/1907.07374(8):1–22 (2015)
15. Torres, J.F., Hadjout, D., Sebaa, A., Martínez-Álvarez, F., Troncoso, A.: Deep learning for time series forecasting: a survey. Big Data **9**(1), 3–21 (2021)
16. Tosun, A.B., Pullara, F., Becich, M.J.M.D., Taylor, D.L., Fine, J.L., Chennubhotla, S.C.: Explainable AI (XAI) for anatomic pathology. Adv. Anat. Pathol. **27**(4), 241–250 (2020)

A Cluster-Based Deep Learning Model for Energy Consumption Forecasting in Ethiopia

Ejigu T. Habtermariam[1], Kula Kekeba[1], Alicia Troncoso[2], and Francisco Martínez-Álvarez[2](✉)

[1] Department of Software Engineering, Addis Ababa Science and Technology University, Addis Ababa, Ethiopia
ejigu.tefera@aastustudent.edu.et

[2] Data Science & Big Data Lab, Pablo de Olavide University, 41013 Seville, Spain
{atrolor,fmaralv}@upo.es

Abstract. The non-linearity and high variability of residential energy consumption data makes household energy prediction more challenging yet vitally important for efficient grid operation and power distribution scheduling. Neither traditional regression techniques nor conventional machine learning models are able to produce accurate forecasts for residential sector. Even though, deep learning methods are demonstrating remarkable performance in many complex problems including time series load forecasting. However, many issues need further investigation by integrating deep learning with clustering algorithms or hybridizing many deep learning models to obtain models with better accuracy. Hence, the aim of the proposed study is to investigate effectiveness of clustering methods to improve deep learning model for energy consumption prediction. In this paper, the K-means algorithm successfully identified interesting cluster groups and energy consumption patterns. Based on well-known error metrics, Long-Short Term Memory networks achieved greater accuracy on clustered data and outperformed the Gated Recurrent Units and Deep Feedforward Neural Network approaches.

Keywords: Clustering · Time series · Forecasting · Electric consumption

1 Introduction

Electricity energy plays a great role in economic development of a nation [1]. Nowadays, electric energy is found to be an important input for sustainable operation of industrial and services sectors. Indeed, all day-to-day activities such as office operation, industries manufacturing, healthcare services and home appliances are energy dependent. However, the difficulty to store electrical energy for further use necessitates a precise energy supply and demand balancing method. Maintaining the stability between electricity generation and demand is crucial to keep power distribution very safe and reliable [2].

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 423–432, 2023.
https://doi.org/10.1007/978-3-031-18050-7_41

In recent years, there has been fast acquisition of time series data from multiple data sources generating huge volume of data on different time resolutions such as hourly, daily and monthly basis, and deep learning has shown to be quite powerful is generating nonlinear models [3].

More specifically, residential electricity consumption is nonlinear [4] highly variable among customers attributed to the electricity prices, geographical conditions, income level of households. Besides to this, in Ethiopia, electric energy consumers specially residents of Addis Ababa city who have access to electricity services have been affected by the power interruption and energy blackouts. Such problems are frequently occurring and affecting the daily operation of business and households. In this regard, power company should have to search better methods of energy demand forecasting and power planning to balance energy demand and supply successfully [2]. One approach to address this problem is the estimation of energy consumption at customer level, which consists in predicting the energy consumption for the next month given a finite history of a customer. To this end, accurate electricity load forecasting has become more challenging yet vitally important for efficient grid operation and power distribution scheduling [5].

In line with this, many studies have been conducted to predict energy consumption using both machine learning and deep learning methods. However, the use of clustering techniques [6] prior to the application of deep learning techniques for residential energy consumption prediction has been rarely studied. For this reason, this is one of the novelties of this work. Furthermore, in Ethiopia there has been no attempt to predict residential power consumption from real world data using the state-of-the-art data driven methodology like deep learning models. Hence, another challenging goal of this study is to investigate the applicability of cluster-based deep learning method for residential energy consumption prediction. The main contribution is to evaluate the impact of the clustering technique in improving deep learning model accuracy for an aggregate residential energy consumption prediction.

The rest of the paper is structured as follows. Section 2 reviews relevant and related articles. The methodology proposed is described in Sect. 3. Results are reported and discussed in Sect. 4. Finally, the conclusions are drawn in Sect. 5

2 Related Works

Many approaches have been proposed recently in the context of deep learning and electric energy consumption forecasting. Thus, Jiang et al. proposed an approach in [7] using deep Long Short-Term Memory (LSTM) networks. It was found that LSTM could extract the feature of power load and better forecasting performance was obtained, compared to support vector regression (SVR) in terms of accuracy.

A pooling LSTM based convolutional neural network (CNN) for short and medium-term electric load forecasting was proposed in [8]. The method resolves the nonlinearity and uncertainty issues of time series load data. More accurate forecast and stable result is obtained by integrating the feature of LSTM and CNN.

Kong et al. [9] proposed a LSTM for individual residential load forecasting. The proposed framework was tested on a publicly available set of real residential smart meter data. As a result, the proposed approach outperformed other algorithms in the task of short-term load forecasting.

To improve the prediction accuracy of building electricity consumption, a hybrid of Genetic algorithm based deep feedforward model is proposed by Luo et al. [10]. The proposed GA-DFNN model showed better ability at tracking the variation of electricity consumption at different hours. Another similar approach can be found in [11], this time, for the Spanish short-term energy consumption.

The authors in [12] proposed an improved day-ahead prediction for load and renewable energy using gated feedback LSTM (GF-LSTM). The best load prediction accuracy is obtained with a 3-layers of GF-LSTM and GF-LSTM with 4 hidden layers for photovoltaic generation, respectively.

Last, the use of clustering to segment clients has also been proposed in the literature. Thus, segmenting residential customers for short-term smart grid load forecasting was investigated by [13]. A K-means clustering algorithm was used to group similar individual consumers and fit distinct models for each cluster. Authors found that clustering similar consumers and aggregating their predictions outperformed a single model in each case and random forest outperform the other proposed techniques with a MAPE of 4.76%. Later, Luna et al. [14] conducted a cluster analysis to discover consumption patterns in a Spanish university. Also clustering was applied to electric energy consumption profiles, but this time, with the aim of detecting fraud [15].

3 Methodology

To construct residential electricity load consumption model, we have used monthly energy consumption data covering 15 months, the first 12 months as training set, the 13^{th} month as validation and the last 2 months as the test set. Hence, we build models to predict one step-ahead monthly residential energy consumption. Each time step contains the individual household energy consumption of a particular month.

With its many tunable hyperparameters, deep learning algorithms have the ability to capture highly abstracted features even for very large and complex data. For this reason, a cluster-based deep learning model is proposed for residential electricity consumption prediction in this paper.

Long-Short Term Memory (LSTM), Gated Recurrent Units (GRU) and Deep Feedforward Neural Network (DFFNN) have been trained and tested on real world residential energy consumption data. However, the analysis has been conducted after the application of clustering techniques to the customers, so particular models have been generated for every cluster, thus improving the performance in terms of accuracy.

In fact, electricity consumption behavior of clients is very much heterogeneous depending on the life style and the economic statues of households [16]. So as to minimize such monthly energy consumption variability, partitioning of electricity consumption profiles into a certain different groups but accommodate more similar data is needed [17]. In this paper K-Means clustering method has been used to group the residential customers into different clusters in which each cluster holds data points closer similarity for which improve model prediction. This is due to the fact that K-Means algorithm has faster computation time and more efficient in grouping of data points into different clusters in the level of members similarity in one cluster is very high while the level of similarity with other cluster is minimized. The Elbow method has been employed to find the optimal number of clusters, K, by computing the sum of squared errors (SSE) for each K value in iterative computations.

It is worth mentioning that for accurate predictions, the input data need to be normalized into a similar range (0,1) when sample data are scattered and if there is large span of sample variable values. Data values within the range (0,1) supports the neural network to converge easily. Hence, the Min-max normalization has been employed to transform data, as given by Eq. (1).

$$X_n = \frac{\left(X_n - X_{min}\right)}{X_{max} - X_{min}} \tag{1}$$

where X_n is a new value for variable X, X_0 is a current value for variable X_{min} is the minimum data point in the dataset and X_{max} is the maximum data point in the dataset.

Different evaluation techniques can be used to determine the prediction performance trained models. However, in this paper, mean absolute error (MAE) is used to compare the absolute values of predicted energy consumption and the actual energy consumption data. In the same way, root mean square error (RMSE) measures the error between the predicted load consumption and the targeted load consumption. Mean absolute percentage error (MAPE) is also used for its easy interpretation.

To sum up, in this paper individual GRU, LSTM and DFFNN algorithms are trained for different clusters of an Ethiopian real electricity load consumption dataset. A flowchart detailing the entire process is illustrated in Fig. 1.

To construct residential electricity load consumption model, we have used monthly energy consumption data covering 15 months, the first 12 months as training set, the 13^{th} month as validation and the last 2 months as the test set. Hence, we build models to predict one step-ahead monthly residential energy consumption. Each time step contains the individual household energy consumption of a particular month.

4 Analysis and Result Discussion

This section reports and discusses the results obtained by the application of the methodology described in the previous section. Thus, Sect. 4.1 describes the data used to assess the methodology performance. Section 4.2 describes the clustering process applied and discusses the clusters obtained. Finally, Sect. 4.3 reports the errors obtained by means of the deep learning models used.

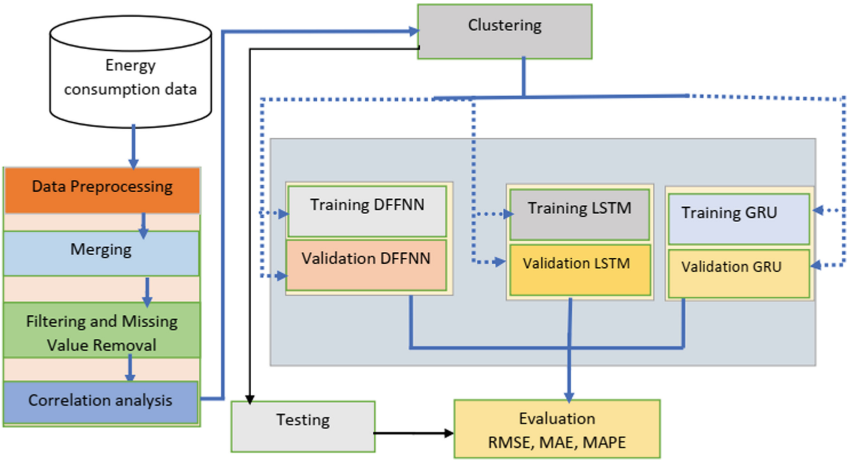


Fig. 1. Proposed methodology to forecast load consumption in Ethiopia.

4.1 Data Description

For this study, monthly residential electricity consumption dataset was collected from the Ethiopian electric utility (EEU). The dataset contains 15 months household energy consumption from November 2019 to January 2020. Firstly, the dataset was obtained in separate excel files for each month from monthly bill reports. To keep the availability and consistency of the sample data in selected months, the individual month consumption files were merged into one file based on the meter number of users using the pandas merge function. Hence, households have been selected in the training and testing sample, if they are connected to the grid and have consumed electricity in the selected months (for 15 months). In this case from each month 8% of households are found to be their energy consumption is zero and therefore removed from samples using filtering technique. Furthermore even we obtained four districts data, due to the issue of the demand of high computation resource and longer LSTM model training time, in this study one district dataset containing 18420 residents for 15 months, leading to a total of 276300 observations.

4.2 Clustering Process

As shown in Fig. 2, where WCSS stands for the sum of squared distance between each point and the centroid in a cluster, the optimal cluster for our data set was determined as three. Therefore, the K-means effectively partitioned energy consumption data points into three different clusters accommodating more similar data.

Figure 3 provides graphical information about the three clusters generated and Fig. 4 shows the size in each cluster. Cluster A comprises the medium energy consumers whose monthly electricity usage can be between 350 KW and 650 KW. On the other hand, Cluster B contains low energy users but large number of customers. Lastly, small number of customers who have higher monthly energy consumption grouped as Cluster C.

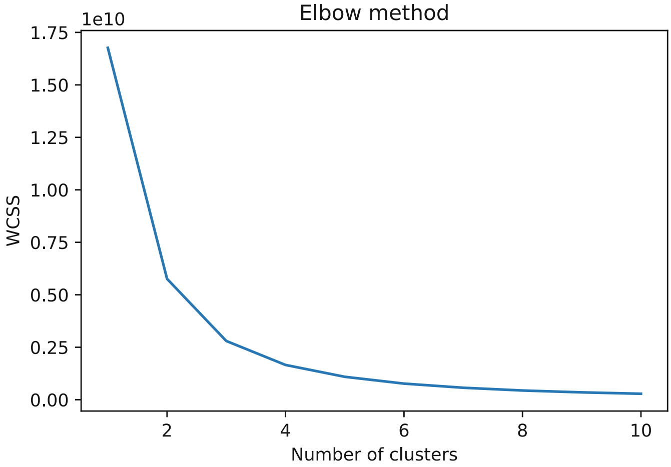


Fig. 2. Elbow method for the number of clusters determination.

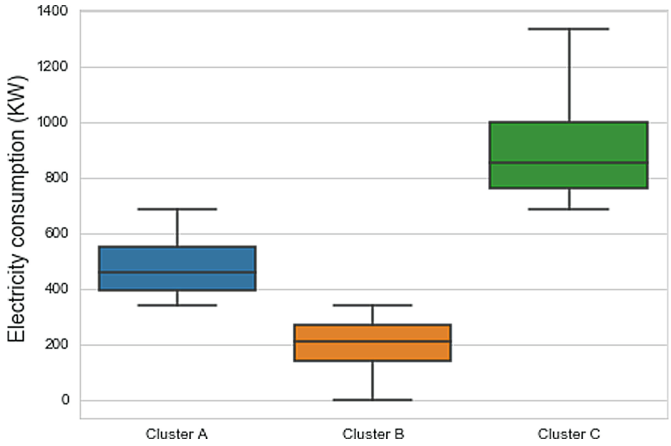


Fig. 3. Energy consumption clusters.

4.3 Results Achieved

The proposed prediction model is trained and evaluated with different parameter configuration of LSTM, GRU and DFFNN. Hyperparameter tuning is one of the best feature of deep learning framework to improve prediction or classification accuracy. Hence, in case of LSTM model training, better prediction results were obtained when we used 3 hidden layers with RELU activation function. At each hidden layer, [16] neurons were used. The epochs and batch size were 50 and 100, respectively. With this configuration, the accuracy of the model was validated with validation set before exposed to the test data. MSE was employed as a metric to signify and compare the training and validation accuracy during prediction model training.

To minimize overfitting problems, we used the L2 regularization. However, the dropout rate can be also used to prevent overfitting issues but this approach may not be effective in case of smaller data due to fact that many important neurons may be suppressed. L2 regularization or weight decay is the sum of the square of the weights multiplied by half the L2 constant λ. This method can tackle over fitting by making the weight values smaller as the training epochs increases.

Table 1 shows model output cluster and unclustered data. From its analysis one can understand that clustering of residential load consumption data before forecasting greatly improves the prediction performance as high variability of the data points captured by the K-means algorithm. Moreover, LSTM and GRU show closer performance for the unclustered case, but LSTM reached quite remarkable values for the clustered data, clearly outperforming the methods it is compared to.

For a better understanding on how LSTM perform over the three clusters, Fig. 5 depicts the individual cluster errors.

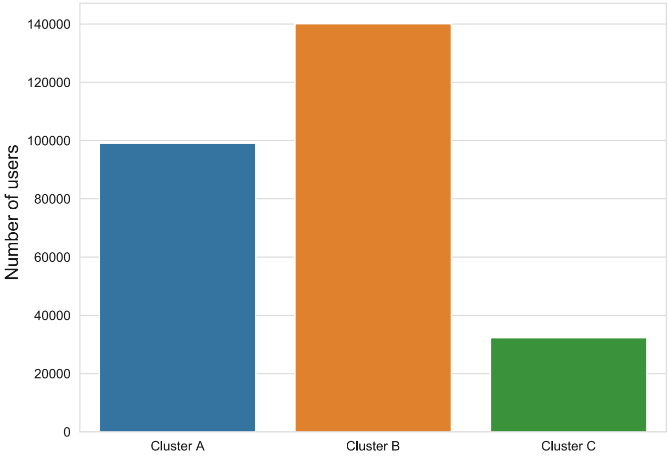


Fig. 4. Size of each cluster.

Table 1. Prediction error for all models, with clustered and unclustered data

Models	Clustered data			Unclustered data		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
DFFNN	81.10	145.32	3.65	347.00	416.00	10.99
LSTM	70.35	83.30	2.35	186.72	245.72	4.10
GRU	72.30	85.90	6.93	189.54	250.96	4.35

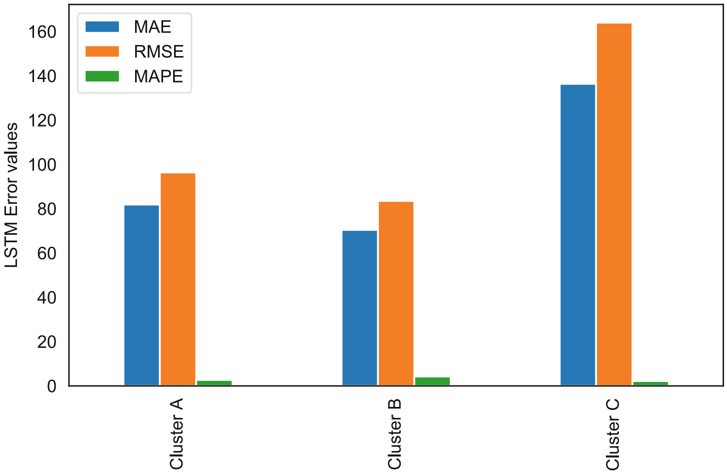


Fig. 5. LSTM prediction error for the different clusters.

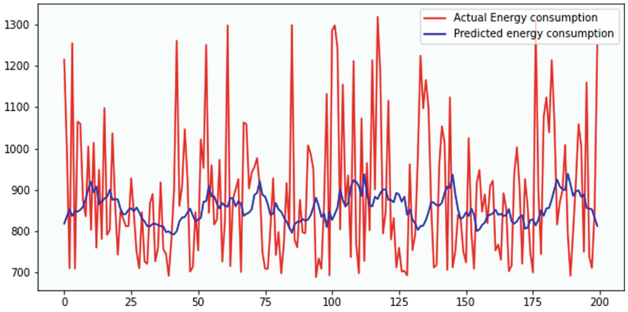


Fig. 6. Actual and predicted energy consumption with LSTM.

As we tested the LSTM predictive capability on each of clustered data with the same network parameter settings, it has been shown that the data size greatly impacts the model performance in addition to the contribution of clustering that minimizes the data variability. In our case it can be observed that Cluster B contains 140109 samples and produces much lower MAE error values as compared with the Cluster C which contains 32287 samples and the LSTM MAE error values higher exactly by half than Cluster B. Based on this result, we confirmed that data clustering effectively detects suitable cluster patterns to improve deep learning prediction accuracy. We also proved that deep learning methods preformed well on relatively larger data than smaller once. Due to the data complexity and the LSTM learning style, plotting predicted and actual data points sometimes may not give readable picture. However, Fig. 6 shows the monthly actual and the predicted energy consumption.

5 Conclusions

A cluster-based deep learning model for residential energy consumption prediction has been developed. Clustering approach is found to be very important to discovering interesting residential consumption patterns from real energy usage data. Cluster based LSTM model achieved better performance as compared with the use of unclustered data in nonlinear energy sector data. The integration of K-means clustering with deep learning algorithms significantly improves the prediction accuracy of the proposed model on complex time series data like energy consumption. As future work, it is expected to develop more complex models and to discover more accurate clusters.

Acknowledgements. The Spanish Ministry of Science and Innovation for the support under the project PID2020-117954RB, the European Regional Development Fund and Junta de Andalucía for projects PY20-00870 and UPO-138516 are acknowledged.

References

1. Hadjout, D., Torres, J.F., Troncoso, A., Sebaa, A., Martínez-Álvarez, F.: Electricity consumption forecasting based on ensemble deep learning with application to the Algerian market. Energy **243**, 123060 (2022)
2. Hwang, J., Suh, D., Otto, M.O.: Forecasting electricity consumption in commercial buildings using a machine learning approach. Energies **13**(22), 5885 (2020)
3. Torres, J.F., Hadjout, D., Sebaa, A., Troncoso, A., Martínez-Álvarez, F.: Deep learning for time series forecasting: a survey. Big Data **9**, 3–21 (2021)
4. Wan, R., Mei, S., Wang, J., Liu, M., Yang, F.: Multivariate temporal convolutional network: a deep neural networks approach for multivariate time series forecasting. Electronics **8**(8), 876 (2019)
5. Rafi, S.H., Nahid-Al-Masood, N.A.: Highly efficient short term load forecasting scheme using long short term memory network. In: Proceedings of the IEEE International Electrical Engineering Congress, pp. 1–4 (2020)
6. Scitovski, R., Sabo, K., Martínez-Álvarez, F., Ungar, Š: Cluster Analysis and Applications. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-74552-3
7. Jiang, Q., Zhu, J.-X., Li, M., Qing, H.-Y.: Electricity power load forecast via Long Short-Term Memory recurrent neural networks. In: Proceedings of the IEEE Annual International Conference on Network and Information Systems for Computers, pp. 265–268 (2018)
8. Yang, Y., Haq, E.U.I., Jia, Y.: A novel deep learning approach for short and medium-term electrical load forecasting based on pooling LSTM-CNN model. In: Proceedings of the IEEE/IAS Industrial and Commercial Power System Asia, pp. 26–34 (2020)

9. Kong, W., Dong, Z.Y., Jia, Y., Hill, D.J., Xu, Y., Zhang, Y.: Short-term residential load forecasting based on LSTM recurrent neural network. IEEE Trans. Smart Grid **10**(1), 841–851 (2017)
10. Luo, X.J., Oyedele, L.O., Ajayi, A.O., et al.: Genetic algorithm-determined deep feedforward neural network architecture for predicting electricity consumption in real buildings. Energy AI **2**, 100015 (2020)
11. Divina, F., Torres, J.F., García-Torres, M., Martínez-Álvarez, F., Troncoso, A.: Hybridizing deep learning and neuroevolution: application to the Spanish short-term electric energy consumption forecasting. Appl. Sci. **10**, 5487 (2020)
12. Toubeau, J.F., Bottieau, J., Vallée, F., de Grève, Z.: Improved day-ahead predictions of load and renewable generation by optimally exploiting multi-scale dependencies. In: Proceedings of the IEEE Innovative Smart Grid Technologies, pp. 1–5 (2017)
13. Kell, A., McGough, A.S., Forshaw, M.: Segmenting residential customers for short term smart grid load forecasting. In: Proceedings of the International Conference on Future Energy Systems, pp. 91–96 (2018)
14. Pérez-Chacón, R., Luna-Romera, J.M., Troncoso, A., Martínez-Álvarez, F., Riquelme, J.C.: Big data analytics for discovering electricity consumption patterns in smart cities. Energies **11**, 683 (2018)
15. Gutiérrez-Avilés, D., et al.: A real big data application for electrical fraud detection. In: Proceedings of the International Conference on Hybrid Artificial Intelligence Systems, pp. 120–130 (2018)
16. Imani, M., Ghassemian, H.: Residential load forecasting using wavelet and collaborative representation transforms. Appl. Energy **253**, 113505 (2019)
17. Fahiman, F., Erfani, S.M., Rajasegarar, S., Palaniswami, M., Leckie, C.: Improving load forecasting based on deep learning and k-shape clustering. In: Proceedings of the International Joint Conference on Neural Networks, pp. 4134–4141 (2017)

Special Session on Optimization, Modeling and Control by Soft Computing Techniques

Abnormal Driving Behavior Identification Based on Naturalistic Driving Data Using LSTM Recurrent Neural Networks

Felipe Barreno[1](✉), Matilde Santos[2], and Manuel Romana[3]

[1] Computer Science Faculty, Complutense University of Madrid, 28040 Madrid, Spain
febarren@ucm.es

[2] Institute of Knowledge Technology, Complutense University of Madrid, 28040 Madrid, Spain
msantos@ucm.es

[3] Civil Engineering School, Technical University of Madrid, 28040 Madrid, Spain
manuel.romana@upm.es

Abstract. This paper presents a classification model to identify abnormal driving behavior on roads. Vehicle dynamics is considered. A LSTM recurrent neural network model-based is applied. The vehicle dynamics features are measured by smartphone inertial sensors. The real data obtained from the GPS, accelerometer, and gyroscope are used to classify the driving maneuvers. A conventional two-lane and a highway roads located in the Madrid Region, Spain, are used for this research. The results obtained with the proposed model are promising and suggest that this intelligent system can be used to warn drivers of a defective or distracted maneuver in real time, aiming at a safer and more comfortable driving.

Keywords: LSTM · Neural networks · Smartphone sensors · Driving · Identification · Classification · Two-lane road · Highway road

1 Introduction

In a world where safety is of utmost importance, the behavior of drivers on the road is receiving special attention from institutions, users and companies. In the automotive sector, the identification of different driving behaviors, mainly those that may entail some risks such as inattention or drowsiness, is essential to keep the driver or other road users safe, especially in highly automatized vehicles [1]. Likewise, in the motor insurance sector, the categorization and observation of driving behavior is fundamental to both the assessment of risk and to the charging of individuals [2].

The road user experience in terms of safe driving depends on different elements. While the most significant factor affecting safety and comfort is the way the driver behaves, the geometry of the road also influences driving safety [3, 4]. In addition, road infrastructure degrades through the use, ageing, etc., and can make an otherwise safe driving maneuver risky. Even more, driving conditions may vary, either because the user's perception may be different depending on the vehicle being driven, or because the geometry of the road may make driving more difficult due to the winding nature of the

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 435–443, 2023.
https://doi.org/10.1007/978-3-031-18050-7_42

road, for example. Hence, the subjective perceptions of the driver must be considered, as the driver may make erroneous or risky maneuvers [6].

Vehicle dynamics help to identify different types of maneuvers based on certain measurements. These characteristics include inertial data, such as accelerometer and gyroscope readings, and GPS measurements, which have been shown to be helpful in this area. For instance, in [5], vehicle's inertial sensors from the CAN bus are used to build a profile of the driver characterizing braking and turning events.

The aim of this work is to identify if a driving event is an anomalous maneuver using inertial sensor measures. To do so, a Soft Computing technique, particularly recurrent neural networks, are applied to identify abnormal maneuvers. The intelligent model here proposed for the driving style characterization includes the road geometry by means of related features such as the vehicle's longitudinal and lateral accelerations as perceived by the driver [13, 21]. That is, the main contribution of this work is that it deals with driving anomalies based on the driver's perception of the vehicle's accelerations derived from the road geometry, as an otherwise appropriate speed could be excessive and thus unsafe for a given road geometry.

The results obtained here on a conventional two-lane road and on a motorway in the Community of Madrid, Spain, are promising and helps define a driving style more in line with safer driving. Besides, this work also supports the interest in using inertial sensors and GPS data from smartphone devices. It is a low-cost system capable of providing information about driving style to detect reckless maneuvers and thus help avoid traffic accidents.

The identification of driving styles using artificial intelligence techniques has been addressed before, which supports the suitability of these strategies. In fact, a popular way of analyzing driving behavior is to focus on maneuvers, as these events provide useful information about the driver performance. Lately, deep learning methods have been proven an efficient time series modelling solution due to their ability to automatically learning the temporal dependencies present in time series [7]. Long short-term memory (LSTM) networks are widely used in pattern recognition applications as a supervised algorithm. This type of network configuration is good for processing time series data [8]. To mention some examples in this particular field, in [9], an abnormal driving behavior recognition method based on a full convolutional LSTM network is proposed. In [10], anomalous driving style is identified using LSTM and replicating NN (RNN). In [11] normal, aggressive, distracted, drowsy, and drunk driving styles are characterized using convolutional NN. In [12] abnormal driving behavior was detected using stacked short-term memory NN. The novelty of this work regarding those previous one yields in the consideration of the road geometry in the driving style.

This rest of the paper is structured as follows. Section 2 presents vehicle dynamics characteristics, which constitutes the basis for the classification model. Section 3 describes the proposed intelligent model that determines when a maneuver is abnormal using real data. Results are discussed in Sect. 4. Conclusions and future works end the paper.

2 Vehicle Dynamics-Based Approach

Whilst driving, the vehicle is exposed to forces that impact its dynamics [14]. If the vehicle is steering, the higher the vehicle speed, the greater the centrifugal force on the vehicle. The parameters that influence vehicle dynamics can change over time (tire pressure, vehicle weight, brake energy dissipation capacity, etc.).

On the other hand, roads are characterized by the geometry, which defines whether a vehicle can be driven at a certain speed with an appropriate comfort and safety level. Vehicle not exceeding the appropriate speed will lead to safe driving and on the contrary, if the vehicle exceeds the maximum speed at a curve, the vehicle accelerations may not be comfortable or safe.

The acceleration measured in the forward direction of the vehicle is the longitudinal acceleration, which is parallel to the linear velocity. The acceleration measured in the orthogonal direction of the vehicle is the lateral acceleration. The acceleration due to the road is calculated, according to the AASHTO Green Book, as [15]:

$$\frac{V^2}{gR} = \rho + f_t \tag{1}$$

$$a_{road} = g(\rho + f_t) \tag{2}$$

where a_{road} (m/s^2), is the critical acceleration due to the road geometry effects, g (m/s^2) is the acceleration of gravity, ρ (m) is the road cross slope and f_t is the maximum transversal friction mobilized coefficient. This road acceleration a_{road} is the limiting acceleration for a horizontal curve to maintain safety and ride comfort.

Regardless of the cross slope and the maximum mobilized coefficient of friction, which can be difficult to obtain, a_{road} can be measured from inertial sensor and GPS. The acceleration due to road effect can be defined as [16]:

$$a_{road} = |\omega| \cdot v_l \tag{3}$$

where ω (rad/s) is the angular velocity and v_l (m/s) is the linear velocity of the vehicle.

According to [16], the lateral acceleration perceived by the driver due to erratic driving is:

$$a_p = |a_m| - a_{road} \tag{4}$$

where a_p (m/s^2) is the driver's perceived acceleration and a_m (m/s^2) is the measured lateral acceleration. From the driver's point of view, this acceleration reflects the driving "feel" due to the effect of the road geometry.

In Fig. 1, some of these dynamic characteristics of a vehicle on the A2 motorway road (Madrid, Spain) are represented [17]. That driving profile corresponds to a normal driving. In the upper image, the accelerometer measure (g, gravity) is shown; at the bottom, the gyroscope readings. All of them for each coordinate, x, y and z.

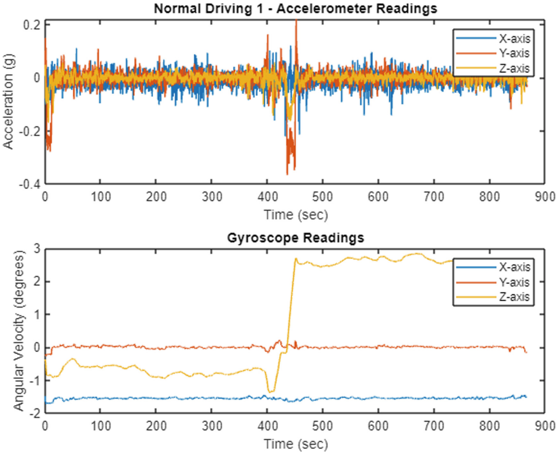


Fig. 1. Inertial measures of a driving on the A-2 highway road.

3 Abnormal Behavior Identification by an LSTM NN Model

The UAH-DriveSet is a public naturalistic driving dataset used is this work [17]. This dataset provides more than 500 min of driving session obtained from 6 different drivers with different vehicles (Table 1). Three different behaviours were simulated (normal, drowsy and aggressive), on two different road classes, a motorway (A-2, Spain) and a secondary road (M-100, Madrid, Spain). The information includes real-time raw data from measurements gathered by inertial sensors (accelerations and gyroscopes) and a GPS located in a smartphone.

Table 1. Driver and vehicle profiles included in UAH-DriveSet [17].

Driver	Genre	Age range	Vehicle	Fuel type
D1	Male	40–50	Audi Q5 (2014)	Diesel
D2	Male	20–30	Mercedes B180 (2013)	Diesel
D3	Male	20–30	Citroën C4 (2015)	Diesel
D4	Female	30–40	Kia Picanto (2004)	Gasoline
D5	Male	30–40	Opel Astra (2007)	Gasoline
D6	Male	40–50	Citröen C-Zero (2011)	Electric

The raw smartphone data contains a considerable amount of noise from vehicle vibrations, so the data are cleaned by a Kalman filter [23]. The cleaned data are found to be well matched to the vehicle's motions. A more comprehensive description of the variables and the data in this dataset can be found at [11].

3.1 Selected Variables for the Classification Model

The variables we are working with are obtained at different sampling rates. The sample time of the inertial sensor is 10 Hz, and the GPS rate is 1 Hz. To synchronize those measures, GPS data have been assumed constant until the next sample, and the same for the accelerometers, that have been subsampled at 1 Hz.

According to the definition of abnormal driving behavior and previous studies, the longitudinal acceleration, lateral acceleration, yaw rate, and speed of the vehicle were selected as the main features.

Linear velocity, v_l (km/h) is given by the GPS. Longitudinal acceleration is measured by the accelerometer (z-axis) in the vehicle onward direction. A positive value means acceleration and a negative one deceleration (braking). Abnormal acceleration and deceleration that increase or decrease the speed are usually manifested as large longitudinal acceleration and deceleration values. Lateral acceleration is obtained from the accelerometer (y-axis) and the angular velocity (ω) is obtained from the gyro sensor. They evaluate the vehicle behavior in a curve since a large increment or decrement may be indicative of excessive turning speed. They are related to abnormal lane changes that are usually manifested as large lateral accelerations and the large yaw rate.

Therefore, the input variables of the neural network model that will be used to detect the abnormal behavior are listed in Table 2.

Table 2. Input variables of LSTM classifier.

Features	Descriptions	Units
a_z	Longitudinal acceleration (from IMU, z-axis)	g
a_y	Lateral acceleration (from IMU, y-axis)	g
w_z	Yaw rate (from gyroscope, z-axis)	rad/s
v_l	Linear speed (from GPS)	km/h

* g = gravity

The standard deviation of speed and longitudinal acceleration can reflect the stability of drivers' vehicle control in the longitudinal direction, while standard deviation of lateral acceleration can reflect the stability of drivers' vehicle control in the lateral direction [18]. Thus, abnormal driving behavior was labeled taking into account events on longitudinal acceleration and lateral acceleration perceived by the driver as follows (5):

$$\overline{X} = \begin{cases} a_z < \mu_{a_z} \;\; x = normal \\ a_p < \mu_{a_p} \;\; x = normal \\ otherwise \; x = abnormal \end{cases} \tag{5}$$

where x is the selected class to a data sample, μ_{ax} is the longitudinal acceleration mean value, μ_{ap} is the lateral perceived acceleration mean value, and $\overline{X}$ is feature vector composed by variables of Table 2.

The characterization of the driver's style is based on the following. If the current speed when going through a curve is excessive, the lateral acceleration will be high, resulting in defective vehicle maneuvering. Thus, the perceived lateral acceleration, which estimates the driver's sensation during the journey, will be noticeable, so the effect of the road geometry is implicitly taken into account through vehicle dynamics by means of the features longitudinal and lateral acceleration, yaw rate and vehicle speed. This could indicate that the vehicle is travelling at a higher speed than the established speed limit.

Thus, from the selected vector of characteristics of the vehicle dynamics (Table 2) it is possible to obtain the driver's perceived acceleration (Eq. (4)), and this way the impact of the road geometry is included.

3.2 Deep LSTM Recurrent Neural Network Model for Classification

A LSTM recurrent neural network [19, 22] is proposed for the classification of abnormal driving maneuvers. The inputs are real measures obtained by the sensors incorporated in a smartphone on the vehicle while travelling.

The dataset for model has a total of 275220 driving maneuvers, of which there are 90464 labeled as abnormal driving.

Because the features have different scales and to avoid extreme value changes in the network weight, the z-score is used to standardize the features (6):

$$z = \frac{x - \mu}{\sigma} \tag{6}$$

where x is the unstandardized data, μ is the mean of the feature vector, σ is the standard deviation of the feature vector, and z is the standardized data.

Two models are proposed, one using D1 profile as pattern profile and another selecting randomly data for training and testing. Several experiments have been performed, of which the best results have been obtained with the following LSTM neural network parameter settings. Computation time became excessively high by increasing the number of hidden layers or by making the number of hidden neurons higher, without increasing the classification performance. The characteristics of the computer used are IntelCore i7 1,2 GHz processor with 8 Gb RAM. For each experiment, the simulations lasted approximately two hours each.

The first model consists in a deep learning neural network, which has an input layer with 4 neurons, one for each feature, and three hidden layers with 125, 100 and 100 neurons, respectively. In these layers there is a dropout layer to avoid overfitting. The initial learning rate used in this study was 0.01. The batch size was 500. The total training epoch number is 100. The second model used has two hidden layers with 125 and 100 neurons, respectively, and the same number of inputs.

4 Results and Discussion

The results of applying the Deep LSTM NN classifier are shown in Tables 3 and 4. To qualify the model accuracy, F_1-score is used [20], given by (7):

$$F = \frac{2pr}{p + r} \tag{7}$$

The expressions that define accuracy, *a*, precision, *p*, and recall, *r*, are the following.

$$a = \frac{TP + TN}{TP + FP + TN + FN} \tag{8}$$

$$p = \frac{TP}{TP + FP} \tag{9}$$

$$r = \frac{TP}{TP + FN} \tag{10}$$

where *TP* is true positive, *TN* is true negative, *FP* is false positive, and *FN* is false negative.

The first classifier uses the D1 driving profile as the model training. A second classifier model was developed using a 5-fold cross validation method. The k-fold strategy was used to ensure that the proportion of samples of each category in the training set and test set were the same as in the original data set.

In Table 3 the confusion matrixes are shown, giving information about the actual and predicted classifications made by each classification model. Results obtained using D1 profile as target pattern and the rest of profiles for testing are shown in Table 4. Comparing them it is possible to see that anomalous driving events are identified correctly because most of the data samples have high percentage of accuracy and large F_1-score. However, the results obtained using the 5 k-fold strategy using 50% data samples for training and validation gives worse results than the first strategy.

To summarize, the LSTM classifier system is able to identify most of the road driving styles, and thus to classify abnormal maneuvers in normal driving. This tool can help achieve a more safety and comfortable driving on roads, using current measurements of smartphones and GPS. It can then lead to a more suitable and appropriate way of driving on these on roads, since if the speed is excessive at passing through a curve, the perceived acceleration is high and this represents a significant risk of an accident.

Table 3. Confusion matrix obtained with LSTM classifier using D1 profile and 5 k-fold strategy.

D2	Predicted class		D3	Predicted class	
True class	30554	2462	**True Class**	31275	2636
	2231	10583		2241	11228
D4	**Predicted class**		**D5**	**Predicted class**	
True class	31503	1654	**True Class**	28029	1610
	681	15702		993	15358
D6	**Predicted class**		**5 k-fold**	**Predicted class**	
True class	27284	1128	**True Class**	83493	9068
	1424	15014		17415	27634

Table 4. Validation result obtained with LSTM classifier using D1 profile pattern and 5 k-fold strategy.

Driver	Accuracy	Precision	Recall	F_1-score
D2	89.76%	87.16%	87.57%	87.36%
D3	89.71%	87.15%	87.79%	87.46%
D4	95.29%	94.18%	95.43%	94.75%
D5	94.34%	93.55%	94.24%	93.88%
D6	94.31%	94.02%	93.68%	93.85%
5 k-fold	81.88%	96.09%	78.06%	78,79%

5 Conclusions and Future Works

In this paper, a deep learning neural network, a recurrent LSTM NN, has been designed and applied to classify abnormal driving maneuvers on conventional two-lane roads and motorway roads. It uses as inputs the real measures of accelerometers, gyroscope, and GPS. Based on those measurements, obtained with low-cost devices (smartphones), it is possible to obtain the acceleration due to road and driver's perceived acceleration. This information allows to classify inefficient, reckless, distraction or inattentive driving behavior while travelling in a vehicle.

The results obtained with this LSTM classifier model are interesting and useful. Some conclusions can be drawn. On the one hand, all driving profiles with abnormal maneuvers have been rightly identified. The database contains only 6 different driving profiles, each one with different characteristics. Besides, different driving styles (normal, drowsy, aggressive) have been considered. Indeed, depending on the driver, normal driving is understood differently.

Several interesting future works can be addressed. First, more data could be considered if available. In addition, following the idea presented in this paper, the relationship of driver behavior to road geometry, mainly road curvature, as well as the perception of speed and acceleration of other vehicles on the road, could be also taken into account.

References

1. Meiring, G.A.M., Myburgh, H.C.: A review of intelligent driving style analysis systems and related artificial intelligence algorithms. Sensors **15**(12), 30653–30682 (2015)
2. Weidner, W., Transchel, F.W., Weidner, R.: Telematic driving profile classification in car insurance pricing. Ann. Actuarial Sci. **11**(2), 213–236 (2017)
3. Santos, M., López, V.: Fuzzy decision system for safety on roads. In: Lu, J., Jain, L.C., Zhang, G. (eds.) Handbook on Decision Making, pp. 171–187 (2012). Springer, Heidelberg. https://doi.org/10.1007/978-3-642-25755-1_9
4. Martín, S., Romana, M.G., Santos, M.: Fuzzy model of vehicle delay to determine the level of service of two-lane roads. Expert Syst. Appl. **54**, 48–60 (2016)
5. Van, L.M., Martin, S., Trivedi, M.M.: Driver classification and driving style recognition using inertial sensors. In: 2013 IEEE Intelligent Vehicles Symposium (IV), pp. 1040–1045. IEEE, June 2013

6. Wu, C., Yu, D., Doherty, A., Zhang, T., Kust, L., Luo, G.: An investigation of perceived vehicle speed from a driver's perspective. PLoS ONE **12**(10), e0185347 (2017)
7. Lara-Benítez, P., Carranza-García, M., Riquelme, J.C.: An experimental review on deep learning architectures for time series forecasting. Int. J. Neural Syst. **31**(03), 2130001 (2021)
8. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1480–1489, June 2016
9. Moukafih, Y., Hafidi, H., Ghogho, M.: Aggressive driving detection using deep learning-based time series classification. In: 2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA), pp. 1–5. IEEE, July 2019
10. Matousek, M., Mohamed, E.Z., Kargl, F., Bösch, C.: Detecting anomalous driving behavior using neural networks. In: 2019 IEEE Intelligent Vehicles Symposium (IV), pp. 2229–2235. IEEE, June 2019
11. Shahverdy, M., Fathy, M., Berangi, R., Sabokrou, M.: Driver behavior detection and classification using deep convolutional neural networks. Expert Syst. Appl. **149**, 113240 (2020)
12. Khodairy, M.A., Abosamra, G.: Driving behavior classification based on oversampled signals of smartphone embedded sensors using an optimized stacked-LSTM neural networks. IEEE Access **9**, 4957–4972 (2021)
13. Barreno, F., Romana, M.G., Santos, M.: Fuzzy expert system for road type identification and risk assessment of conventional two-lane roads. Expert Syst. e12837 (2021)
14. Rajamani, R.: Vehicle Dynamics and Control. Springer Science & Business Media (2011)
15. Transportation Officials. A Policy on Geometric Design of Highways and Streets, AASHTO (2011)
16. Bergasa, L.M., Almería, D., Almazán, J., Yebes, J.J., Arroyo, R.: Drivesafe: An app for alerting inattentive drivers and scoring driving behaviors. In: 2014 IEEE Intelligent Vehicles symposium proceedings, pp. 240–245. IEEE, June 2014
17. Romera, E., Bergasa, L..M., Arroyo, R.: Need data for driver behaviour analysis? presenting the public UAH-DriveSet. In: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), pp. 387–392. IEEE, November 2016
18. Fitch, G.M., et al.: The impact of hand-held and hands-free cell phone use on driving performance and safety-critical event risk (No. DOT HS 811 757) (2013)
19. Chen, S., Zhang, S., Shang, J., Chen, B., Zheng, N.: Brain-inspired cognitive model with attention for self-driving cars. IEEE Trans. Cogn. Dev. Syst. **11**(1), 13–25 (2017)
20. Wang, X., Xu, R., Zhang, S., Zhuang, Y., Wang, Y.: Driver distraction detection based on vehicle dynamics using naturalistic driving data. Transp. Res. Part Emerg. Technol. **136**, 103561 (2022)
21. Barreno, F., Santos, M., Romana, M.G.: A novel adaptive vehicle speed recommender fuzzy system for autonomous vehicles on conventional two-lane roads. Expert Syst. e13046 (2022)
22. Guevara, C., Santos, M.: Intelligent models for movement detection and physical evolution of patients with hip surgery. Logic J. IGPL **29**(6), 874–888 (2021)
23. Maldonado, C.B.G., Penas, M.S., Lopez, M.V.L.: Negative selection and Knuth Morris Pratt algorithm for anomaly detection. IEEE Lat. Am. Trans. **14**(3), 1473–1479 (2016)

Identification of Critical Subgraphs in Drone Airways Graphs by Graph Convolutional Networks

Igone Morais-Quilez and Manuel Graña(✉)

Computational Intelligence Group, University of the Basque Country (UPV-EHU),
Donostia-San Sebastian, Spain
{igone.morais,manuel.grana}@ehu.es

Abstract. Our original aim was to study the control of drones in a congested airspace that may be structured in airways much similar to road or subway traffic networks. The question posed was if it is possible to identify subgraphs whose congestion that may lead to the blockade of the entire network, i.e. if there are dominant subgraphs in the network. We resort to semi-supervised trained Graph Convolutional Networks (GCNs) to formulate and solve the problem. For traffic networks structured as streets or subway lines, the notion of subgraph dominance is very intuitive, because we can postulate that if the labeling of the graph assigns a line label to many nodes belonging to other lines, then the flow in this line should affect these nodes and their corresponding lines. We have carried out experimental work over a manageable network, the Vienna subway network, achieving high and robust accuracy results in the semi-supervised labeling process. However, line dominance results are less robust, they seem to be highly influenced by the experimental setting given by the actual seed nodes provided to networks, which calls for further experiments and analysis in order to try to find stable responses.

1 Introduction

Applications of drones have augmented in the last decade, and their uses have increased in number. Recently, the availability of relatively cheap hardware has aroused the interest for aerial swarms where several flying robots collaborate to achieve a collective task [2,7]. Multi-drone systems are proposed for a broad spectrum of missions, including search and rescue [1], long-term monitoring [10], sensor data collection [8], indoor navigation, environment exploration [5], and cooperative grasping and transportation [6]. However we are interested in the study of the management of airspaces filled with a large number of independent drones carrying independent missions, such as the delivery of goods or persons. In this setting, a natural way to structure the airspace is by the definition of airways that may mimic the road network of a city or its subway network. A natural question then is: "are there subgraphs that may compromise the entire

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 444–453, 2023.
https://doi.org/10.1007/978-3-031-18050-7_43

network or big chunks of it?". Centrality or graph embedding approaches do not seem to provide answers to this question. Therefore we have turned to Graph Convolutional Networks (GNN) [9] to tackle this question. Specifically, we use semi-supervised GCN training in order to identify dominant subgraphs in traffic networks structured like the subway network, where the dominant linear structures provide a quick intuitive interpretation of the subgraph dominance: a line dominates others if the GCN labeling assigns its label to their nodes. In this paper we formulate the approach and provide some computational results over a small and manageable subway network, given our resources (experiments have been carried out in Google Colab).

2 Computational Approach

Our approach to find dominant subgraphs in a traffic network that can be critical for the smooth flow all over the network is based on the application of Graph Convolutional Networks (GCN) [9] for the semi-supervised classification of network nodes into traffic subgraphs. Essentially, traffic subgraphs are characterized by a small subset of nodes that are used as seeds for the classification of the entire graph. Dominant subgraphs are able to colonize the entire graph, thus becoming critical. In this section we review the three GCN approaches for semi-supervised learning that we have tested on a small example below.

2.1 Semi-supervised Learning with Graph Convolutional Network

The problem of classifying nodes in a graph, given seed labels for a small subset of nodes can be framed as graph-based semi-supervised learning, where label information is diffused over the graph via graph-based regularization by using a graph Laplacian regularization term in the loss function [4]:

$$\mathcal{L} = \mathcal{L}_0 + \lambda \mathcal{L}_{reg}, \text{with } \mathcal{L}_{reg} = \sum_{i,j} A_{ij} \; \|f(X_i) - f(X_j)\|^2 = \; f(X)^\top \Delta f(X) \quad (1)$$

Here, $\mathcal{L}_0$ denotes the supervised loss w.r.t. the labeled part of the graph, $f(\cdot)$ can be a neural network-like differentiable function,λ is a weighing factor and X is a matrix of node feature vectors X_i. $\Delta = D - A$ denotes the unnormalized graph Laplacian of an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with N nodes $v_i \in \mathcal{V}$, edges $(v_i, v_j) \in \mathcal{E}$, an adjacency matrix $A \in \mathbb{R}^{N \times N}$ (binary or weighted) and a degree matrix $D_{ii} = \sum_j A_{ij}$. The formulation of Eq. 1 relies on the assumption that connected nodes in the graph are likely to share the same label.

We can encode the graph structure directly using a neural network model $f(X, A)$ and train on a supervised target $\mathcal{L}_0$ for all nodes with labels, thereby avoiding explicit graph-based regularization in the loss function. Conditioning $f(\cdot)$ on the adjacency matrix of the graph will allow the model to distribute gradient information from the supervised loss $\mathcal{L}_0$ and will enable it to learn representations of nodes both with and without labels. Figure 1 shows the architecture of the basic GCN with two hidden graph convolution layers. Its input is the graph adjacency matrix and the labels for a subset of the nodes. Its output is the labeling of all nodes in the graph.

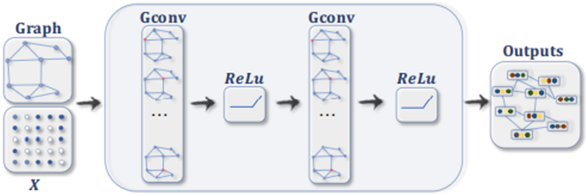


Fig. 1. Architecture of the basic GCN

The formal description of the GCN is as follows:

1. Let $X = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^{n \times p}$ be the collection of n data vector in p dimensions.
2. Let $G(X, A)$ be the graph representation of X with $A \in \mathbb{R}^{n \times n}$ encoding pair wise relationship (similarities between neighbours) among data X.
3. The GCN contains one input layer, several propagation (hidden) layers and one final perceptron layer.
4. Given an input $X^{(0)=X}$ and a graph A, GCN conducts the following layer-wise propagation show in Eq. 2

$$X^{(k+1)} = \sigma(D^{-1/2}AD^{-1/2}X^{(k)}W^{(k)}), \tag{2}$$

where, $k = 0, 1, \ldots, K-1$ and $D = diag(d_1, d_2, \ldots, d_n)$ is a diagonal matrix with $d_i = \sum_{j=1}^{n} A_{ij}$. $W^{(k)} \in \mathbb{R}^{d_k \times d_{k+1}}$, $d_0 = p$ is a layer-specific weight matrix needing to be trained. $\sigma(\cdot)$ denotes an activation function, in this case we used $ReLU(\cdot) = max(0, \cdot)$, and $X^{k+1} \in \mathbb{R}^{n \times d_{k+1}}$ denotes the output of activations in the k-th layer.

2.2 Enhanced GCN

The second approach is an enhanced GCN [3], that includes a final perceptron layer defined as

$$Z = softmax(D^{-1/2}AD^{-1/2}X^{(K)}W^{(K)}) \tag{3}$$

where $W^K \in \mathbb{R}^{d_K \times c}$, and c denotes the number of classes. The final output $Z \in \ltimes^{n \times c}$ denotes the label prediction for all data X in which each row Z_i denotes the label prediction for the i-th node. The optimal weight matrices $W^{(0)}, W^{(1)}, \ldots, W^K$ are trained by minimizing the cross-entropy loss function defined as follows,

$$\mathcal{L}_{Semi-GCN} = -\sum_{i \in L} \sum_{j=1}^{c} Y_{ij} ln Z_{ij} \tag{4}$$

where L indicates the set of labeled nodes.

The architecture of the enhanced GCN shown in Fig. 2 consists of 3 hidden layers topped with a softmax layer and loss, a size of the hidden layer of 16.

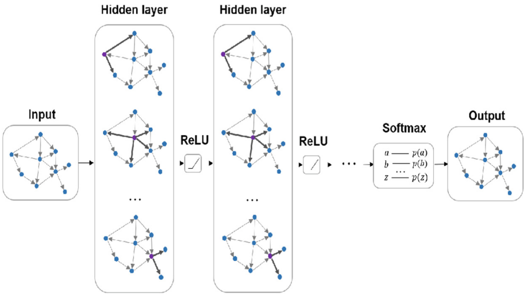


Fig. 2. Architecture of the enhanced GCN

2.3 Advanced GCN

Finally the third approach is a more advanced GCN, which includes 8 repetitions of the same block composed of two GCN layers with intermediate pooling and a final MLP layer, Therefore, the entire network contains 16 GCN layers and 8 MLP layers, as shown in Fig. 3 [9].

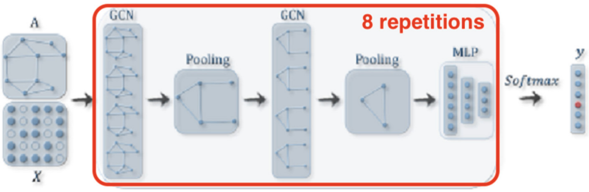
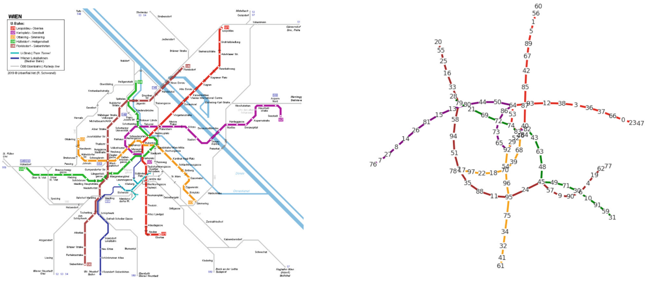


Fig. 3. Architecture of the advanced GCN

3 Some Experimental Results

The graph data that has been used for the computational experiments reported below have been obtained from a Kaggle tutorial. It consists in the graph corresponding to the Vienna subway network. Such graph is very convenient to illustrate the proposed approach because it consists of linear subgraphs with some node intersections, thus the idea that some subgraph is critical for the traffic flow in the entire network can easily be interpreted as the saturation of one subway line leading to the collapse the entire network traffic flow. Figure 4 provides a visualization of the original subway map and its abstract graph representation. We keep the color identification of each line as the label of the nodes in the graph. For intersection nodes, this color label is arbitrary.

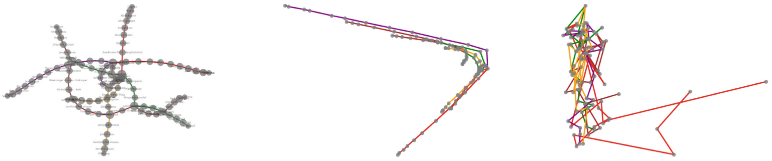


(a) Subway map network

(b) Subway network graph

Fig. 4. (a) Map network of the Vienna subway train (b) The Vienna subway network visualized as an abstract graph.

Figure 5 illustrates the inability of well known unsupervised approaches to give response to our research question. The PageRank analysis does not detect significant differences among nodes and does not give any hint on the possibility that a subgraph may collapse the network. The Node2vec embedding in a 2D space seems to highlight the similarity of nodes in lines that have mores intersections, while the Role2vec embedding seems to collapse all nodes and lines in a small region without apparent structure.



(a) PageRank analysis

(b) Node2vec embedding based on DeepWalk analysis

(c) Role2Vec embedding

Fig. 5. Unsupervised analysis of Vienna subway network by PageRank, Node2vec embedding generated from a DeepWalk analysis, and a Role2Vec embedding

We apply GCN to our problem of detecting dominant subgraphs as follows: we characterize each subway line by a couple of nodes, so we provide the GCN with the color labels of these nodes as the seed labels to generate the labeling of all nodes in the graph. The dominance of a line consists in the labeling of nodes belonging to other lines with its color after convergence of the learning process. Because of the inherent instability of the GCN learning process, which leads to different labelings in each repetition of the training process, we repeat the training of each GCN architecture 200 times. We compute the accuracy of the resulting labeling as its matching against the ground truth labeling of the nodes belonging to the lines. Therefore, we report the average accuracy of each GCN architecture over the 200 repetitions of the training process. Moreover, in order to have some hint about the effect of the selected seed nodes, we have designed two different experiments, carrying out the 200 repetitions for each.

- **First experiment:** Red(60, 47, 38), Brown(20,77,35), Purple(76,80), Green(31,69), Orange(78,61), that correspond to the first station of the line (the ends of the graph), the end station of the line, and in some cases when the line is to long, an intermediate station is chosen (near a point of many intersections).
- **First experiment:** Red(64, 87, 93), Brown(58,78,95), Purple(13,86), Green(21,84), Orange(70,17), that correspond to points with many intersections.

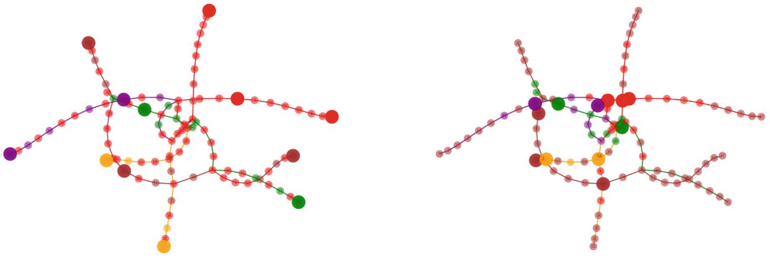
The accuracy results are shown in Table 1 The first row corresponds to the subset of experiments where the labels are the starts and end nodes for each of the lines, and the second row corresponds to the subset of experiments where the labels are another set of nodes, for this case the one with many intersections are selected. Accuracy results seem to be quite independent on the selection of the seed nodes, and they show the superiority of the Advanced GCN. But accuracy results do not provide much information about subgraph dominance. In the following we have computed for each node in the network its dominant label, i.e. the label that has been assigned to it mostly along the 200 training repetitions. We visualize this graph coloring and provide short statistics of the dominant colors for each approach. The reasoning is that if a node has been labeled with a color of a line, then this line flow performance will affect the node, in other words, the subgraph corresponding to this line dominates over the node.

Table 1. Mean accuracy over 200 semi-supervised training repetitions

	Basic GCN	Enhanced GCN	Advanced GCN
1ST experiment	59.936	72.606	**85.086**
2ND experiment	60.161	72.872	**85.023**

3.1 Basic GCN Dominance Results

The visual results of the line dominance computed by the basic GCN are shown in Fig. 6(b), where, for instance, a brown node on a purple line would indicate that this node is dominated by the brown line. If the majority of nodes in the purple line are labeled as brown, then the brown line dominates over the purple line. The big nodes in the figure represent the seed nodes entered to the GCN training. In the Table 4 we can see the percentage of nodes classified for each color on average over the 200 repetitions. For the first experiment setting, the red line seems to dominate over the others, while in the second experiment is the brown line that dominates over the others. This dependency on the seed nodes was not apparent in the accuracy results, but it may greatly influence conclusions about the dominant subgraph lines (Table 2).



(a) First Experiment Basic GCN (b) Second Experiment Basic GCN

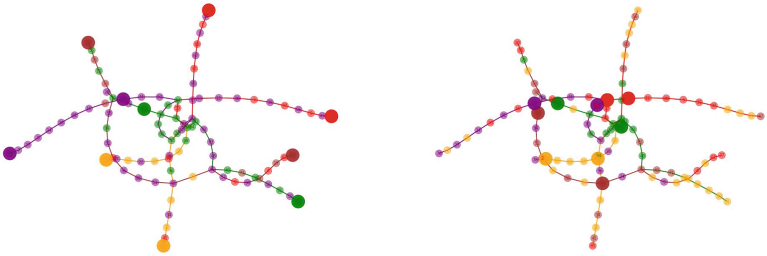
Fig. 6. Visualization of the majority lablels found by the Basic GCN after 200 repetitions for the two considered seed label configurations. Big nodes correspond to seed nodes

Table 2. Percentage of nodes classified by majority color labels assigned by the Basic GCN. Average over 200 repetitions

	Red	Brown	Green	Purple	Orange
1ST experiment	**68.37**%	11.22%	10.20%	6.12%	4.08%
2ND experiment	19.39%	**60.20**%	9.18%	8.16%	3.06%

3.2 Enhanced GCN Dominance Results

The Fig. 7 shows the majority labelings found by the Enhanced GCN in the two experimental settings, where the sensitivity to the seed nodes setting is also apparent. In the Table 3 we can see the percentage of nodes classified for each color. The dominant colors are different from the ones found by the Basic GCN, pointing further to the need to carry out more computational experiments. In this case the purple dominance in the first experiment is not so big as in the previous results with the Basic GCN. Moreover, in the second experiment setting, the dominance of the orange label is very short, and the results point to a balanced labeling that may well reflect the actual structure of the network, that is, there is no actual dominance found.



(a) First Experiment Enhanced GCN (b) Second Experiment Enhanced GCN

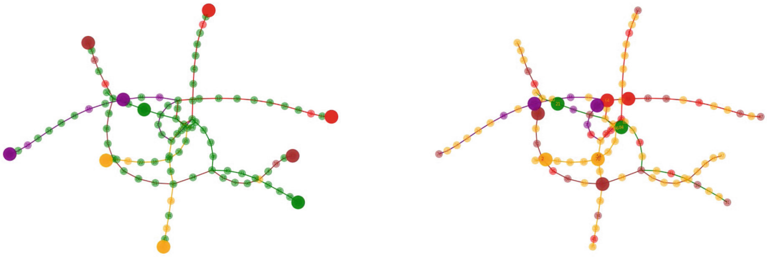
Fig. 7. Visualization of the majority labels found by the Enhanced GCN after 200 repetitions for the two considered seed label configurations. Big nodes correspond to seed nodes

Table 3. Percentage of nodes classified by majority color labels assigned by the Enhanced GCN. Average over 200 repetitions

	Red	Brown	Green	Purple	Orange
1ST experiment	15.31%	8.16%	21.43%	**46.94%**	8.16%
2ND experiment	20.41%	16.33%	16.33%	16.33%	**29.59%**

3.3 Advanced GCN Dominance Results

In this case the Fig. 8(c) shows majority dominance predictions by the Advanced GCN, which are, again, quite different depending on the seed node settings. In the Table 4 we can see the percentage of nodes classified for each color. Dominance of the green line is quite strong in the first experiment setting, while in the second is the orange line that dominates. Such instabilities are not according to the accuracy results.



(a) First Experiment Advanced GCN Predictions (b) Second Experiment Advanced GCN Predictions

Fig. 8. Visualization of the majority labels found by the Advanced GCN after 200 repetitions for the two considered seed label configurations. Big nodes correspond to seed nodes

Table 4. Percentage of nodes classified by majority color labels assigned by the Advanced GCN. Average over 200 repetitions

	Red	Brown	Green	Purple	Orange
1ST experiment	5.10%	4.08%	**79.59**%	5.10%	6.12%
2ND experiment	14.29%	16.33%	2.04%	6.12%	**61.22**%

4 Conclusions and Future Work

Our original aim was to study the control of drones in a congested airspace. The natural consideration of the structure of the cities where such drones will be flying leads us to consider that structures similar to road or subway traffic networks could be the natural shape of the drone airways. The question posed was if it is possible to identify subgraphs whose congestion that may lead to the blockade of the entire network, i.e. if there are dominant subgraphs in the network. Conventional centrality and embedding approaches do not seem to provide an answer to this question, therefore we resort to GCNs and their semi-supervised training as means to formulate and solve the problem. For traffic networks structured as streets or subway lines, the notion of subgraph dominance is very intuitive, because we can postulate that a line may be critical if the labeling of the graph assigns its label to many nodes belonging to other lines, meaning that flow in this line should affect these nodes and their corresponding lines.

We have carried out experimental work over a manageable network, the Vienna subway network, achieving high and robust accuracy results in the semi-supervised labeling process. However, line dominance results are less robust, they seem to be highly influenced by the experimental setting given by the actual seed nodes provided to networks, which calls for further experiments and analysis in order to try to find stable responses.

Future works will address the extraction of airways graphs from simulation and from real data, in order to extend our analysis to more complex graphs and to test the identification of critical nodes against the actual congestion in the simulated or real network when such nodes fail or are compromised. This lead to the notion of traffic security that may be further examined.

References

1. Bernard, M., Kondak, K., Maza, I., Ollero, A.: Autonomous transportation and deployment with aerial robots for search and rescue missions. J. Field Robot. **28**, 914–931 (2011). https://doi.org/10.1002/rob.20401
2. Floreano, D., Wood, R.J.: Science, technology and the future of small autonomous drones. Nature **521**, 460–466 (2015)
3. Jiang, B., Zhang, Z., Lin, D., Tang, J., Luo, B.: Semi-supervised learning with graph learning-convolutional networks. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11305–11312 (2019). https://doi.org/10.1109/CVPR.2019.01157

4. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks (2016). https://doi.org/10.48550/ARXIV.1609.02907, https://arxiv.org/abs/1609.02907
5. Mcguire, K., De Wagter, C., Tuyls, K., Kappen, H., Croon, G.: Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment. Sci. Robot. **4**, eaaw9710 (2019). https://doi.org/10.1126/scirobotics.aaw9710
6. Mellinger, D., Shomin, M., Michael, N., Kumar, V.R.: Cooperative grasping and transport using multiple quadrotors. In: DARS (2010)
7. Schilling, F., Lecoeur, J., Schiano, F., Floreano, D.: Learning vision-based flight in drone swarms by imitation. IEEE Robot. Autom. Lett. **4**, 4523–4530 (2019)
8. Tuysuz Erman, A., Hoesel, L., Havinga, P., Wu, J.: Enabling mobility in heterogeneous wireless sensor networks cooperating with UAVs for mission-critical management. Wireless Commun. **15**, 38–46 (2009). https://doi.org/10.1109/MWC.2008.4749746
9. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on graph neural networks. IEEE Trans. Neural Netw. Learn. Syst. **32**(1), 4–24 (2021). https://doi.org/10.1109/tnnls.2020.2978386
10. Zhang, J., Hu, J., Lian, J., Fan, Z., Ouyang, X., Ye, W.: Seeing the forest from drones: testing the potential of lightweight drones as a tool for long-term forest monitoring. Biol. Cons. **198**, 60–69 (2016). https://doi.org/10.1016/j.biocon.2016.03.027

Robust Velocity Control of an Automated Guided Vehicle Using Artificial Neural Networks

Javier Argente Mena[1], Jesus Enrique Sierra-García[2](✉), and Matilde Santos Peñas[3]

[1] UNED-UCM, Madrid, Spain
jaargent@ucm.es
[2] Department of Electromechanical Engineering, University of Burgos, Burgos, Spain
jesierra@ubu.es
[3] Institute of Knowledge Technology, Complutense University of Madrid, Madrid, Spain
msantos@ucm.es

Abstract. Automated Guided Vehicles (AGV) are currently becoming very popular in terms of internal logistics for factories, as they can transport several tons of weight and can be adapted to different payloads. Nevertheless, when these vehicles transport loads, their dynamics change, and the speed control becomes more complex. Normally the speed control of these systems is based on PI regulators. This requires fine tuning and provides low robustness against changes. In order to improve the velocity control performance, a controller based on neural networks with reference model is developed in this paper, which is compared to PI control. Given the results, it is concluded that the neural control is much more robust than PI control.

Keywords: AGV · Modelling · Neural networks · Proportional integral derivative control · Model reference neural network controller

1 Introduction

Automatic Guided Vehicles (AGVs) can be adapted for any type of task and environment, keeping the option of being used manually [1, 2]. A very common type of AGV in factories is the magnetic tape guided vehicle, the one used in this paper. These robots move along a magnetic tape installed under the floor. The tape emits a magnetic field, captured by a magnetic sensor on board the AGV, sending the position of the magnetic field along its horizontal axis [3]. This sensor measures the lateral displacement from the center with high accuracy. The displacement is sent to the controller either by means of an analog signal, communications or PWM.

But including the system dynamics makes the controller tuning a complex task [4] and greatly influences the trajectory flowing and cruise velocity accuracy, which varies for different working conditions such as payload or mechanical parameter variations due to specific failures or wearing. Hence, the necessity to develop robust controllers that

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 454–463, 2023.
https://doi.org/10.1007/978-3-031-18050-7_44

can adapt to new working conditions or, at least, minimize the influence of disturbances to avoid losing of trajectory.

In [5, 6] reinforcement learning techniques are used for tracking control of an AGV, since the wearing of mechanical parts causes deviations from the trajectory. The results obtained on complex trajectories exceed by far those ones obtained by a PID controller. Also, in [4] fine-tune PID control for trajectory tracking, using genetic algorithms, is presented. It shows how important fine tuning is for controllers to achieve a good performance under parameter variations.

In this paper, robustness for AGV velocity control is compared using two different control techniques: conventional PI and Neural Network control. The robot must reach velocity setpoints and maintain them despite variations of its payload and viscous motor friction. Simulation experiments show how the neural controller adapts better to changes than the PI regulator.

The rest of the paper is organized as follows. Section 2 describes the dynamic features of the AGV and its movement equations. Section 3 explains the control architecture used and the design of the Model Based Neural Network Controller. Section 4 presents the results obtained and a comparation is made. The last section gathers conclusions and future jobs.

2 Description of the System

2.1 AGV Model

In this paper a differential AGV configuration has been used to test the control techniques. Its axis of rotation is in the middle of the traction unit (Fig. 1) and it usually has one or two idle support wheels.

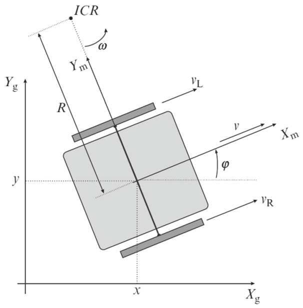


Fig. 1. AGV differential configuration [7].

where w_L, w_R, v_L, v_R are the angular speed (rad/s) and tangential velocity (m/s) of the left and right wheels, respectively, and w is the angular speed of the robot around its instantaneous center of rotation, *ICR*.

In [1] a complex dynamic model for a hybrid configuration (Differential-Tricycle) can be found. It includes the kinematics, wheels, drive unit, body of the AGV, vertical and lateral loads, acceleration and deceleration, among others. Since the aim of this paper is to design an adaptive control, only the first-order differential equations of the robot are considered. These equations were obtained with the Lagrangian of the system, considering non-holonomic constraints for lateral slipping and perfect rolling conditions for the wheels [7]:

$$\frac{di_L}{dt} = \frac{1}{L_m}(u_L - R_m\, i_L - K_b w_L) \tag{1}$$

$$\frac{dw_L}{dt} = \frac{1}{J_L}(K_m\, i_L - B_m w_L) \tag{2}$$

$$\frac{di_R}{dt} = \frac{1}{L_m}(u_R - R_m\, i_R - K_b\, w_R) \tag{3}$$

$$\frac{dw_R}{dt} = \frac{1}{J_R}(K_m\, i_R - B_m\, w_R) \tag{4}$$

$$\frac{dv_D}{dt} = \frac{K_m(i_L + i_R) \ - B_m\,(w_L - \ w_R)}{f \ \cdot m_D \cdot \ r} \tag{5}$$

$$\frac{dw_D}{dt} = -\frac{B\,(K_m\, i_L - K_m\, i_R - B_m\, w_L + B_m\, w_R)}{2J \cdot f \cdot \ r} \tag{6}$$

$$\frac{dx}{dt} = v_D\cos(\beta),\ \frac{dy}{dt} = v_D\sin(\beta),\ \frac{d\beta}{dt} = w_D \tag{7}$$

where i_L, i_R, w_L, w_R, u_L and u_R are the currents (A), angular speeds (rad/s) and voltages (V) of the left and right motors, respectively; K_m, K_b and B_m are the torque (Nm/A), counter-electromotive force $(V/rad/s)$ and viscous friction coefficient (Nms) constants for the DC motors, respectively; v_D and w_D are longitudinal and angular velocities of the robot; β is the angle (rad) between the robot and the non-inertial frame and J, f, r and B are the moment of inertia (Kgm^2), gearbox ratio, wheel radius (m) and distance between the wheels (m), respectively.

3 Description of the Control Strategy

In this work, a neural network controller based on a reference model is used to control the AGV cruise speed. This controller aims to adapt to model parameter variations and non-modeled dynamics. The structure of the controller is shown in Fig. 2.

Its architecture is composed by a neural model of the AGV, a reference model and a neural controller. The neural model that identifies the dynamics of the AGV system must be as accurate as possible. The reference model imposes the desired dynamics to the robot. The neural controller generates the required control actions for the system to follow that reference model. As feed-forward neural network with at least one hidden layer can approximate any arbitrary continuous function, this structure is used for both, the neural model and the neural controller.

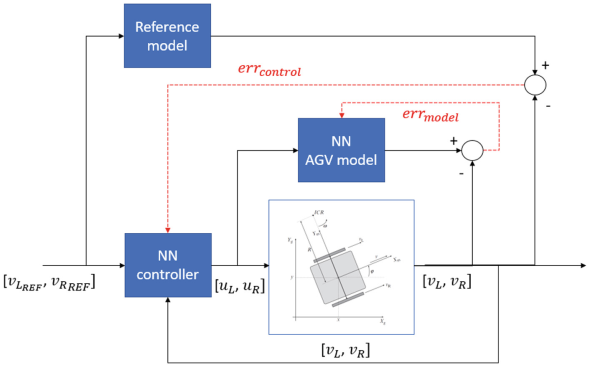


Fig. 2. Neural network controller architecture

3.1 Neural Network Model

The AGV model is a coupled MIMO system. Its neural model has as inputs the voltages from the DC motors u_R and u_L, and its outputs are the wheel tangential speeds v_R and v_L. The neural network is a multilayer perceptron (MLP) with: 18 inputs; one hidden layer with 25 neurons (sigmoid tangent) and an output layer (linear) with 2 neurons. The number of neurons of the hidden layers has been determined empirically as the results obtained were satisfactory. For its training, Levenberg Marquardt algorithm was used. The input vectors for this network are:

$$U_R = [u_R(k), \ldots, u_R(k-3)]\ U_L = [u_L(k), \ldots, u_L(k-3)]$$

$$V_R = [v_R(k-1), \ldots, v_R(k-5)]\ V_L = [v_L(k-1), \ldots, v_L(k-5)]$$

The forward propagation of the inputs can be represented by:

$$v_{NN} = f_2\left(W_2^T f_1\left(W_1^T x + B_1\right) + B_2\right) \tag{8}$$

where v_{NN} are the neural model speed and W_1, W_2, B_1 and B_2 are the weights and biases of layers 1 (hidden) and 2 (output), respectively, and x is the input tuple. As cost function for weigh and bias updating, Eq. (9) is used:

$$E_m = \frac{1}{2}\sum\nolimits_{i=1}^{n}\left(v_{ref_i} - v_i\right)^2 \tag{9}$$

The backwards propagation of the model error can be expressed as,

$$S_i^2 = \frac{\delta E_m}{dh_i{}^2}\dot{f_i}{}^2 \tag{10}$$

$$\frac{\delta E_m}{dw_{ij}^2} = \sum\nolimits_{j=1}^{m_2} S_j^2 h_i^1 \tag{11}$$

$$\frac{\delta E_m}{db_i^2} = S_i^2 \tag{12}$$

where S_i^p is the sensitivity of neuron i in layer p [8], f_i^2 is the derivative of the activation function of output layer neuron i, h_i^1 is the output of previous layer neuron i, i.e., the input of the output layer, and m_2 is the number of neurons of the output layer. The contributions to the error from weights and biases of the hidden layer are:

$$S_i^1 = \dot{f}_i^1 \sum\nolimits_{j=1}^{m_2} S_j^2 w_{ij}^2 \tag{13}$$

$$\frac{\delta E_m}{dw_{ij}^2} = \sum\nolimits_{j=1}^{m_1} S_j^2 x_i \tag{14}$$

$$\frac{\delta E_m}{db_i^1} = S_i^1 \tag{15}$$

where m_1 is the number of neurons in the hidden layer. Finally, parameter updating is achieved by applying the equations:

$$W_v^n(k+1) = \beta W_v^n(k) + (1-\beta)\frac{\delta E_m}{dW^n} \tag{16}$$

$$B_v^n(k+1) = \beta B_v^n(k) + (1-\beta)\frac{\delta E_m}{dB^n} \tag{17}$$

$$W^n(k+1) = \left(1 - \alpha\frac{\lambda}{N}\right)W^n(k) - \alpha W_v^n(k+1) \tag{18}$$

$$B^n(k+1) = B^n(k) - \alpha B_v^n(k+1) \tag{19}$$

where N is the number of outputs, β is the quantity of momentum applied, α is the learning rate and λ is the L2 regularization factor.

3.2 Reference Model

The purpose of the reference model is to define the desired dynamics to the controlled system. By its use, a predefined rising and settling time, overshoot and other response specifications of speed wheels are considered. To do this, a first-order transfer function is used to decouple the otherwise coupled AGV MIMO system:

$$G(s) = \frac{K}{\tau s + 1} \tag{20}$$

To achieve $t_{e98\%} = 2,5s$, $K = 1$ and $\tau = 0,6391$ are determined.

3.3 Neural Network Controller

The neural network controller is also a MLP with: 28 inputs; one hidden layer with 35 neurons (also determined empirically); and an output layer with 2 neurons, which represent the motor voltages u_R and u_L. The activation functions are sigmoid tangent (hidden layers) and linear functions (outputs). For training and online parameter updating, gradient descent with momentum and L2 regularization are used. The input tuple for this network is built by the following vectors:

$$V_{Rref} = \left[v_{Rref}(k), \ldots, v_{Rref}(k-3)\right] V_{Lref} = \left[v_{Lref}(k), \ldots, v_{Lref}(k-3)\right]$$

$$U_R = [u_R(k-1), \ldots, u_R(k-5)]\, U_L = [u_L(k-1), \ldots, u_L(k-5)]$$

$$V_R = [v_R(k-1), \ldots, v_R(k-5)]\, V_L = [v_L(k-1), \ldots, v_L(k-5)]$$

Assuming a good performance of the neural network model [9, 10], that is, $v \approx v_{NN}$, and using the following cost function (control error):

$$E_c = \frac{1}{2}\sum\nolimits_{i=1}^{n}\left(v_{Mref_i} - v\right)^2 \tag{21}$$

the error control is obtained and used in the training (10–15). The propagation of the control error is,

$$\frac{\delta E_c}{du_i} = \sum\nolimits_{j=1}^{m_1} S_j^1 w_{ij}^1 \tag{22}$$

where S_j^1 is the sensitivity of hidden layer neuron j of the model and w_{ij}^1 are the weights of the same layer. The sensitivities of the output layer neurons is,

$$S_i^2 = \frac{\delta E_c}{du_i}\dot{f}_i^2 \tag{23}$$

Using Eqs. (11–19) applied to control error E_c, network parameter are updated. In order to optimize the parameters, a genetic algorithm (GA) was used. The values obtained by the GA are:

$$\beta = 0.015\ \alpha = 0,72\ \lambda = 0.0001 \tag{24}$$

4 Results

Simulations have been carried out using Matlab software with a HP 255 G Notebook PC with a 2.1 GHz processor under Windows OS. The AGV has been subjected to random reference speeds for each wheel. Simulation lasts 75 s; the reference signal frequency varies from 0 s to 10 s, with a velocity range from 0 to 1 m/s. Besides, during the simulation, sudden variations in the payload of the AGV and viscous friction parameter of the DC motors are applied in order to compare the robustness of each control technique.

These variations are 400% extra over their nominal values (40 kg mass and 0.4 Nms the viscous friction coefficient). Robustness is tested for a total mass of 200 kg and coefficient friction equals to 2 Nms.

Results obtained with the proposed neural controller and a PI regulator (one for each wheel) are compared and discussed. The MSE (mean square error), MAE (mean absolute error), MAPE (mean absolute percentage error) and VAR (variance) are used to analyze the performance. Settling time for both controllers is set to 2.5 s. The PI equation is given by,

$$u(t) = K\left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau\right) \tag{25}$$

where $u(t)$ is the control action (V) of the motor, $e(t)$ is the speed error, $v_{Ref} - v$, K is the proportional gain and T_i is the integral time. Tuning is carried out to set $t_{e98\%} \approx 2.5$ s, obtaining the following values for that purpose:

$$K = K_p = 10,3920629 \; \frac{K}{T_i} = K_i = 15,6988750 \tag{26}$$

Figure 3 shows how the PI control (blue line) reaches the velocity reference (green line) for each wheel in a accurate and smooth way while parameters do not vary. Given the overload from $t = 10$ s to $t = 30$ s, the PI speed control performance decreases significantly. Overshoots up to 9% are shown but they are quickly corrected by the PI control. But this is reflected in the trajectory tracking control thus, it should be avoided.

During the period when sudden viscous friction parameter variation occurs, from $t = 40$ s to $t = 60$ s, the performance of the AGV gets much worse. The overshoot at those times ranges from 49.2% to 56%. These results are completely unacceptable because the high probability of losing control. In spite of this, the PI control is able to recover the control and stabilizes the AGV velocity. Hence, PI control would be still

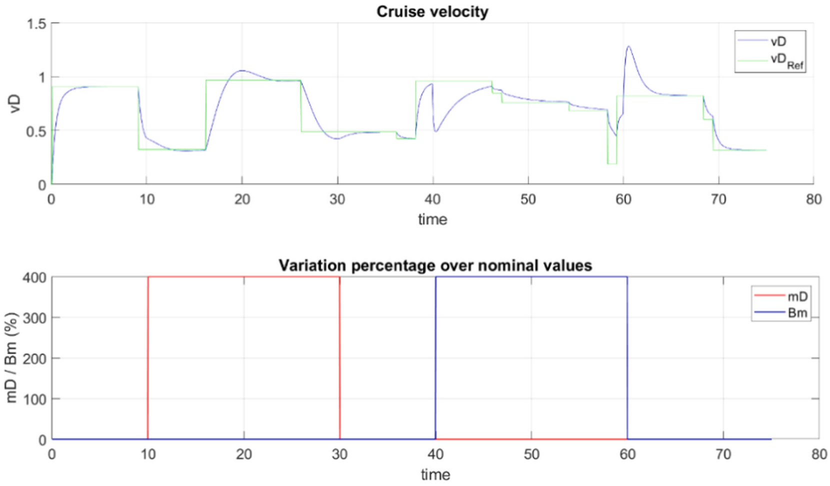


Fig. 3. Cruise velocity with the PI controller.

suitable for viscous friction variations as long as they appear progressively and relatively slow over the time.

With regards to the neural control, two different tests are carried out: one without online adaptation and another with online adaptation n. With online adaptation (Fig. 4), there is an improvement in terms of robustness. This figure shows how the neural control makes the system response (blue) to follow the reference model (red dots). In the event of overload, though overshoots appear, they do not get over 3% of the reference signal (green) but for around 1 s that if reaches 8%.

This effect may be due to the necessity of better training the neural network controller or to the aggressive response of the DC motors since their parameter values have been only estimated. Even though, the performance is still good. In a similar way, when sudden variation of the viscous friction coefficients, oscillations appear at the beginning and end of these disturbances. Although the overshoot does not exceed 3% at those times, the oscillations are more abrupt than with the PI control. Even so, their duration is no longer than 1,3 s. During the rest of the disturbance period, the performance of the neural controller is as expected.

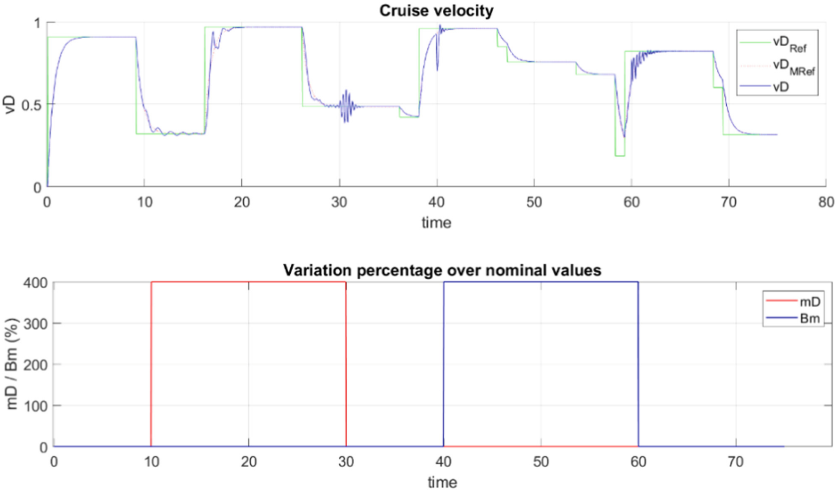


Fig. 4. Cruise velocity with the neural network controller (with online adaptation).

When simulating the controller without online updating, Fig. 5 clearly shows how the controller only follows the reference at the beginning. Given the payload and viscous friction coefficient variations, performance decreases significantly, even after disturbances disappear. This could mean that the neural model still needs a better training. Since the aim of this test was just to compare the effects of the online updating, more extended training has not been carried out.

The PI and neural controllers have been compared according to the metrics previously mentioned and also in terms of computational time (Table 1). The results prove that neural control has a better performance than the PI control. Despite Windows is not a real time

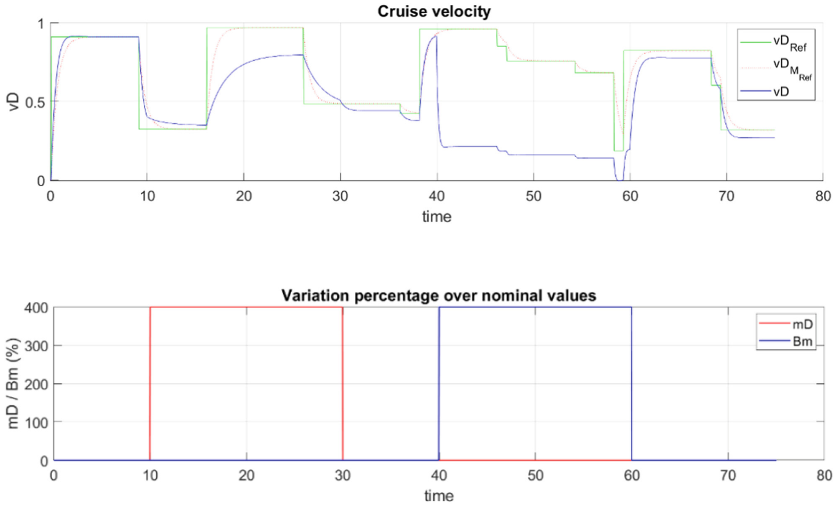


Fig. 5. Cruise velocity for neural network controller (without online adaptation).

OS and Matlab code is normally slow, the result of 21ms is still quick enough to make it feasible to apply the neural controller to this control problem. Furthermore, much faster results could be obtained if the operations were optimized to be run in a real time processor.

Table 1. Metrics for the simulations (v_{Ref} vs v)

Metrics	PI	MRNN
mse	0.0204	0.0121
mae	0.0768	0.0425
mape	0.1271	0.0897
Matlab time	4.6 µs	12.7 ms

5 Conclusions and Future Works

In this paper, a velocity control of an AGV has been developed and simulated. Initially, a dynamic model for the robot is created. Secondly, proportional integral controllers have been tuned to obtain a settling time (98%) of 2.5 s. Finally, a model reference neural network controller is designed, which approximates the dynamics of the AGV and impose a linear and decoupled multivariable system dynamics to the modeled system, with the mentioned settling time of 2.5 s for the motor wheel speeds. Simulations for these controllers have been carried out, submitting them to different reference signals and varying their working conditions. Given the results, it is concluded that the MRNN control is much more robust than PI control and the one that could be implemented in the AGV as it shows an improvement in disturbance rejection.

As future works, we propose to use a dynamic model more complex for this kind of industrial mobile robots and apply this proposal to control wind turbines.

Acknowledgement. This work was partially supported by the Spanish Ministry of Science, Innovation and Universities under MCI/AEI/FEDER Project no. RTI2018–094902-B-C21.

References

1. Sánchez, R., Sierra-García, J.E., Santos, M.: Modelado de un AGV híbrido triciclo-diferencial. Revista Iberoamericana de Automática e Informática industrial **19**(1), 84–95 (2022)
2. Espinosa, F., Santos, C., Sierra-García, J.E.: Transporte multi-AGV de una carga: estado del arte y propuesta centralizada. Revista Iberoamericana de Autom. e Inform. Ind. **18**(1), 82–91 (2021)
3. Sierra-García, J., Enrique, Santos, M.: Mechatronic modelling of industrial AGVs: a complex system architecture. Complexity, **21** (2020)
4. Abajo, M. R., Sierra-García, J. E., Santos, M.: Evolutive tuning optimization of a pid controller for autonomous path-following robot. In: Sanjurjo González, H., Pastor López, I., García Bringas, P., Quintián, H., Corchado, E. (eds.) International Workshop Soft Computing Models in Industrial and Environmental Applications, pp. 451–460. Springer, September 2021. https://doi.org/10.1007/978-3-030-87869-6_43
5. Sierra-García, J.E., Santos, M.: Control of industrial AGV based on reinforcement learning. In: Herrero, Á., Cambra, C., Urda, D., Sedano, J., Quintián, H., Corchado, E. (eds) 15th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2020). SOCO 2020. Advances in Intelligent Systems and Computing, vol. 1268. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-57802-2_62
6. Sierra-Garcia, J.E., Santos, M.: Combining reinforcement learning and conventional control to improve automatic guided vehicles tracking of complex trajectories. Expert Syst. e13076, (2022). https://doi.org/10.1111/exsy.13076
7. Klancar, G., Zdesar, A., et al.: Wheeled Mobile Robotics 1st Edition". Butterworth-Heinemann, Oxford, UK (2017)
8. Hagan, M.T.B., Demuth, H., et al: Neural Network Design 2nd Edtion. Martin Hagan (2014)
9. Slama, A., Errachdi, A., et al.: A Neural Model Reference Adaptive Controller Algorithm for Nonlinear Systems. Tunis El Manar University, ENIT, Department of Electrical Engineering, Tunissa, Automation Research Laboratory (2017)
10. Sierra-García, J.E., Santos, M.: Switched learning adaptive neuro-control strategy. Neurocomputing **452**, 450–464 (2021)

Studying the Use of ANN to Estimate State-Space Variables for MIMO Systems in a NMPC Strategy

Aimar Alonso[1], Asier Zabaljauregi[1], Mikel Larrea[1], Eloy Irigoyen[1](✉), and Javier Sanchís[2]

[1] University of the Basque Country (UPV/EHU), Leioa, Spain
{aalonso198,azabaljauregi001}@ikasle.ehu.eus,
{m.larrea,eloy.irigoyen}@ehu.eus

[2] Universitat Politècnica de València (UPV), Valencia, Spain
javier.sanchis@upv.es

Abstract. The integration of new technologies in advanced control systems continues to be a current challenge today, as control devices gain in performance and new alternatives to existing techniques are offered. This is based on the fact that the techniques from Intelligent Computing allow the development of efficient and robust control systems, without the need to fully understand the dynamics of the plant. On the other hand, the use of predictive control is a reality within the world of the process industry. It is widely used in the industrial environment due to its simplicity and robustness. The objective of this work is to study and assess the possibility of using neural models to reproduce the dynamics of complex multivariable systems, together with their applicability in control strategies that use optimization processes to generate the appropriate control actions and considering operational constraints, as is the case of the nonlinear MPC. In this work, the results obtained along the this study will be presented, where it is analyzed how to include said neuronal model in a previously established non-linear MPC strategy.

1 Introduction

Nowadays, with the continuous progress of technology and the development of new advanced control strategies, both in the field of research and in the industry itself, it has been possible to satisfactorily control complex multivariable (MIMO) systems that previously presented great difficulties in control, due to their complex dynamics, with demanding constraints, strong couplings, their non-linear character or their ease of becoming unstable.

For this reason, model-based predictive control, both in its simplified linear approach (MPC) and in the more complex and realistic non-linear approach (NMPC), have become very popular in recent years, owing to its simplicity and robustness [5,9,10,13,16]. This control strategy makes use of an approximate model of the process to predict its outputs in a supposed future, being able

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 464–473, 2023.
https://doi.org/10.1007/978-3-031-18050-7_45

to respond beforehand to possible changes in references. Based on this model, the strategy calculates the control actions that correspond to the aforementioned future to achieve the control objectives, which are associated with a cost function and are minimized through optimization processes.

On the other hand, in order to address the control of complex systems, control engineers have been inspired by the behavior of the human being to provide “intelligence” to the controllers, hence the techniques of Intelligent Computing or Soft Computing [1–3,7,8,11]. Through this “intelligence”, the control loop is given the ability to learn, adapt, anticipate and predict.

In the present work, in order to combine the capabilities of nonlinear predictive control and Soft Computing techniques, the feasibility of using neural models to reproduce the behavior of MIMO systems will be studied, in order to take advantage of said ability to incorporate it in the processes of calculation and selection of optimal control actions. These models work as black box models, which will be trained and tuned from examples obtained from experimentation with the system to be controlled, in a simulation context. In this case, it will be made with the non-linear mathematical models of the Twin-Rotor model, provided by the company Feedback Instruments Ltd [4]. The model-based nonlinear predictive control strategy will be based on the control algorithms already tested in previous works and provided by Mathworks®.

This paper is organized as follows: After this Introduction section, the Twin-Rotor multivariable nonlinear system is presented. The mathematical model will be presented with its non-linearities and the couplings inherent to it will be emphasized. In the next section, a brief description of the neural networks selected and the training methodology followed in this work will be made. Subsequently, the results obtained in this procedure will be presented and analyzed to validate the proposal. Finally, it will conclude with a summary of the contributions of this work and its possible future lines.

2 MIMO System: Twin-Rotor

The multivariable system chosen for this work is the Twin-Rotor from Feedback Instruments Ltd. This device shows an interesting multivariable control problem that, although it does not have very marked non-linear dynamics, has strong couplings between its variables, which makes it complex to control if the aim is to simultaneously achieve stability, robustness, smoothness and precision objectives [12,14,15].

2.1 Mathematical Model Development

The Twin-Rotor (TR) mechanical platform consists of two rotors placed on a beam together with a counterbalance; while the whole unit is attached to a tower allowing safe control experiments, as schematically shown in Fig. 1.

The composition of non-linear equations for both degrees of freedom (θ for the *pitch* or elevation angle, and φ for the *yaw* or azimuth angle) are derived from the momentum balance in each movement and can be found in the User Manual

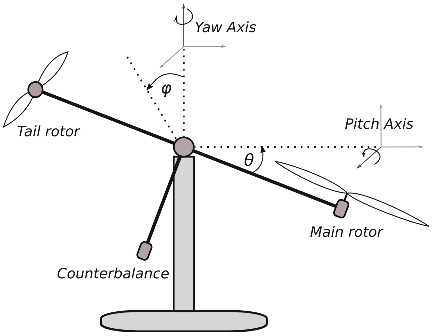


Fig. 1. Twin-Rotor system diagram

provided by Feedback Instruments Ltd. [4]. Rearranging all the equations coming from that user manual, the TR dynamics is presented compactly:

$$\begin{aligned}\frac{d\dot{\theta}}{dt} = \frac{a_1}{I_1}\tau_1^2 + \frac{b_1}{I_1}\tau_1 - \frac{M_g}{I_1}\sin(\theta) + \frac{0.0326}{2I_1}\sin(2\theta)\dot{\varphi}^2 \\ -\frac{B_{1\theta}}{I_1}\dot{\theta} - \frac{k_{gy}}{I_1}\cos(\theta)\left(a_1\tau_1^2 + b_1\cdot\tau_1\right)\dot{\varphi}\end{aligned} \tag{1}$$

$$\frac{d\dot{\varphi}}{dt} = \frac{a_2}{I_2}\tau_2^2 + \frac{b_2}{I_2}\tau_2 - \frac{B_{1\varphi}}{I_2}\dot{\varphi} - \frac{1.75k_c}{I_2}\left(a_1\tau_1^2 + b_1\tau_1\right) \tag{2}$$

$$\frac{d\tau_1}{dt} = -\frac{T_{10}}{T_{11}}\tau_1 + \frac{k_1}{T_{11}}u_1 \tag{3}$$

$$\frac{d\tau_2}{dt} = -\frac{T_{20}}{T_{21}}\tau_2 + \frac{k_2}{T_{21}}u_2 \tag{4}$$

Taking into account the type of models used in the NMPC strategy analyzed in this paper, which are defined in the State Space, all these state variables are ordered as shown in (5). Moreover, considering as inputs u_1 and u_2 the corresponding motor voltages that produce the rotation of the angles, and taking into account as outputs of the TR MIMO system the mentioned angles, pitch and yaw (θ and φ), the input and output matrixes are as follows:

$$x = \begin{pmatrix}\theta \\ \dot{\theta} \\ \varphi \\ \dot{\varphi} \\ \tau_1 \\ \tau_2\end{pmatrix} = \begin{pmatrix}x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6\end{pmatrix}; \qquad u = \begin{pmatrix}u_1 \\ u_2\end{pmatrix}; \qquad y = \begin{pmatrix}\theta \\ \varphi\end{pmatrix} \tag{5}$$

Extracting the *Space State derivatives*, the non-linear model is as follows:

$$\frac{dx_1}{dt} = x_2 \tag{6}$$

$$\frac{dx_2}{dt} = \frac{a_1}{I_1}x_5^2 + \frac{b_1}{I_1}x_5 - \frac{M_g}{I_1}\sin(x_1) - \frac{B_{1\theta}}{I_1}x_2 + \frac{0.0326}{2I_1}\sin(2x_1)\,x_4^2 \\ - \frac{k_{gy}}{I_1}\cos(x_1)\left(a_1x_5^2 + b_1x_5\right)x_4 \tag{7}$$

$$\frac{dx_3}{dt} = x_4 \tag{8}$$

$$\frac{dx_4}{dt} = \frac{a_2}{I_2}x_6^2 + \frac{b_2}{I_2}x_6 - \frac{B_{1\varphi}}{I_2}x_4 - \frac{1.75k_c}{I_2}\left(a_1x_5^2 + b_1x_5\right) \tag{9}$$

$$\frac{dx_5}{dt} = -\frac{T_{10}}{T_{11}}x_5 + \frac{k_1}{T_{11}}u_1 \tag{10}$$

$$\frac{dx_6}{dt} = -\frac{T_{20}}{T_{21}}x_6 + \frac{k_2}{T_{21}}u_2 \tag{11}$$

These final Eqs. (6–11) will be included into the NMPC strategy to develop each of the predictive horizons analyzed during the optimization process.

2.2 Study of Twin-Rotor Dynamics

Several non-linearity effects are present throughout the entire working range of the system, but can only be significantly and clearly exposed around certain working areas. Thus, the appearance of strongly non-linear behaviors is not fundamentally appreciated in either of the two angles, as can be seen in Fig. 2, except when considering very distant work points.

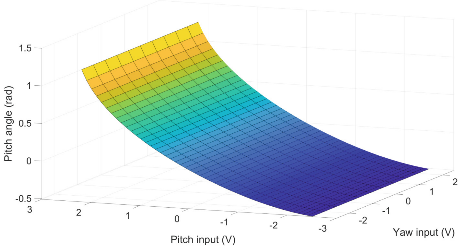


Fig. 2. 3D representations of the workspace for *pitch* angle.

This does not mean that the control of this MIMO system is not still complicated. The strong inherent couplings between the inputs and outputs of the system have an important influence on the control, deteriorating it significantly in certain circumstances, especially when the control actions reach the voltage limits (constraints) by needing to simultaneously control both outputs, such as the one shown in Fig. 3. Although there is no significant decline in its performance when the control is carried out based on the mathematical model, except for a delay in the progress of both angles, it can be seen that the control actions are influenced by this effect (from sample 1000). Obviously, this effect will be greater when working with other approximate models, less precise than its own mathematical model.

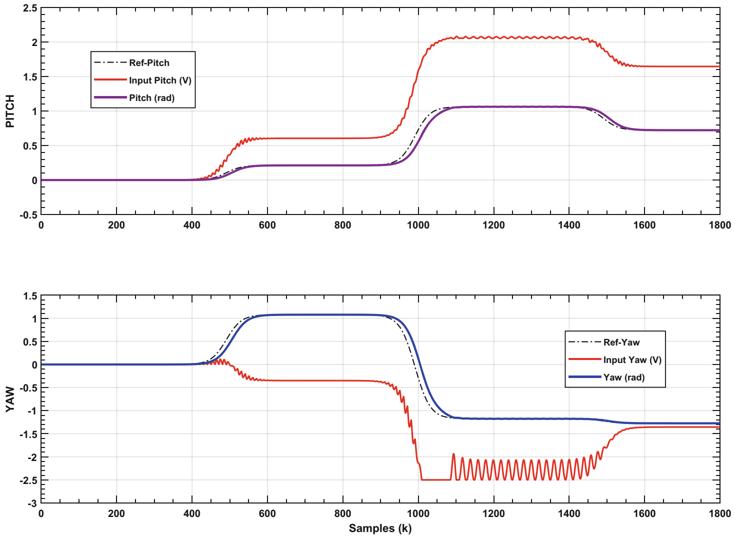


Fig. 3. Coupling influence, at sample 1000 and later, in simultaneous NMPC control of *pitch* and *yaw* angles with the mathematical model. Input voltages in volts (red lines): (Constraints in ± 2.5 V.)

For this reason, addressing the Twin-Rotor system control problem is a significant challenge, even more so if the objective is to model the system with advanced computational tools. Above all, when it is intended to develop said control from a more general approach that does not allow obtaining a precise mathematical model of the MIMO system.

3 Experiments and Results

3.1 ANN Model

ANNs have the proven property of universal approximators [6], so they are perfect candidates to reproduce the dynamics of systems such as the TR. Specifically, one of the most used structures, which is used in this work as it has been validated in previous developments [16], is the NARX structure (Non-linear AutoRegressive with Exogeneous inputs).

To integrate the neural models into the NMPC control scheme which is developed with MATLAB algorithms, they have to calculate and estimate the different discrete state variables used at each sample time (k), according to the following equation, based on Eq. (5):

$$x(k) = \begin{pmatrix} \theta(k) \\ \dot{\theta}(k) \\ \varphi(k) \\ \dot{\varphi}(k) \\ \tau_1(k) \\ \tau_2(k) \end{pmatrix} \simeq \begin{pmatrix} \theta(k) \\ \hat{\dot{\theta}}(k) \\ \varphi(k) \\ \hat{\dot{\varphi}}(k) \\ \tau_1(k) \\ \tau_2(k) \end{pmatrix} \rightarrow \dot{x}(k) = \begin{pmatrix} \dot{\theta}(k) \\ \ddot{\theta}(k) \\ \dot{\varphi}(k) \\ \ddot{\varphi}(k) \\ \dot{\tau}_1(k) \\ \dot{\tau}_2(k) \end{pmatrix} \simeq \begin{pmatrix} \hat{\dot{\theta}}(k) \\ \hat{\ddot{\theta}}(k) \\ \hat{\dot{\varphi}}(k) \\ \hat{\ddot{\varphi}}(k) \\ \dot{\tau}_1(k) \\ \dot{\tau}_2(k) \end{pmatrix} \tag{12}$$

being $\theta(k)$ and $\varphi(k)$ the discrete values of pitch and yaw angles respectively, $\dot{\theta}(k)$ and $\dot{\varphi}(k)$ their corresponding velocities, $\ddot{\theta}(k)$ and $\ddot{\varphi}(k)$ the real angular accelerations, and $\hat{\dot{\theta}}(k)$, $\hat{\dot{\varphi}}(k)$, $\hat{\ddot{\theta}}(k)$ and $\hat{\ddot{\varphi}}(k)$ the estimated velocities and accelarations, calculated with the ANN outputs.

$$\hat{\dot{\theta}}(k) = \frac{\hat{\theta}(k) - \hat{\theta}(k-1)}{T} \tag{13}$$

$$\hat{\ddot{\theta}}(k) = \frac{\hat{\dot{\theta}}(k) - \hat{\dot{\theta}}(k-1)}{T} \tag{14}$$

$$\hat{\dot{\varphi}}(k) = \frac{\hat{\varphi}(k) - \hat{\varphi}(k-1)}{T} \tag{15}$$

$$\hat{\ddot{\varphi}}(k) = \frac{\hat{\dot{\varphi}}(k) - \hat{\dot{\varphi}}(k-1)}{T} \tag{16}$$

Looking for the optimal structure after the training process, a deep study about the ANNs performing based on the number of hidden layer neurons (5...50), along with the input and output delays (1...5), has been made. After carrying out the corresponding experimental tests with different NARX structures, very satisfactory results were achieved with different topologies. Of all the structures evaluated, a synthesized set of results is presented in Table 1 that shows in a summarized way those structures that have offered the best results.

Table 1. The best NARX structures found and the chosen one.

Hidden layer neurons	Input delays	Output delays	MSE 1	MSE 2	MSE 3	MSE 4	MSE 5	MSE average
20	0	1	3.7027e−6	3.6938e−6	3.7547e−6	3.8600e−6	3.5017e−6	3.7026e−6
20	**0:2**	**1:2**	**3.0835e−9**	**2.5469e−9**	**2.3335e−9**	**3.2383e−9**	**3.2699e−9**	**2.8944e−9**
20	0	1:2	2.4869e−8	2.3271e−8	2.6169e−8	2.7861e−8	2.4506e−8	2.5335e−8
30	0	1:2	2.1141e−8	2.1250e−8	2.5131e−8	1.9771e−8	2.2700e−8	2.1997e−8
30	0:1	1:2	1.2658e−6	1.1799e−6	1.0008e−6	1.1818e−6	1.0368e−6	1.1330e−6

For this work, one of the simplest in its configuration and with satisfactory performance was selected, as indicated in Table 1. The selection of an ANN of smaller dimension is critical, since it is later used as a model of the TR system in the NMPC strategy, where the simpler its structure, the less computational cost it causes.

3.2 NMPC Performance with ANN Model

Next, the neural model was included in the NMPC strategy, where it was used to estimate the output values of the pitch and yaw angles, in addition to calculating the values of the successive derivatives of those angles. These first tests were carried out by hybridizing the NMPC strategy with the neural model in the same scheme, as can be seen in the Fig. 4.

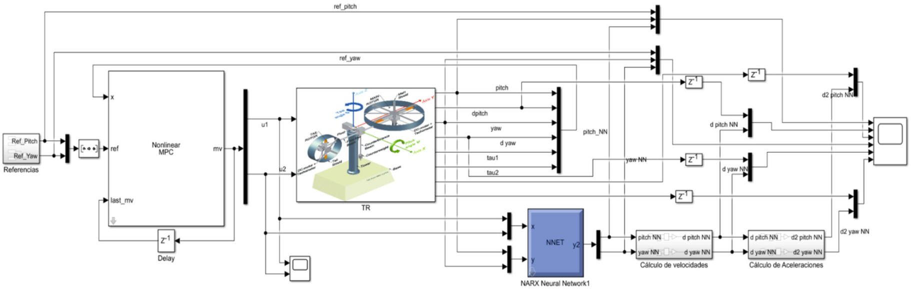


Fig. 4. SIMULINK model to validate the ANN performance.

The results obtained were very satisfactory, showing that the estimated predictions of both angle velocities were achieved with a MSE of $1.12 \cdot 10^{-07}$ and $4.76 \cdot 10^{-07}$ respectively, with a maximum error values of 0.0016 and 0.0040 in the transition zones where both angle variations make the predictions difficult. In the same way, the calculated accelerations reached MSE values of $1.04 \cdot 10^{-06}$ and $1.01 \cdot 10^{-05}$, with maximum errors of 0.0086 and 0.0339, respectively. In addition, once ANN was incorporated into the NMPC control strategy, the control results for pitch and yaw angles were very satisfactory, as shown in Fig. 5.

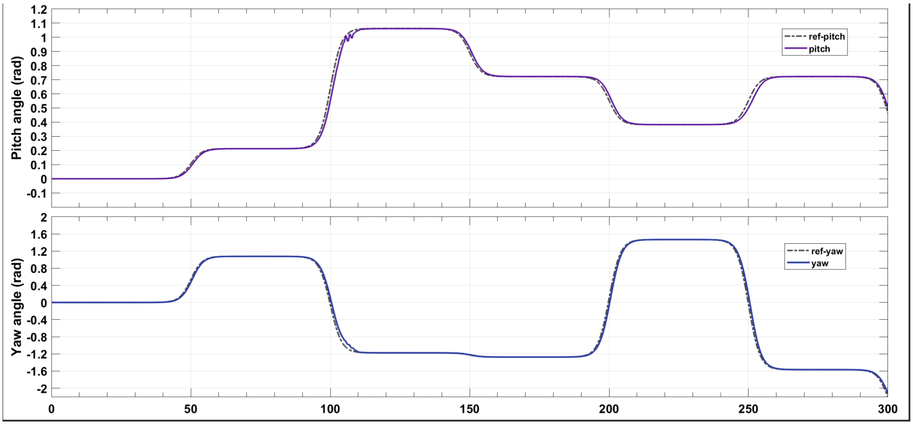


Fig. 5. Pitch and yaw control: Hybrid model with ANN predicting angles and their derivatives.

4 Conclusions

This paper presents a modeling study of MIMO systems using NARX neural networks. The results obtained regarding the performance of the dynamics of a multivariable Twin-Rotor system have been very satisfactory, despite the fact that the strong couplings between its variables make the control actions greatly influence the behavior of the evolution of the pitch and yaw angles.

Furthermore, the use of these ANNs in an MPC control strategy for non-linear systems has been presented and evaluated. The results shown in several experiments have demonstrated the validity of this hybridization, where an ANN has been used to estimate the future outputs of the TR system (pitch and yaw angles) and to calculate the derivatives corresponding to both angles.

As a continuation of this work, the following steps will be aimed to the assessment of this proposal in the real TR platform. A next challenge will be to implement this Intelligent Control strategy in a Hardware in the Loop configuration, to validate a first control prototype.

Acknowledgments. This work comes under the framework of the projects PID2020-120087GB-C22 and PID2020-120087GB-C21 granted by the Ministry of Science and Innovation of the Government of Spain. (AEI/http://dx.doi.org/10.13039/501100011033).

References

1. Abderrahmene, S., Mohammed, C., Abderrahmane, K., Rachida, H.: Neural network NARMA-L2 control of a twin rotor MIMO system. In: 2019 International Conference on Advanced Electrical Engineering (ICAEE), pp. 1–6 (2019). https://doi.org/10.1109/ICAEE47123.2019.9015131.
2. Carrillo-Ahumada, J., Reynoso-Meza, G., García-Nieto, S., Sanchis, J., García-Alvarado, M.A.: Tuning of pareto-optimal robust controllers for multivariable systems. Application on helicopter of two-degress-of-freedom. RIAI J. **12**, 177–188 (2015). https://doi.org/10.1016/j.riai.2015.03.002
3. Dendaluce, M., Valera, J.J., Gómez, V., Irigoyen, E., Larzabal, E.: Microcontroller implementation of a multi objective genetic algorithm for real-time intelligent control. In: Herrero, Á., et al. (eds.) SOCO 2013, vol. 239, pp. 71–80. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-01854-6_8
4. Feedback Instruments, Ltd.: Twin Rotor MIMO System Control Experiments 33–949S. Feedback (2001). https://www.feedback-shop.co.uk/twin-rotor-mimo-system-33-007i.html
5. García, J.J.V., Garay, V.G., Gordo, E.I., Fano, F.A., Sukia, M.L.: Intelligent multi-objective nonlinear model predictive control (iMO-NMPC): towards the 'on-line' optimization of highly complex control problems **39**, 6527–6540 (2012). https://doi.org/10.1016/j.eswa.2011.12.052
6. Hornik, K., Stinchcombe, M., White, H.: Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. Neural Netw. **3**, 551–560 (1990)
7. Larrea, M., Larzabal, E., Irigoyen, E., Valera, J., Dendaluce, M.: Implementation and testing of a soft computing based model predictive control on an industrial controller. J. Appl. Logic **13**(2, Part A), 114–125 (2015). https://doi.org/10.1016/j.jal.2014.11.005, sI: SOCO12
8. Larzabal, E., Cubillos, J.A., Larrea, M., Irigoyen, E., Valera, J.J.: Soft computing testing in real industrial platforms for process intelligent control. In: Snasel, V., et al. (eds.) SOCO 2012. AISC, vol. 188, pp. 221–230. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32922-7_23
9. Lu, C.H., Tsai, C.C.: MIMO neural-network predictive controller design. In: 30th Annual Conference of IEEE Industrial Electronics Society, IECON 2004. vol. 2, pp. 1733–1738 (2004). https://doi.org/10.1109/IECON.2004.1431843
10. Marchante, G., Acosta, A., González, A., Zamarreño, J., Álvarez, V.: Comfort constraints evaluation in predictive controller for energy efficiency. RIAI J. **18**(2), 146–159 (2021). https://doi.org/10.4995/riai.2020.13937
11. Norsahperi, N., Danapalasingam, K.: Particle swarm-based and neuro-based FOPID controllers for a twin rotor system with improved tracking performance and energy reduction. ISA Trans. **102**, 230–244 (2020). https://doi.org/10.1016/j.isatra.2020.03.001
12. Raghavan, R., Thomas, S.: Practically implementable model predictive controller for a twin rotor multi-input multi-output system. J. Control Autom. Electr. Syst. **28**(3), 358–370 (2017). https://doi.org/10.1007/s40313-017-0311-5
13. Schaaf, M.: Hybrid model predictive control of a gravity separator with intermittent product extraction. RIAI J. **17**(3), 318–328 (2020). https://doi.org/10.4995/riai.2020.11957

14. Tiwalkar, R.G., Vanamane, S.S., Karvekar, S.S., Velhal, S.B.: Model predictive controller for position control of twin rotor MIMO system. In: 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), pp. 952–957 (2017). https://doi.org/10.1109/ICPCSI.2017.8391852
15. Ulasyar, A., Zad, H.S.: Robust & optimal model predictive controller design for twin rotor MIMO system. In: 2015 9th International Conference on Electrical and Electronics Engineering (ELECO), pp. 854–858 (2015). https://doi.org/10.1109/ELECO.2015.7394488
16. Viana, K., Larrea, M., Irigoyen, E., Diez, M., Zubizarreta, A.: MIMO neural models for a twin-rotor platform: comparison between mathematical simulations and real experiments. In: Herrero, Á., Cambra, C., Urda, D., Sedano, J., Quintián, H., Corchado, E. (eds.) SOCO 2020. AISC, vol. 1268, pp. 407–417. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-57802-2_39

Control of MIMO Systems with iMO-NMPC Strategy

Asier Zabaljauregi[1,2], Aimar Alonso[1,2], Mikel Larrea[1,2(✉)], Eloy Irigoyen[1,2],
and Javier Sanchis[1,2]

[1] University of the Basque Country (UPV/EHU), Leioa, Spain
{azabaljauregi001,aalonso198}@ikasle.ehu.eus,
{m.larrea,eloy.irigoyen}@ehu.eus
[2] Universitat Politècnica de València (UPV), Leioa, Spain

Abstract. This work presents the development of the intelligent Multi-objective non-linear MPC (iMO-NMPC) strategy applied to MIMO non-linear systems. This strategy has been validated with nonlinear SISO and MISO systems, and its natural evolution is to be validated with MIMO systems. In this work, the MIMO system consists of two nonlinear SISO systems stacked and without any coupling. Since iMO-NMPC is a predictive controller, Neural Networks will be used to predict the dynamics of the MIMO system. A step by step validation procedure is presented, starting with an analysis of the quality of the predictions. Finally, parameter influence of the proposed iMO-NMPC on the control performance is studied.

1 Introduction

Most of real control problems are multi-variable. Typically, there are several process variables involved that should be controlled or manipulated. In industry applications, advanced control techniques instead of decentralized control architectures based on SISO control loops, are being used more and more [1]. Most of these new control architectures need a multiple input multiple output (MIMO) model able to reproduce the multi-variable system dynamics [1]. However, for nonlinear dynamic systems, it is not always feasible the obtainment of useful analytical MIMO models (e.g. first principles models) and other alternatives must be used. The selection of Neural Networks (NN) has a long backing history as system models, since they had been proofed to be universal identifiers [5,6].

One of the widespread control strategies that needs a model is the Model-based Predictive Control (MPC) and its nonlinear variant (NMPC) [2,3,9,11]. The MPC strategy tries to take advantage of the previously know references and future predictions made by the system model to anticipate accurately those changes. MPC or NMPC calculates the control actions to achieve the control objectives, that are associated with a cost function and are minimized thanks to an optimization problem solving. Usually, if a nonlinear model is used the optimization problem changes from a convex quadratic form to a non convex

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 474–483, 2023.
https://doi.org/10.1007/978-3-031-18050-7_46

one, increasing the difficulty to solve the problem and without a guarantee a global optimum was achieved. [2]. The use of NN as a model has been carried out both in MPC and in NMPC strategies [3]. In [13], NN modelling MIMO systems has been extensively studied, analyzing its feasibility to reproduce the dynamics of a twin-rotor, whose controlled variables are significantly coupled.

The intelligent multiobjective nonlinear MPC (iMO-NMPC) strategy, presented in [12], has been tested and validated in industrial controllers under real time conditions [7]. However, the scope of these works is limited to SISO and MISO systems, but the promising results obtained previously justify the natural step to develop the strategy to MIMO systems.

This work presents the application of the iMO-NMPC strategy to control nonlinear MIMO systems. In this first attempt, an uncoupled MIMO system consisting of two nonlinear SISO systems have been selected. Both SISO systems have been yet controlled independently by the iMO-NMPC strategy, proving its ability to control this kind of systems and the feasibility of this strategy.

This paper is organized as follows: after this section, the MIMO system to be controlled is presented emphasizing its characteristic nonlinearities. In the next section, the NN model of the system is obtained and its prediction performance is tested for the selected control horizon. Then, the selection of iMO-NMPC parameters and options is presented. This selection will be used along the experiments to validate the application of the iMO-NMPC to MIMO systems. And finally, the results, the conclusions and possible future works are discussed.

2 MIMO System Presentation

The systems that will conformed the MIMO system presented in this work have been used in diverse papers related to system identification [7,8]. Both systems are benchmarks that present nonlinear behaviours and they are difficult to control with some operating regions where performing small input changes produces steep changes in the output. This behaviour can be observed in Figs. 1 and 2 where the input/output relationship of each system under study is presented.

Since it had been proven they can be controlled independently in a SISO structure with the iMO-NMPC strategy [7], an uncoupled MIMO system can be set up and test the control strategy a step further.

2.1 SISO System: SNL5

This is the first benchmark system that will conform the MIMO system. It had been firstly presented in [4] within a book about advances in Intelligent Control.

$$\begin{aligned} y_1(k+1) = f1(y_1(k), y_1(k-1), u_1(k)) = \frac{1.5 \cdot y_1(k) \cdot y_1(k-1)}{1 + y_1(k)^2 + y_1(k-1)^2} \\ +0.7 \sin\left[0.5\left(y_1(k) + y_1(k-1)\right)\right] \cdot \cos\left[0.5\left(y_1(k) + y_1(k-1)\right)\right] + 1.2u_1(k) \end{aligned} \tag{1}$$

Figure 1 represents the static input-output relation where the nonlinearity of the SNL5 can be seen.

2.2 SISO System: SNL1

This is the second system that will conform the MIMO system. It had been firstly presented in [10] within an article about the usage of Neural Networks in system identification.

$$y_2(k+1) = f2(y_2(k), u_2(k)) = u_2(k)^3 + \frac{y_2(k)}{1 + y_2(k)^2} \tag{2}$$

Figure 2 represents the static input-output relation where the nonlinearity of the SNL1 can be observed.

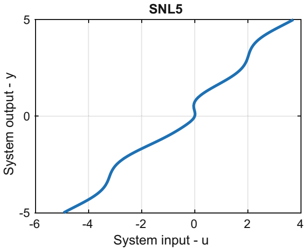


Fig. 1. SNL5 i/o relation

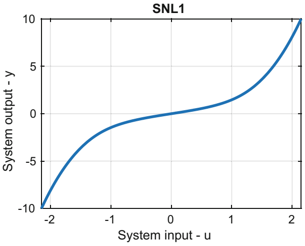


Fig. 2. SNL1 i/o relation

2.3 MIMO System: SNL5 and SNL1

The stacking of the SNL5 and SNL1 will produce the uncoupled 2×2 MIMO system that will be tested under the iMO-NMPC strategy.

$$\begin{bmatrix} y_1(k+1) \\ y_2(k+1) \end{bmatrix} = \begin{bmatrix} f1(\cdot) & 0 \\ 0 & f2(\cdot) \end{bmatrix} \cdot \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix}, \tag{3}$$

where $f1(\cdot)$ and $f2(\cdot)$ are the nonlinear functions in (1) and (2) respectively.

3 NN Model

In this section, the prediction model used by the iMO-NMPC control strategy will be presented. Neural Networks have been extensively used in system identification as they have been proven to be universal approximators [5], therefore different topologies to find the most suitable network for the iMO-NMPC control strategy will be analyzed. On the other hand, a decision between multiple SISO NARX or single MIMO NARX structures must be taken when dealing with MIMO system identification. However, since there is no coupling in the MIMO system under study both approaches are valid.

3.1 Topology Analysis

A training sweep of NARX networks has been carried out. With this methodology, an optimized NARX topology can be found. Notice that the selected topology has to perform well on one step prediction and on a multi-step prediction scenarios. The neural networks have in common the following parameters: Tansig is the hidden layer activation function, whilst the output layer activation function is Purelin; The training algorithm employed is the Levenberg-Mardquart. The variables subject to the sweep are as follows:

- ➔ N. of inputs: [1 to 2].
- ➔ N. of outputs: [1 to 2].
- ➔ N. of neurons of the hidden layer: [6 to 20] (with steps of two neurons to reduce the amount of networks to be trained).
- ➔ N. of input delays: [0 to 1].
- ➔ N. of target delays: [1 to 2].

In the training sweep, each NARX topology configuration is repeated 10 times because single training and validation sets prove to be inconsistent due to the strong statistical dependency. The data presented in the Table 1 are mean values of these 10 repetitions, for the mean square error (MSE) and the maximum absolute error (MaAE) of the best topology.

Table 1. Comparison of the MSE and MaAE produced by the best NARX.

	MSE	max.Abs error
SNL5	5.15e−05	0.1045
SNL1	4.27e−04	0.2153
SNL5+SNL1	0.018	0.4678

It is worth mentioning, that the training sweep methodology is stopped in 20 neurons for the hidden layer. As this is a work in progress, it is not considered worthwhile to go further in the best NN search.

Notice that the MIMO NARX performance is worse than the corresponding SISO performances. It can be explained because the MIMO identification is harder to identify with the same amount of hidden layer neurons. Further, it should be stressed that the MIMO NN has the same internal topology as the SISO NN, thus limiting the hidden layer to 20 neurons. This should not be considered as a drawback but as an approach to show the capability to approximate a MIMO model which inputs could not be decoupled.

After the training process, a deep study about the NARX structure performances based on the number of hidden layer neurons and the input and output delays was made, obtaining the best neural network topologies selected for each system, as shown below in Table 2.

Table 2. Best NARX structures.

	Inputs	Input delays	Outputs	Hidden layer neurons
SNL5	2	3/2	1	20
SNL1	2	3/2	1	10
SNL5+SNL1	4	3/2	2	20

3.2 Prediction Analysis

In the iMO-NMPC strategy, as a MPC control strategy, the main purpose of the model is to predict the system behaviour for a subset of control actions. In this section, the NN model predictions and their suitability for the iMO-NMPC strategy will be analyzed. Firstly, the one step prediction is presented, to follow with the control horizon h step ahead prediction error.

Analysing the one step ahead prediction of the MIMO NARX (Figs. 3 and 4), it can be seen that the MIMO model reproduces closely the SNL5 and SNL1 dynamics.

The prediction performance is satisfactory for the SNL5 and SNL1 systems, showing little degradation along the 10 step ahead predictions. As it was mentioned previously, the MIMO model performs worse than SISO models, but it still shows acceptable predictions for all the topology restrictions that it suffers.

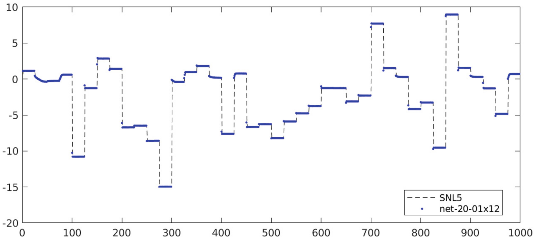


Fig. 3. SNL5 prediction (MIMO NARX)

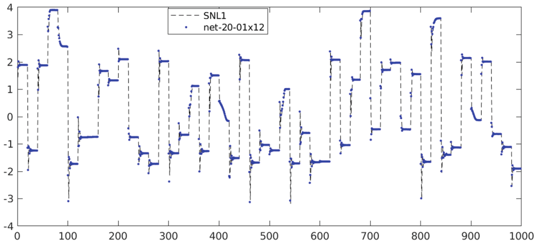


Fig. 4. SNL1 prediction (MIMO NARX)

Table 3. Prediction error for the control horizons H = 1 .. 10

	SNL5		SNL1		SNL5+SNL1 (MIMO)			
	MSE	MaAE	MSE	MaAE	MSE y1	MSE y2	MaAE y1	MaAE y2
H = 1	8,37E–05	1,65E–01	3,85E–08	5,16E–03	1,63E–01	3,05E–01	1,12E+00	1,51E+00
H = 2	5,19E–04	6,53E–01	6,30E–08	5,16E–03	1,66E–01	1,63E–01	1,08E+00	1,31E+00
H = 3	1,17E–03	7,91E-01	6,45E–08	5,16E–03	1,81E–01	1,93E–01	1,09E+00	1,28E+00
H = 4	1,52E–03	7,91E–01	6,49E–08	5,16E–03	1,91E–01	2,55E–01	1,09E+00	1,29E+00
H = 5	1,89E–03	7,91E–01	6,53E–08	5,16E–03	1,99E–01	2,63E–01	1,27E+00	1,31E+00
H = 6	2,29E–03	7,91E–01	6,56E–08	5,16E–03	2,06E–01	2,59E–01	1,31E+00	1,39E+00
H = 7	2,68E–03	7,91E–01	9,67E–05	3,11E–01	2,12E–01	2,72E–01	1,30E+00	1,41E+00
H = 8	3,06E–03	7,91E–01	4,50E–04	5,95E–01	2,17E–01	2,78E–01	1,37E+00	1,40E+00
H = 9	3,45E–03	7,91E–01	1,01E–03	7,50E–01	2,22E–01	2,80E–01	1,36E+00	1,40E+00
H = 10	3,84E–03	7,91E–01	1,64E–03	7,91E–01	2,25E–01	2,83E–01	1,36E+00	1,40E+00

Nevertheless, the control of these systems shows good performance as it will be presented in the results section (Table 3).

4 iMO-NMPC Parameters (Control Strategy)

Inherent to all standard MPC controllers, the iMO-NMPC strategy presented in [12] includes soft computing techniques to carry out the optimization process required for the calculations of the manipulated variables under a control horizon. This novel strategy states the model predictive control problem as a multi-objective optimization one, where each objective is evaluated separately and the selection of the optimal solution must be taken from a Pareto Front, using any decision making technique. With this strategy, control problems for SISO and MIMO systems can be solved by using the corresponding models. The intelligent techniques provided, such as artificial neural networks, allow working with system models where mathematical representation is not possible. Besides, evolutionary computing techniques are incorporated, such as Genetic Algorithms, to carry out a global search for optimal control solutions that meet the established objectives, which are represented in their corresponding Objective Functions. To make use of the iMO-NMPC strategy, a set of parameters must be tuned, which can be chosen depending on the context and the control problem to be solved. Among all of them, the main parameters evaluated in this work and their corresponding values are the following:

➔ Prediction Horizon = Control Horizon = 4
➔ Initial Population = 200
➔ Number of Generations = 200

Furthermore, the objective functions, classic in the Control field, are:

$$J_1 = min\left((ref_1(k) - y_1(k))^2\right) \tag{4}$$

$$J_2 = min\left((ref_2(k) - y_2(k))^2\right) \tag{5}$$

$$J_3 = min\left((u_1(k) - u_1(k-1))^2\right) \tag{6}$$

$$J_4 = min\left((u_2(k) - u_2(k-1))^2\right) \tag{7}$$

where J_1 and J_2 are the objectives assigned to each input/output pair, $ref_i(k)$, $y_i(k)$ are the i-th reference and the output of the MIMO system respectively. J_3 and J_4 are the objectives related to the inputs of the system, where $u_i(k)$ is the control action calculated by the iMO-NMPC controller. All the objectives are calculated for the control horizon h where $k = 1..h$.

The 4 objectives are disaggregated to show the capability of the iMO-NMPC to tackle with problems where the objectives are clearly opposing each other and their aggregation could be a challenging task. After the optimization process, a decision making phase is needed to select the most appropriate control action from the Pareto front. So, the decision maker is an algorithm which selects the solution with least distance to the origin.

5 Results

The experiments presented in the following sections cover the steps to ensure the proper behaviour of the control strategy for the proposed MIMO system.

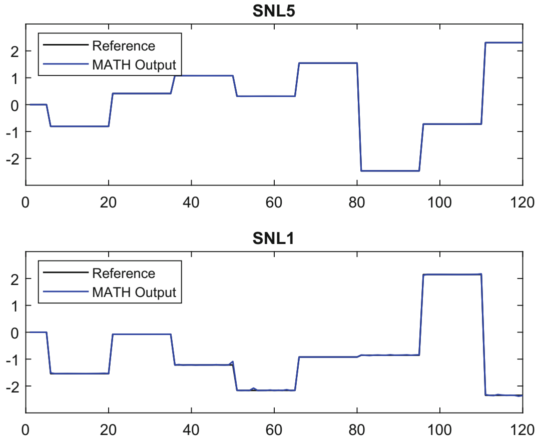


Fig. 5. Control results with mathematical model prediction.

5.1 Using Mathematical Model with iMO-NMPC

Firstly the iMO-NMPC strategy has been tested with the predictions of the non-linear mathematical equations presented in (3). This is the non-realistic scenario where the system dynamics are perfectly known and described by mathematical equations. In this work, the results obtained in this experiment are used as benchmark.

Figure 5 shows the control results, where both references in black are tracked by the system outputs in blue.

5.2 Using NN Model with the iMO-NMPC

The second experiment makes use of the NN model to predict system input/output relation within the iMO-NMPC strategy. The topology and NN model performance has been analysed in the third section.

The control results, using the NN model, are presented in the Fig. 6, where both references (*SNL5_ref* and *SNL1_ref* in black) are tracked by the system outputs (*SNL5_yk* and *SNL1_yk* in red).

5.3 Comparison of Prediction Models: Math. vs. NN

For comparison purposes, the mean squared error values has been calculated from both experiments (Table 4). The results show little variance in the overall performance. It should be remarked that although the table shows SNL5 and SNL1 separately, the NN model used in the experiments is the MIMO NN.

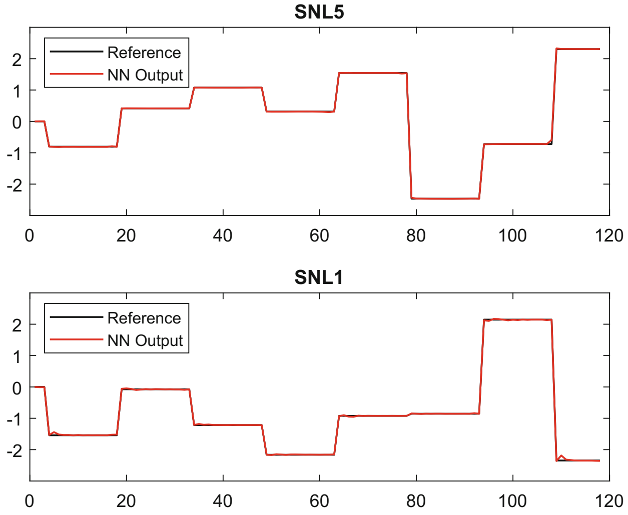


Fig. 6. Control results with Neural Network model predictions.

Table 4. Comparison of the mean squared error of each experiment.

	Math	NN
yk SNL5	1.8394E–06	1.7771E–04
yk SNL1	2.6938E–04	4.3003E–04

6 Conclusions

This work presents a first attempt to control MIMO systems with the iMO-NMPC strategy. This strategy was validated previously with SISO and MISO systems, therefore moving one step ahead towards control of MIMO systems is desired. The uncoupled MIMO system chosen, consist of the stacking of two previously tested SISO systems. These systems are benchmark systems that have been used previously for testing different non linear control algorithms.

The iMO-NMPC strategy needs a dynamic model of the system under control to perform the prediction of the output signals. In this work, the initial NN model topology has been analyzed fulfilling the requirements for a good control. This topology search has been deliberately stopped early, accepting the possible worse performance of the MIMO model to concentrate the effort in the control strategy adaptation. Nevertheless the NN models present acceptable performance and good prediction capabilities.

The first experiments carried out and presented in the results section, produce good control performance for the nonlinear MIMO system. Two scenario has been tested, on one hand the mathematical model has been used for predict the system outputs. On the other hand, the NN has been used within the controller, showing good performance. This initially validates the utilization of the iMO-NMPC strategy with MIMO systems.

As future research lines, implementation of the strategy to control a real system with the iMO-NMPC strategy in industrial HW will be studied. Another future line lies in the improvement of the NN model and the application of more complete topology search methodology.

Acknowledgments. This work comes under the framework of the projects PID2020-120087 GB-C22 and PID2020-120087 GB-C21 granted by the Ministry of Science and Innovation of the Government of Spain. (AEI / http://dx.doi.org/10.13039/501100011033).

References

1. Albertos, P., Sala, A.: Multivariable Control Systems. Springer, London (2004). https://doi.org/10.1007/b97506
2. Camacho, E.F., Bordons, C.: Nonlinear model predictive control: an introductory review. In: Findeisen, R., Allgöwer, F., Biegler, L.T. (eds.) Assessment and Future Directions of Nonlinear Model Predictive Control. LNCIS, vol. 358. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72699-9_1

3. Camacho, E.F., Bordons, C.: Model Predictive Control. Springer, Singapore. 2nd edn. (2007). https://doi.org/10.1007/978-981-13-0083-7
4. Harris, C.: Advances in Intelligent Control. CRC Press, Boca Raton (1994)
5. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Netw. **2**, 359–366 (1989)
6. Jagannathan, S., Lewis, F.: Identification of nonlinear dynamical systems using multilayered neural networks. Automatica **32**(12), 1707–1712 (1996)
7. Larrea, M., Larzabal, E., Irigoyen, E., Valera, J., Dendaluce, M.: Implementation and testing of a soft computing based model predictive control on an industrial controller. J. Appl. Log. (2014). https://doi.org/10.1016/j.jal.2014.11.005
8. Larrea, M., Irigoyen, E., Gómez, V.: Adding nonlinear system dynamics to Levenberg-Marquardt algorithm for neural network control. In: Diamantaras, K., Duch, W., Iliadis, L.S. (eds.) ICANN 2010. LNCS, vol. 6354, pp. 352–357. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15825-4_47
9. Marchante, G., Acosta, A., González, A., Zamarreño, J., Álvarez, V.: Comfort constraints evaluation in predictive controller for energy efficiency. RIAI J. **18**(2), 146–159 (2021). https://doi.org/10.4995/riai.2020.13937
10. Narendra, K.S., Parthasarathy, K.: Identification and control of dynamical systems using neural networks. IEEE Trans. Neural Netw. **1**, 4–27 (1990)
11. Schaaf, M.: Hybrid model predictive control of a gravity separator with intermittent product extraction. RIAI J. **17**(3), 318–328 (2020). https://doi.org/10.4995/riai.2020.11957
12. Valera, J., Gómez, V., Irigoyen, E., Artaza, F., Larrea, M.: Intelligent multi-objective nonlinear model predictive control (IMO-NMPC): towards the 'on-line' optimization of highly complex control problems. Expert Syst. Appl. **39**(7), 6527–6540 (2012). https://doi.org/10.1016/j.eswa.2011.12.052
13. Viana, K., Larrea, M., Irigoyen, E., Diez, M., Zubizarreta, A.: MIMO neural models for a twin-rotor platform: comparison between mathematical simulations and real experiments. In: Herrero, Á., Cambra, C., Urda, D., Sedano, J., Quintián, H., Corchado, E. (eds.) SOCO 2020. AISC, vol. 1268, pp. 407–417. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-57802-2_39

Optimization of Trajectory Generation for Automatic Guided Vehicles by Genetic Algorithms

Eduardo Bayona[1,2](✉), Jesús Enrique Sierra-García[1,2], and Matilde Santos[3]

[1] Electromechanical Engineering Department, University of Burgos, Burgos, Spain
{ebayona,jesierra}@ubu.es
[2] Michelin España-Portugal, Aranda de Duero, Spain
[3] Institute of Knowledge Technology, Complutense University of Madrid, Madrid, Spain
msantos@ucm.es

Abstract. Continuous improvement of industrial and production processes for efficiency optimization has led to an increase in the need for automatization. Automatic guided vehicles (AGV) are key transport elements when it comes to fulfilling this function, as well as the development and improvement of their navigation and positioning systems. Thus, in this work the use of a Soft Computing evolutive technique, genetic algorithms (GA), is proposed in order to obtain the optimal parameters of a trajectory generation method. An occupancy map model for the environment layout is used in order to check collision events with the AGV. Different scenarios have been simulated to optimize the trajectory length avoiding collisions if possible. Simulation results show that the algorithm is able to minimize the length of the path successfully.

1 Introduction

Production processes are constantly changing and being studied for their optimization in terms of both productivity and safety. From this need for continuous improvement, it arises the intention to automate parts of a process where the physical action of a person is not directly necessary. A key aspect of the production processes is the intralogistics, thus there are different types of solutions that allow to automate the internal transfer of raw material, products, merchandise, etc. [1, 2].

Particularly, automatic guided vehicles (AGV) fulfill this function, and they have a diversified field of applications that is growing over time. Business sectors in which AGV can be used include container terminals, flexible manufacturing systems, logistics warehouses, agriculture, military operations, health management, among others [3, 4].

AGV are mobile robots capable of self-direction within an environment following physical or virtual paths, and they provide a notable improvement in the production processes, reducing costs, increasing production and optimizing safety during them [5].

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 484–492, 2023.
https://doi.org/10.1007/978-3-031-18050-7_47

Through the use of different movement sensing and monitoring technologies, AGVs are capable of performing different functions in parallel to their main movement functions, such as loading and unloading goods or interacting with other agents within their operating environment, such as humans or other robots. Using different navigation and guidance sensors to obtain information of their state, AGVs are able to use controllers (usually PIDs) to follow the paths and trajectories correctly [6]. Some previous works have focused on the definition of motions curves [7] for these trajectories.

In order to generate these paths to be followed by the AGV, numerical methods based on Frenet-Serret formulas are used in this paper in order to work with their mathematical expression. This method fits a fourth and fifth-order polynomial smooth continuous curve along a set of waypoints previously defined. These way-points determine the final shape of the trajectory. Reducing the length of the path, thus reducing the travel time, has a direct impact on the production time and therefore on the productivity of the processes where automated transport solutions are implemented using these AGVs. This paper is focused on the optimization by genetic algorithms of the path to be followed by the AGV to reach a destination avoiding obstacles.

Indeed, these evolutionary techniques have been successfully used in other previous works. In [8] GA are used to optimize the rudder control of an unmanned boat. In [9], they are used to improve the tracking of the trajectory of a mobile robot. A tuning optimization method is proposed in [10] for an AGV PID controller that commands the robot to follow different trajectories.

Soft Computing Techniques have been successfully applied to model and control physical systems [11, 12]. In this work, genetic algorithms are used to optimize the AGV's path from a starting point to an end point. The algorithm considers if there is a collision event between the AGV and an element in the physical environment. In this case, the fitness function is designed to achieve two different aims: to minimize the total length of the path between starting and end points, and to prevent the AGV from colliding with the occupancy map. Therefore, the trajectory with the lowest fitness value is the shortest possible one that avoids collisions. This may be useful for the automation of the path generation process in an AGV implementation in an industrial scenario.

The rest of the paper is organized as follows. Section 2 describes the model used to simulate the AGV workspace using an occupancy map. Section 3 explains the methodology and the genetic algorithm implemented for the optimization of the trajectory generation process. In Sect. 4 the results are summarized and discussed. The last section presents the conclusions and future works.

2 Environment Layout Model

The definition of the environment layout for the algorithm to consider collisions is carried out by means of a binary occupancy map. This is used to model and visualize the robot workspace, creating an spatial representation of the location of the obstacles as a discrete grid. The occupancy grids are used in robotics algorithms for path planning and mapping applications. Each cell of the grid has a value representing the occupancy status of that cell. Occupancy grid formal representation is shown in (1)

$$a_{GRID} = \begin{pmatrix} a_{11} & a_{12} \dots a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{pmatrix} \tag{1}$$

where (n, m) are integers representing the dimensions of the workspace. If a location is occupied in the map, it is represented by a 1, otherwise by a 0, as given in (2).

$$a_{nm} \in \mathbb{N} : a_{nm} = \begin{cases} 1\, Ocuppied\ Location \\ 0\, Free\ Location \end{cases} \tag{2}$$

This map is used as a constraint for the trajectory definition, comparing the path points with each occupancy position of the map. When one way point of the trajectory ($t(x, y)$) meets an occupied cell of the map (a_{nm}), it will be consider as a collision event, and the trajectory is not feasible.

$$For\, x, y \in t | \begin{cases} a_{xy} = 0 \Rightarrow No\, Collision \\ a_{xy} = 1 \Rightarrow Collision \end{cases} \tag{3}$$

3 Description of the Trajectory Optimization Process Based on Genetic Algorithms

3.1 Optimization Methodology

The purpose of the trajectory generation is to obtain the shortest smooth continuous path that fits a set of waypoints without collision events. In this work the set of predefined waypoints are the starting point and the destination point. The intermediate waypoints of the trajectory are optimized by the genetic algorithm so that the length of the trajectory, l_t, is minimized avoiding obstacles. The architecture of the trajectory generation method is shown in Fig. 1. The algorithm is able of generating multiple intermediate waypoints, but in this work we will only consider test scenarios with a single additional waypoint.

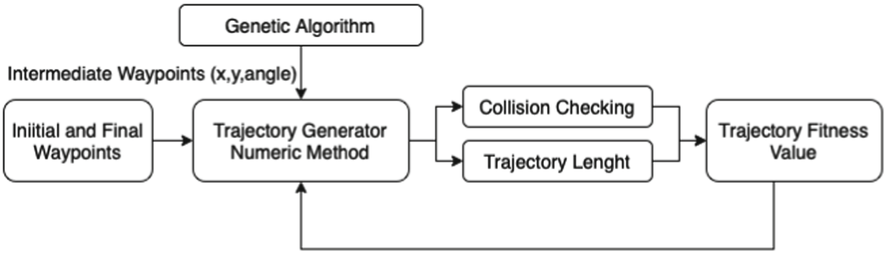

Genetic Algorithm
Intermediate Waypoints (x,y,angle)
Iniitial and Final Waypoints
Trajectory Generator Numeric Method
Collision Checking
Trajectory Lenght
Trajectory Fitness Value

Fig. 1. Trajectory generation methodology

The trajectory generated is given by the Frenet's equations [14], that is a planar curve in $\mathbb{R}^2$ where the tangent vector T and the normal vector N satisfies (4). These equations are called the Frenet equations of the trajectory [14].

$$\begin{cases} T'(s) = k(s)N(s) \\ N'(s) = -k(s)T(s) \end{cases} \tag{4}$$

where s is the distance from the origin at each point, k is the curvature of the trajectory at the distance s, $T(s)$ is the tangent vector at distance s, $N(s)$ is the normal vector at distance s, and the symbol $'$ denotes the perpendicular vector.

Based on the definition of the Frenet's curve, the arc length of the trajectory can be calculated in cartesian coordinates, where the path is a planar curve in $\mathbb{R}^2$ with the form $y = f(x)$, where f is continuously differentiable. The length of each infinitesimal segment of the curve can be given by (5):

$$ds = \sqrt{dx^2 + dy^2} = \sqrt{1 + \left(\frac{dy}{dx}\right)^2} dx \tag{5}$$

This leads us to the form of the arc length s of the calculated trajectory based on (5), that is given by (6):

$$s = \int_{(x_i, y_i)}^{(x_f, y_f)} \sqrt{1 + \left(\frac{dy}{dx}\right)^2} dx \tag{6}$$

where (x_i, y_i) and (x_f, y_f) are the predefined initial and final points of the trajectory.

3.2 Genetic Algorithm

Genetic algorithms are a computational evolutive technique used for optimization purposes. It is based on the evolution of an initial population of individuals. The right codification and the genetic operators (mutation and crossover) are key to the convergence of the algorithm to the best solution [10].

Possible solutions of this problem are defined as an array of 3 values: [x, y, θ], being x and y the rectangular coordinates of the intermediate waypoint and θ the input and output angle of the trajectory at this point. In this work only one intermediate waypoint is used, although multiple waypoints could be used if necessary. The first two elements can be any integer within the dimensions of the occupation map and the third one is any angle between –π/2 and π/2. These conditions can be formalized by Eqs. (7–9).

$$x \in [0, a_n] \tag{7}$$

$$y \in [0, a_m] \tag{8}$$

$$\theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \tag{9}$$

Considering that $x = r\cos\theta$ and $y = r\sin\theta$ in coordinate polars, the length of the considered solution trajectory curve will be given by (10):

$$l_t = \int_{\theta i}^{\theta f} \sqrt{r^2 + \left(\frac{dr}{d\theta}\right)^2} d\theta \tag{10}$$

Thus, l_t is considered by the algorithm to optimize the generated trajectory. In order to guide the genetic algorithm towards the optimal solution, a fitness function is used to include the collision factor into the trajectory length optimization problem.

This fitness function is defined in (11) as the trajectory length times the collision coefficient C. . This coefficient is 1 when there is not a collision event and 1000 when and event takes place, making not feasible those trajectories that even being shorter than others, have a collision event with the occupancy map. Equations (11–12) define the fitness function and the value of the collision coefficient.

$$f_{fitness} = l_t \cdot C \tag{11}$$

$$C = \begin{cases} 1, \textit{No collision Event} \\ 1000, \textit{ Collision Event} \end{cases} \tag{12}$$

4 Results and Discussion

The simulation of the trajectory generation method and the optimization have been carried out using Matlab/Simulink software. An occupancy map has been set to check the collision event variable. Initial and ending points and their respective exit and arrival angles have been previously set. Intermediate point coordinates and angle are the variables used by the genetic algorithm to optimize the length of the trajectory depending on the constraints. The configuration of the genetic algorithm is as follows. The size of the population is 50 individuals, the crossover fraction is set to 0.8, the mutation rate is 0.2, and the elite size is 5%. The individuals are randomly initialized. On the other hand, data used in this study are available on demand.

Three different scenarios have been simulated in order to check different behaviors of the genetic algorithm optimizing the trajectory generation function. The first one will not limit the position of the intermediate waypoint, being all the occupancy map possible positions for this point. Secondly, constraints for the x and y coordinates within a zone of the occupancy map have been set. These second scenario constraints are defined in order to show how the algorithm works for a trajectory with an intermediate waypoint in a clearly separated position from the one obtained in the first scenario, and to check the differences. Finally, the third scenario simulates a case in which constraints does not allow to find an intermediate point where the trajectory do not have a collision event. The purpose of these experiments is to show how an incorrect definition of the constraints can negatively affect the fitness function value.

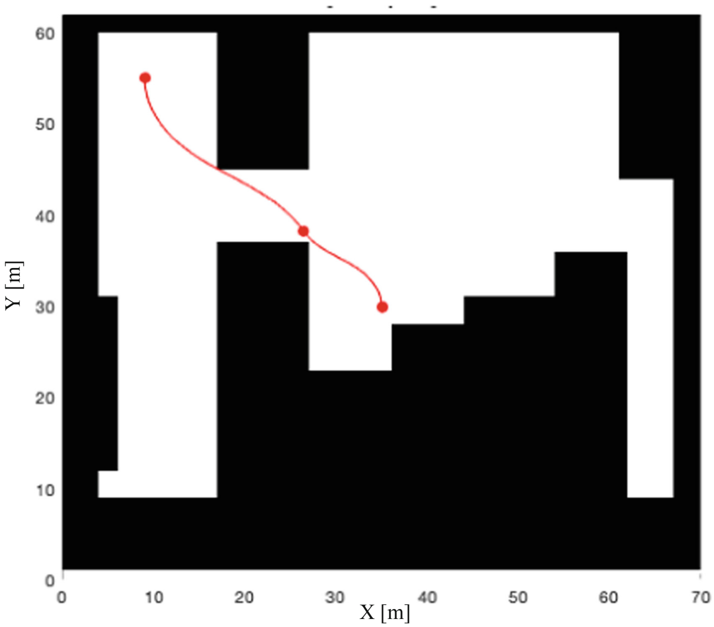


Fig. 2. Scenario 1: Optimized trajectory without area constraints

Figure 2 shows the results of the optimized trajectory for the first scenario. In this case, no boundaries for the intermediate point have been used. The optimized trajectory and the intermediate point given by the genetic algorithm is shown in red as well as the initial and final points. It is also defined the optimal angle for the trajectory in the intermediate point calculated.

Figures 3 and 4 show the results for the second and third scenarios, respectively. In these cases the position of the intermediate point has been constrained and correspondingly, the shape of the trajectory adapts to the shortest path according to these parameters. In second scenario, the possible intermediate points zone has been set between $x \in [25, 30]$ and $y \in [43, 50]$ and is represented by a blue square in the occupancy map.

In Fig. 3, a blue square represents the area used as a constraint for the position of the intermediate point for the trajectory. With this constraint, the algorithm seeks the solutions which position is within the space in the blue square. In this case the algorithm finds a different trajectory to fulfil the requirements, reaching a higher curvature and a longer path that in the case of the free positioned middle point. This is significative when considering different AGV models with different possible curvature movement capabilities as depending of the conditions for the waypoint positions, this parameter can change significantly.

Figure 4 shows the result for the third scenario. In this simulation, the restricted area for the intermediate point of the trajectory makes impossible to the genetic algorithm to find a path without colliding with the occupied positions of the map. In this case, the shortest path is also selected, although the fitness function given by the algorithm will have a very high value according to this situation. In this third scenario, the possible

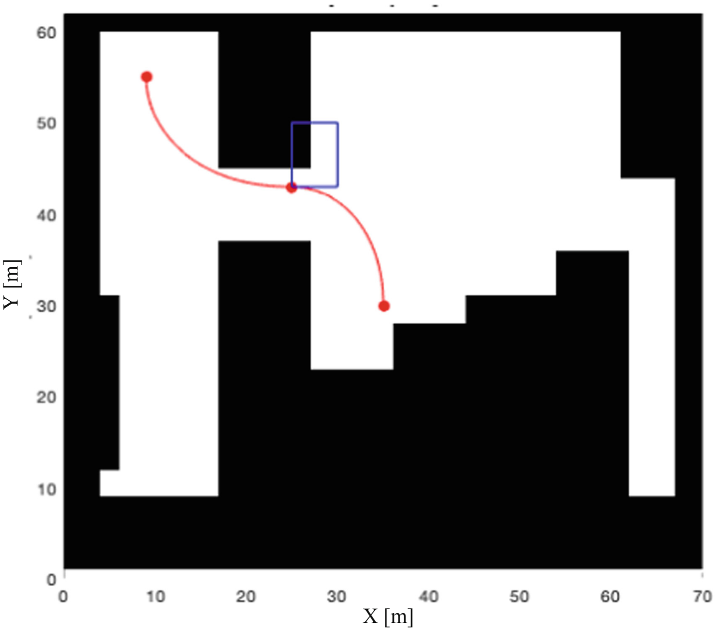


Fig. 3. Scenario 2: Area-conditioned optimized trajectory

intermediate points zone has been set between $x \in [19, 26]$ and $y \in [30, 36]$. Again, a blue square represents the conditioned area for the middle point and a blue circle marks the point in which the function recognizes a collision event.

$$for\, t(25.85, 37) \Rightarrow a_{xy} = 1 \Rightarrow Collision \tag{13}$$

In this case, is in the blue mark at $\left[x, y\right] = [25.85, 37]$, where the condition of the collision event is met.

For each combination of the system parameters, the best individual is obtained by the genetic algorithm by means of optimizing the fitness function, returning the fitness value. If there is a collision event during the trajectory, the fitness function value will not be acceptable, and the associated scenario will not be valid for a suitable AGV trajectory.

The results of Table 1 show that depending on the constraints that are set for the trajectory path, the set of optimized parameters returned by the GA gives different performance. In fact, for cases in which the constraints are very restrictive, the genetic algorithm is unable to find the right tunning parameters for an optimal trajectory without colliding with the occupancy map. When no constraints are specified, the algorithm finds the lower fitness value for the trajectory.

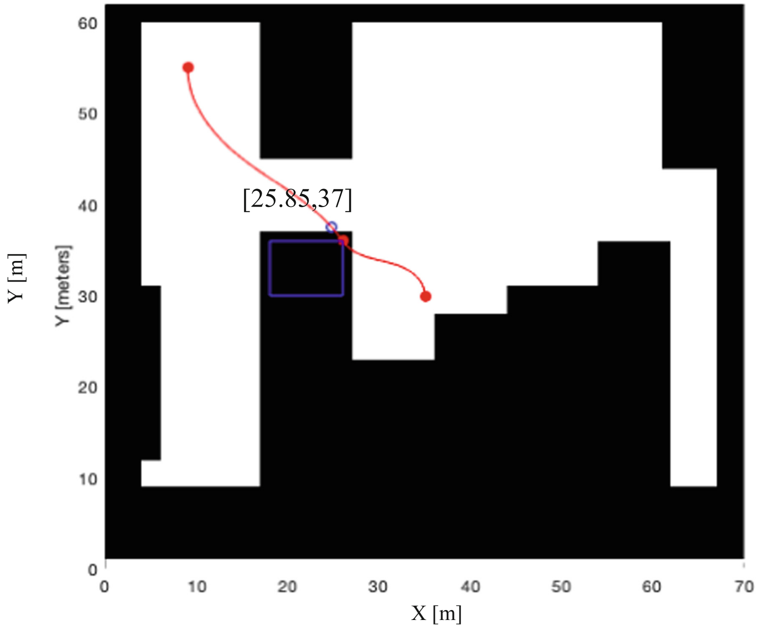


Fig. 4. Scenario 3: Area conditioned with collision length optimized trajectory

Table 1. Optimization results for different simulation scenarios

Scenario	Mid-Point Boundaries	Best Individual	Fitness Value	Valid
1	None	[26.4,38.27, −0,98]	37.54	✓
2	x ∈ [25,30], y ∈ [43,50]	[25,43,0]	39.48	✓
3	x ∈ [19,26], y ∈ [30,36]	[26,36, −0.96]	37846	X

5 Conclusions and Future Works

In this work, genetic algorithms have been applied to optimize the length and to avoid collision events in a trajectory generation function for automatic guided vehicles. The collision events are tested between the trajectory points and path segments and a binary occupancy map. An intermediate waypoint of the trajectory is the parameter used for the algorithm for the optimization.

Two cases of different constraints are compared and analyzed drawing some conclusions.

- Constraints of the intermediate waypoints of the trajectory are an important decision, as it directly affects the length and curvature of the resulting trajectory in order to avoid the collision events.

- This application fits perfectly for the automatization of trajectory generation when implementing AGV in industrial processes or modifying the paths in existing implementations as it gives a way to automatically find the optimal trajectory for a guided vehicle from a particular constraints and parameters.

As future works, we propose the improvement of the methodology in order to consider further parameters and conditions to evaluate the better solution. Also, the collision events detection, must be updated to consider the physical model of the AGV used to follow the optimal trajectory.

References

1. Echeverri Estrada, J.M., Escobar Murcia, P.A.: Caracterización de un AGV (vehículo guiado automáticamente) en el sistema de manufactura flexible, caso Centro Tecnológico de Automatización CTAI de la Pontificia Universidad Javeriana (2012)
2. Vis, I.F.: Survey of research in the design and control of automated guided vehicle systems. Eur. J. Oper. Res. **170**(3), 677–709 (2006)
3. Bechtsis, D., Tsolakis, N., Vlachos, D., Iakovou, E.: Sustainable supply chain management in the digitalisation era: the impact of automated guided vehicles. J. Clean. Prod. **142**, 3970–3984 (2017)
4. Bostelman, R., Messina, E.: Towards development of an automated guided vehicle intelligence level performance standard. Autonom. Ind. Veh. Lab. Factory Floor, 1–22 (2016)
5. Espinosa, F., Santos, C., Sierra-Garcia, J.E.: Multi-AGV transport of a load: state of art and centralized proposal. Revista Iberoamericana de Automática e Informática Industrial **18**(1), 82–91 (2021)
6. Zhang, H., Wu, D., Yao, T.: Research on AGV trajectory tracking control based on double closed-loop and PID control. In: Journal of Physics: Conference Series, vol. 1074, no. 1, p. 012136. IOP Publishing, September 2018
7. Farouki, R.T.: Curves from Motion, Motion from Curves. California Univ, Davis Dept. of Mechanical and Aeronautical Engineering (2000)
8. Larrazabal, J.M., Peñas, M.S.: Intelligent rudder control of an unmanned surface vessel. Expert Syst. Appl. **55**, 106–117 (2016)
9. Alouache, A., Wu, Q.: Genetic algorithms for trajectory tracking of mobile robot based on PID controller. In: 2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP), pp. 237–241. IEEE, September 2018
10. Abajo, M.R., Sierra-García, J.E., Santos, M.: Evolutive tuning optimization of a PID controller for autonomous path-following robot. In: Sanjurjo González, H., Pastor López, I., García Bringas, P., Quintián, H., Corchado, E. (eds.) 16th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2021). SOCO 2021. Advances in Intelligent Systems and Computing, vol 1401. Springer, Cham.https://doi.org/10.1007/978-3-030-87869-6_43
11. Sierra-García, J.E., Santos, M.: Switched learning adaptive neuro-control strategy. Neurocomputing **452**, 450–464 (2021)
12. Santos, M., Cantos, A.J.: Classification of plasma signals by genetic algorithms. Fusion Sci. Technol. **58**(2), 706–713 (2010)
13. Kühnel, W.: Differential geometry. student mathematical library. Am. Math. Soc. Providence, RI, **16** (2002)
14. Alencar, H., Santos, W., Neto, G.S.: Differential Geometry of Plane Curves, vol. 96. American Mathematical Society (2022)

Special Session on Soft Computing Applied to Renewable Energy Systems

Complementing Direct Speed Control with Neural Networks for Wind Turbine MPPT

Eduardo Muñoz Palomeque[1](✉), Jesús Enrique Sierra-García[2], and Matilde Santos[3]

[1] Complutense University of Madrid, Madrid, Spain
edumun04@ucm.es

[2] Electromechanical Engineering Department, University of Burgos, Burgos, Spain
jesierra@ubu.es

[3] Institute of Knowledge Technology, Complutense University of Madrid, Madrid, Spain
msantos@ucm.es

Abstract. The natural operation of wind turbines (WT) shows a nonlinear behavior which makes it difficult for the system to be controlled. Because of this, artificial intelligence techniques appear as promising control solutions. In this work, artificial neural networks (ANN) are used to complement the Direct Speed Control (DSC) of a wind turbine. Specifically, a neural network is used for the Maximum Power Point Tracking (MPPT) of a wind turbine model, controlling the generator speed and maintaining the active power into the correct levels to reach a power coefficient (C_p) within its optimum values. The real characteristics of a 1.5 MW wind turbine are considered. OpenFast and Matlab/Simulink software tools are used to model and simulate the non-linear WT and the controller, respectively. The intelligent proposed solution is compared with the standard control embedded in OpenFast with satisfactory results.

Keywords: Neural networks · Direct control · MPPT · Power coefficient · Wind turbine

1 Introduction

Nowadays, wind turbines (WT) are a research field that involves an important natural renewable resource, the wind. The energy of wind is extracted and transformed into electrical energy. The many advantages of using wind turbines have made their deployment and use grow exponentially in the last decades. Indeed, wind turbines have reached the fastest growing as energy generators around the world [1].

The wind turbine control includes different approaches according to the wind energy converter elements. Some of the control objectives are: reaching the nominal output power, maximization of energy that is generated during its operation, minimize losses, reduce vibrations, etc. [2, 3].

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 495–504, 2023.
https://doi.org/10.1007/978-3-031-18050-7_48

It is also important to highlight the issues that may directly affect the control efficiency such as those derived from the nonlinear dynamics of the WT, changes in the operation conditions, random wind speed variations, disturbances and noise in the involved signals, vibration of the structure, external loads, etc. All these reasons compel the necessity to implement more efficient and flexible control solutions, some of them based on soft computing methods.

Particularly, the goal of maximum power extraction is strongly related to the power coefficient, C_p. To address it, the maximum power point tracking (MPPT) principle is followed. This scheme is adopted during the operation stage named the MPPT region, as can be seen in Fig. 1 (left), between the lower wind speed at which WT starts working with a minimum angular speed, and the nominal wind speed where the turbine reaches the rated power. In this region, blades are fixed at the lowest angle to capture as much wind as possible. Hence, the turbine speed control aims at tracking the C_p curve to guarantee the maximum power extraction. In this operating region and under the presence of complex phenomena previously mentioned, it is clear the necessity of the design of smart control strategies. Artificial intelligent techniques are powerful alternatives to handle these issues.

Different control methods for MPPT have been proposed, some of them based on artificial intelligence techniques. For instance, in [4] a Fuzzy Logic Controller (FLC) is implemented for the direct speed control (DSC) of a WT, and in [5] the same control strategy is used for MPPT in presence of disturbances. In [6], a combination of neural networks (NN) and fuzzy logic is used to estimate the wind turbine maximum voltage and the control of the DC-DC boost converter. In those articles, other classical techniques are also mentioned and compared with, such as Hill Climb Search (HCS), Power Signal Feedback (PSF) or Perturb and Observer (P&O). Other investigations directly apply NN as control strategy in wind turbines, mainly for pitch control [7–10]. To summarize, NNs have proved to be a suitable method to develop control solutions for non-linear and complex systems in different fields [11–13].

Inspired by this line of research, in this paper neural networks are applied as part of the DSC of a WT to obtain, based on a reference generator speed, the more adequate electromagnetic torque (T_{em}) and thus, the reference for the output power (P_{ref}) to track the optimum C_p. OpenFast and Matlab/Simulink software tools have been used to model and simulate the wind turbine and the neural control strategy, respectively. This approach is compared with the standard variable-speed generator constant-torque control embedded in OpenFast for a standard 1.5 MW WT, providing better results.

The article is structured as follows. In Sect. 2 the turbine and generator models are described, including the dynamics. Section 3 presents the control based on neural networks. Simulations results of the intelligent control are discussed in Sect. 4. In Sect. 5 the conclusions of this study and future works are presented.

2 Mathematical Model of the Wind Turbine

The mechanical power extracted by the wind turbine is given by (1) [4].

$$P = \frac{1}{2}\rho C_p(\lambda, \beta) A {V_w}^3 \tag{1}$$

where the area swept by the blades is A [m^2], the wind speed is V_w [m/s], the air density is ρ [kg/m^3], λ is the tip speed ratio (TSR), β is the blades angle [rad], also called pitch angle, and C_p is the power coefficient.

TSR is given by Eq. (2), and it depends on the rotor radius, R [m], the turbine angular speed, ω_r [rad/s], and wind speed.

$$\lambda = \frac{\omega_r R}{V_w} \tag{2}$$

Power coefficient C_p is a function of TSR and β. It describes a set of curves depending on these two parameters and six coefficients (c1 to c6), related through expression (3), as follows [14]. It is specific for each turbine.

$$\begin{gathered} C_p(\lambda, \beta) = c_1\left(\frac{c_2}{\lambda_i} - c_3\beta - c_4\right)e^{-\frac{c_5}{\lambda_i}} + c_6\lambda \\ \frac{1}{\lambda_i} = \frac{1}{\lambda + 0.08\beta} - \frac{0.035}{\beta^3 + 1} \end{gathered} \tag{3}$$

The representative C_p curve for maximum power extraction can be obtained using OpenFast software. The optimal curve for the WT model we are working with is shown in Fig. 1 (right).

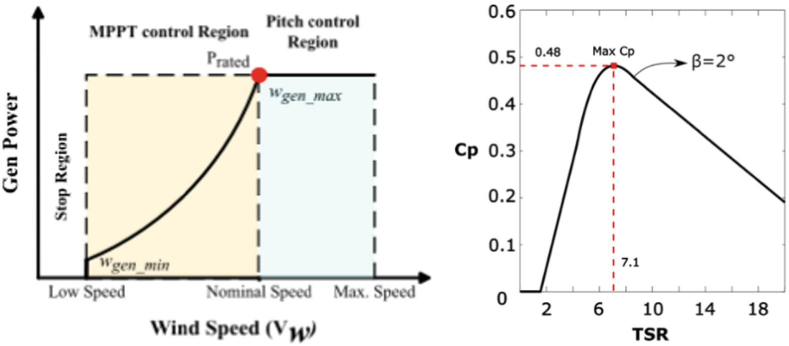


Fig. 1. WT Regions of Operation (left) and TSR vs C_p curve (right).

The maximum C_p coefficient is 0.48 for the simulated model, while the ideal blade angle is 2º.

2.1 DFIG Mathematical Description

For this study, a Doubly Fed Induction Generator (DFIG) wind turbine has been considered. The turbine has the blades, the gearbox with their two sides named low-speed shaft (LSS) and high-speed shaft (HSS), and the generator [4].

To model the generator, the DFIG type can be mathematically expressed in terms of the voltage and fluxes using Park transform in the d-q frame, represented by Eq. (4), [15], where V is the voltage [V], i denotes the current [A], φ is the flux [Wb], L is the inductance, and M refers to the mutual inductance [H], and R is the resistance [Ω], all of them respectively in the stator side (s) or rotor side (r). The angular speed of the reference frame rotating synchronously is ω_s [rad/s].

$$\begin{array}{ll} V_{ds} = R_s i_{ds} + \frac{d_{\varphi ds}}{dt} - \omega_s \varphi_{qs} & \varphi_{ds} = L_s i_{ds} + M \cdot i_{dr} \\ V_{qs} = R_s i_{qs} + \frac{d_{\varphi qs}}{dt} + \omega_s \varphi_{ds} & \varphi_{qs} = L_s i_{qs} + M \cdot i_{qr} \\ V_{dr} = R_r i_{dr} + \frac{d_{\varphi dr}}{dt} - \omega_r \varphi_{qr} & \varphi_{dr} = L_r i_{dr} + M \cdot i_{ds} \\ V_{qr} = R_r i_{qr} + \frac{d_{\varphi qr}}{dt} + \omega_r \varphi_{dr} & \varphi_{qr} = L_r i_{qr} + M \cdot i_{qs} \end{array} \tag{4}$$

The DFIG electromagnetic torque is given by (5), with p as the pair of poles:

$$T_{em} = \frac{3}{2} p \frac{M}{L_s} \left(\varphi_{qs} i_{dr} - \varphi_{ds} i_{qr} \right) \tag{5}$$

Regarding the mechanical operation, in the gearbox the speed and torque are converted from the LSS to the HSS. Equation (6) summarizes the relationship between the two sides of the mechanical model, detailed in [4, 16], where the variables are the gearbox ratio N, generator and rotor moments of inertia, J_g and J_r [$Kg \cdot m^2$], respectively, electromagnetic torque (T_{em}) [Nm], aerodynamic torque (T_r) [Nm] and rotor angular acceleration ($\dot{\omega}_r$) [rad/s^2].

$$T_r = \dot{\omega}_r \left(J_r + N^2 J_g \right) + N T_{em} \tag{6}$$

3 MPPT Control Based on DSC and Neural Networks

The direct speed control (DSC) can be expressed by (7), [17].

$$\omega_{gen\_ref} = N \sqrt{\frac{T_r}{K_{opt}}} \tag{7}$$

where the optimal constant K_{opt} is obtained through Eq. (8), while T_r is established after solving Eq. (6).

$$K_{opt} = \frac{1}{2} \pi \rho C_{p.opt} \frac{R^5}{{\lambda_{opt}}^3} \tag{8}$$

DSC control refers to the identification of the best T_{em} electromagnetic torque so to provide the optimal angular generator speed (ω_{gen_ref}) in terms of the rotor and generator dynamics, when wind speed varies.

Once the standard DSC is designed by (7), this speed control is complemented with an intelligent control strategy based on neural networks. The NN calculates the reference of the electromagnetic torque, T_{em_ref}, from the adjusted reference of the generator velocity, $\omega_{gen_{adj}}$, and its derivative, $\dot{\omega}_{gen_{adj}}$. The first is defined as the difference between ω_{gen_ref} and the generator rated speed. Then, the output power reference, P_{ref}, is obtained from (9). Now, the DFIG model has all the variables required to obtain the internal electrical parameters.

$$P_{ref} = T_{em\_ref} \cdot \omega_{gen} \tag{9}$$

A multi-layer perceptron (MLP) neural network with backpropagation is used to obtain the T_{em_ref}. The ANN configuration is set to 2 inputs, one hidden layer with 25 neurons, and 1 output (Fig. 2).

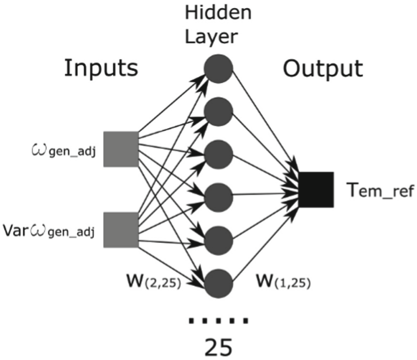


Fig. 2. Neural Network Model.

This supervised NN is trained with the data obtained from the application of the Indirect Speed Control (ISC) scheme [17, 18]. The ISC control gives an approximation of the reference electromagnetic torque based on the generator angular speed, ω_{gen}, and the optimal constant, K_{opt} (8). This control does not consider the influence of the rotor speed, ω_r, in the relation between the electromagnetic torque, T_{em}, and the rotor torque, T_r (6) [17]. For its implementation, the simplified Eq. (10) and the detailed values in Fig. 1 (right) are used.

$$T_{em\_ref\_ISC} = {\omega_r}^2 \cdot K_{opt} \tag{10}$$

While the ISC control method is applied, T_r, T_{em} and K_{opt} are monitored and saved. These T_{em} values are used as the output of the training dataset. In turn, the inputs of the training dataset are calculated by Eq. (7), and the values T_r and K_{opt} obtained with the ISC. Once the neural network is trained, the ANN gives real-time values for $T_{em_{ref}}$ in the proposed DSC-ANN architecture. This DSC-ANN combined control strategy of the WT is shown in Fig. 3.

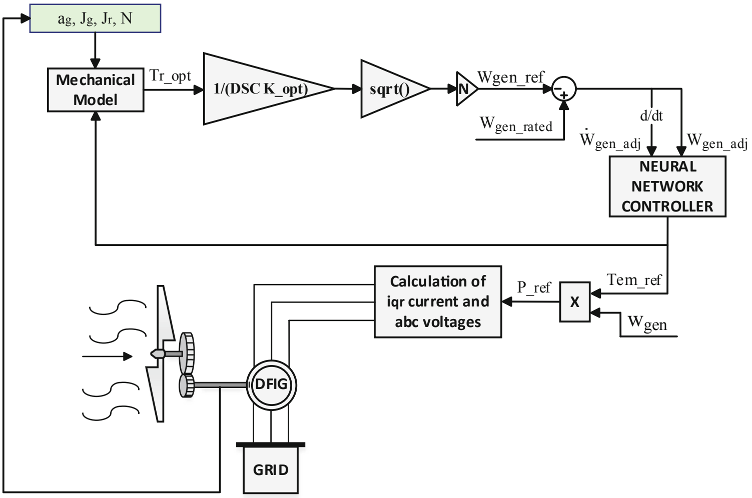


Fig. 3. DSC-ANN control architecture of a WT.

4 Discussion of the Results

The mechanical and electrical systems of the 1.5 MW WT are implemented. Simulations of the WT model following [4] are carried out using OpenFast and Matlab/Simulink. The blade radius of this WT is 42.75 m and the air density is 1.225 kg/m^3. The nominal torque is 12 kNm and the generator nominal speed is 1200 rpm. The rotor resistance is 4.496 mΩ, the stator resistance is 6.352 mΩ, the rotor inductance is 140.643 mH and the stator inductance is 154.253 mH. The inertia constant is 4.55 s.

The results of applying the WT control strategy based on DSC-NN are described below. Simulations have been carried out with a train of steps wind profile during 280 s (Fig. 4, left) and with a more realistic random wind signal during 350 s (Fig. 4, right). The latter has an average wind speed between 9 m/s and 11.5 m/s. Within this wind speed range, the WT operates in the MPPT control region, with blade angle fixed to their optimal value. Results are compared with the standard OpenFast variable-speed generator torque control. This requires specifying the generator rated speed and rated torque to establish the generator speed-torque relation that is followed in this region of operation.

For the step wind speed profile, Fig. 5 (left) shows the response of the generator angular speed for both, the OpenFast control and the proposed control strategy. It can be seen how it gets the rated speed, and the NN provides a smoother response and keeps the speed close to the maximum value for a longer time. Similarly, Fig. 5 (right) shows the electromagnetic torque. Using the NN based control, the values follow the reference signal accurately but with faster changes in contrast to the OpenFast control which maintains a slower response for each step variation.

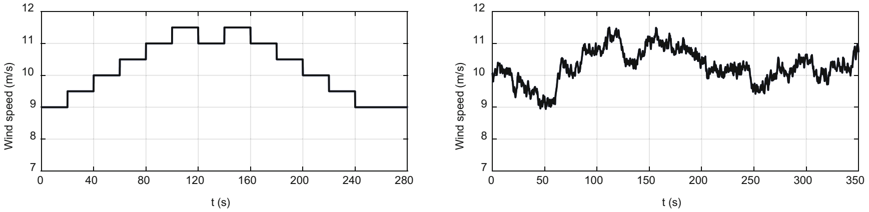


Fig. 4. Steps wind speed (left) and random wind speed (right) [m/s].

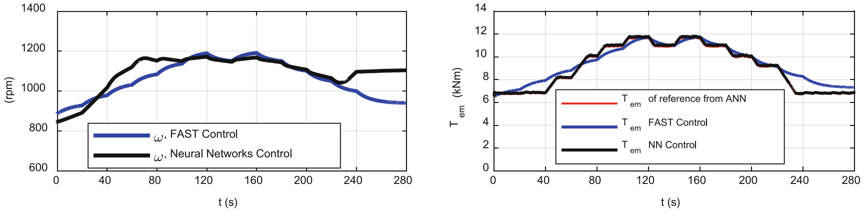


Fig. 5. Generator angular speed [rpm] (left) and electromagnetic torque (Tem) [kNm] (right), for step-shaped wind

Accordingly, the output power extracted (Fig. 6, left) is higher with the neural control. It is worth it to note that the power reaches a value nearer to the rated 1.5 MW and faster with the DSC-NN control. In the same way, Fig. 6 (right) shows the power coefficient. On the whole, the values reached with the NN control (black line) are very close to the optimal value of 0.48 (red line).

It can be concluded that the proposed DSC-ANN control is efficient for the torque control of the wind turbine with this wind profile.

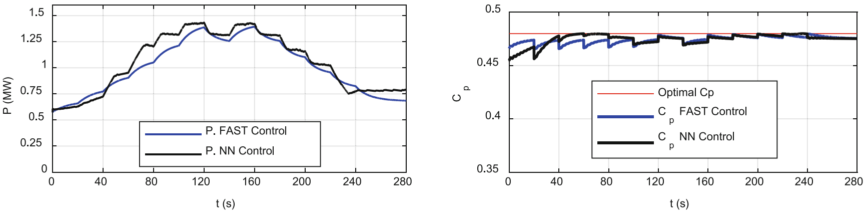


Fig. 6. Output Power (P) [MW] (left) and Power Coefficient (C_p) right), for step-shaped wind

Now, the random wind profile shown in Fig. 4 (right) is applied. Results are the following. Figure 7 (left) shows the generator angular speed. The maximum speed reaches a value slightly above the rated 1200 rpm for the fastest wind speed, and it is also higher than the one obtained with OpenFast. This behavior is possible due to the DFIG generator model feature that allows, during optimal operation, a speed variation of approximately 20% above or below the rated speed. The response clearly follows the WT dynamics influenced by the wind input.

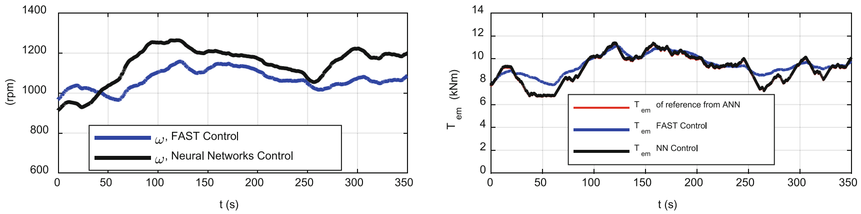


Fig. 7. Generator angular speed [rpm] (left) and electromagnetic torque (Tem) [kNm] (right), for random wind

The same behavior is reflected in the T_{em} signal (Fig. 7, right). The designed DSC-ANN control gives an adequate estimation of the reference torque from the speed error. The changes shown by the T_{em} are proportional to the input speed error. The final torque follows the reference with great precision, adapting the wind turbine operation quickly to the controlled variable.

In the case of the output power (Fig. 8, left), it also presents a quick response, influenced by the electromagnetic torque, thus following the increments and decrements of the latter, which makes it fluctuate around the nominal power of 1.5 MW for the highest wind speed. Even more, the neural network based control produces larger power during wind turbine operation.

The power coefficient (Fig. 8, right) is also very close to the optimum value, according to the TSR-C_p curve (Fig. 1, right). Indeed, the variations of this parameter with respect to the optimal value are very small. Moreover, the OpenFast control presents bigger error with respect to the reference C_p.

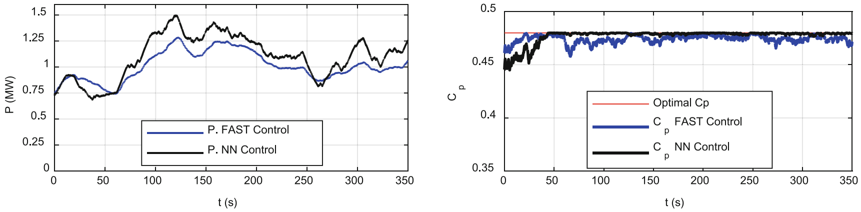


Fig. 8. Output Power (P) [MW] (left) and Power Coefficient (C_p) (right), for random wind

Finally, the estimated TSR response is shown in Fig. 9. The small changes are due to the WT dynamics variation because of the wind speed. It reaches the optimal value of 7.1 for which the control algorithm was designed, without any significant decrement, especially after big changes in wind speed, unlike the OpenFast control that gives smaller values.

To summarize, these results validate the proposed intelligent MPPT control technique based on NN for the wind turbine.

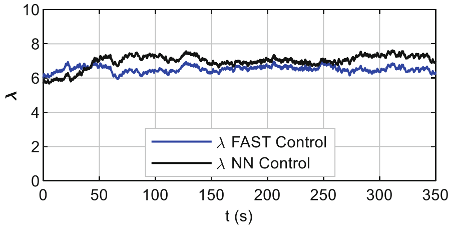


Fig. 9. Tip speed ratio

5 Conclusions and Future Works

Artificial intelligence techniques, such as Neural Networks, have been proved to be a successful alternative to solve control problems of complex dynamic systems. In this paper, a control strategy that combines direct speed control and neural networks is designed and applied to the generator control of a wind turbine. Results have shown the efficiency of this proposal, that keeps the power coefficient close to its maximum value and thus, optimizes the output power. It has been compared with the standard generator control implemented with the software OpenFast that is used to simulate the wind turbine real model, obtaining a better performance with the neural control.

So, neural networks demonstrate to be a robust, effective, and useful strategy to be applied in wind turbine MPPT control and, moreover, the synergy of different control approaches, such as in this case DSC and NN, allows it to get the most of the wind turbine.

Futures goals would be the consideration of disturbances and also completing the study with the pitch control, using this type of intelligent control.

Acknowledgments. This work has been partially supported by the Spanish Ministry of Science and Innovation under the project MCI/AEI/FEDER number RTI2018–094902-B-C21.

References

1. Chaoui, H., Miah, S., Oukaour, A., Gualous, H.: Maximum power point tracking of wind turbines with neural networks and genetic algorithms. In: IECON 2014–40th Annual Conference of the IEEE Industrial Electronics Society, pp. 197–201. IEEE, October 2014
2. Sierra-García, J.E., Santos, M.: Redes neuronales y aprendizaje por refuerzo en el control de turbinas eólicas. Revista Iberoamericana de Automática e Informática industrial **18**(4), 327–335 (2021)
3. Sierra-Garcia, J.E., Santos, M.: Deep learning and fuzzy logic to implement a hybrid wind turbine pitch control. Neural Comput. Appl. 1–15 (2021). https://doi.org/10.1007/s00521-021-06323-w
4. Muñoz, E., Ayala, E., Pozo, N., Simani, S.: Fuzzy PID control system analysis for a wind turbine maximum power point tracking using FAST and Matlab simulink. In: Brazilian Technology Symposium, pp. 905–917. Springer, Cham, October 2020
5. Muñoz, E., Ayala, E., Pozo, N.: Estrategia de Control Fuzzy PI en una Turbina Eólica con Generador de Inducción Doblemente Alimentado para Maximizar la Extracción de Potencia en Presencia de Perturbaciones. Revista Técnica" energía", **18**(1), 1–10 (2021)

6. El Aissaoui, H., El Ougli, A., Tidhaf, B.: Neural networks and fuzzy logic based maximum power point tracking control for wind energy conversion system. Adv. Sci. Technol. Eng. Syst. J. **6**(2), 586–592 (2021)
7. Rhaili, S.E., Abbou, A., El Hichami, N., Marhaoui, S.: A new strategy based neural networks MPPT controller for five-phase PMSG based variable-speed wind turbine. In: 2018 7th International Conference on Renewable Energy Research and Applications (ICRERA), pp. 1038–1043. IEEE, October 2018
8. Mesemanolis, A., Mademlis, C.: A neural network based MPPT controller for variable speed wind energy conversion systems. In: 8th Med. Conf. Power Generation, Transmission, Distribution and Energy Conversion (MEDPOWER 2012), pp. 1–6 (2012)
9. Thongam, J.S., Bouchard, P., Beguenane, R., Fofana, I.: Neural network based wind speed sensorless MPPT controller for variable speed wind energy conversion systems. In: 2010 IEEE Electrical Power & Energy Conference, pp. 1–6. IEEE, August 2010
10. Sierra-García, J.E., Santos, M.: Performance analysis of a wind turbine pitch neurocontroller with unsupervised learning. Complexity (2020)
11. Sierra-García, J.E., Santos, M.: Switched learning adaptive neuro-control strategy. Neurocomputing **452**, 450–464 (2021)
12. Sierra, J.E., Santos, M.: Wind and payload disturbance rejection control based on adaptive neural estimators: application on quadrotors. Complexity (2019)
13. Santos, M., Dexter, A.L.: Temperature control in liquid helium cryostat using self-learning neurofuzzy controller. IEE Proc. Control Theory Appl. **148**(3), 233–238 (2001)
14. Yang, B., Zhang, X., Yu, T., Shu, H., Fang, Z.: Grouped grey wolf optimizer for maximum power point tracking of doubly-fed induction generator based wind turbine. Energy Convers. Manage. **133**, 427–443 (2017)
15. Sahri, Y., Tamalouzt, S., Belaid, S.L.: Direct torque control of DFIG driven by wind turbine system connected to the grid. In: 2018 International Conference on Wind Energy and Applications in Algeria (ICWEAA), pp. 1–6. IEEE, November 2018
16. Semrau, G., Rimkus, S., Das, T.: Nonlinear systems analysis and control of variable speed wind turbines for multiregime operation. J. Dyn. Syst. Measure. Control **137**(4) (2015)
17. Abad, G., Lopez, J., Rodriguez, M., Marroyo, L., Iwanski, G.: Doubly Fed Induction Machine: Modeling and Control for Wind Energy Generation. John Wiley & Sons (2011)
18. Senatore, P.: Modelado y control de la máquina sincrónica de reluctancia. Aplicación en un aerogenerador. Tesis de Máster. Universidad de la República, Montevideo (2013)

A Control Approach on Hybrid Floating Offshore Wind Turbines for Platform and Generated Power Oscillations Reduction at Below-rated Wind Speed

Payam Aboutalebi(✉), Fares M'zoughi, Irfan Ahmad, Aitor J. Garrido, and Izaskun Garrido

University of the Basque Country (UPV/EHU), Leioa, Spain
{payam.aboutalebi,fares.mzoughi,irfanahmad.irfan,aitor.garrido, izaskun.garrido}@ehu.eus

Abstract. This paper presents the use of four oscillating water columns integrated into the platform of a barge-type floating offshore wind turbine. A control strategy has been proposed to decrease the system's oscillations and generated power fluctuations by adequately controlling the opening of the airflow valves. The switching time for below-rated wind speed has been calculated using the platform's pitch response amplitude operator. The blades' pitch has been adjusted to harness the maximum energy at below-rated wind speed and a constant torque method has been employed for the generator. A comparative study has been carried out between uncontrolled traditional barge-type and controlled oscillating water columns-based barge floating offshore wind turbine to determine the performance of the control technique. The findings demonstrate that the suggested control approach can effectively decrease both the oscillations in the system's states and the fluctuations in the generated power. The results show a 26.6% fluctuation reduction in the generated power for the controlled OWCs-based barge platform, compared with the standard barge platform.

1 Introduction

Although the use of conventional fossil fuels such as oil, gas, and coal may promote economic growth, excessive use of nonrenewable sources emits a large amount of carbon dioxide into the atmosphere, likely to result in global warming. Policy initiatives have resulted in the development of renewable energies in recent years, giving rise in the harness of wind and wave energies. As a result, the configuration of floating offshore wind turbines (FOWT) has been growing substantially throughout the world in order to achieve a low-carbon society and improve the utilization of sea spaces [6].

The oscillations caused by waves and winds on the FOWT states are a challenge in the offshore wind field. These oscillations may cause mechanical problems by putting undesired loads on the blades, rotor shaft, yaw bearing, and

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 505–514, 2023.
https://doi.org/10.1007/978-3-031-18050-7_49

tower, decreasing aerodynamic performance, tower fatigue life and generated power [4]. Oscillating Water Columns (OWC), one of the most studied Wave Energy Converters (WEC), could be incorporated inside the floating platform to decrease the vibrations in the system cite.

Researchers have proposed hybrid FOWT-WECs to harvest energy from both wave and wind. For example, a structure of three rotating-flaps WECs attached on a semisubmersible FOWT [7], a hybrid tension leg platform-type FOWT and three point-absorber WECs [3], and a hybrid model of a spar-type FOWT and a torus-shaped point-absorber [8]. Several documents have described how to modify the structure of FOWTs to prevent vibrations in the system. M. Palraj et al. introduced in [10] the use of a gyro-stabilizer installed in the barge of a FOWT to control the vibrations. J. Yang et al. in [12] installed a tuned mass damper inside the barge platform to decrease the vibrations.

On the other hand, J. Jonkman constructed a square-shaped moonpool placed in the center of the barge, designed to allow the integration of an OWC inside the wind turbine's tower [5]. P. Aboutalebi et al. [1] evaluated the dynamics of hybrid barge-type FOWT-OWCs in different sea states. F. M'zoughi et al. in [9] designed a linear model of combined two OWCs-based FOWTs, controlled by a PID controller for the FOWT stabilisation. P. Aboutalebi et al. in [2] developed a switching control strategy based on Response Amplitude Operators (RAO) to improve the overall performance of system in the absence of wind. This paper describes a control strategy based on RAOs that takes into account different wave periods as well as the wind at below-rated wind speed. In addition to designing a controller for OWCs, a second and third controllers for the blades' pitch and generator have been developed to maximize the energy harness at below-rated wind speed.

This paper is organized as follows: Sect. 2 describes the equations of motion for the system. Section 3 explains the problems and challenges due to the platform's oscillations. In Sect. 4, the control strategies have been examined under different sea states and below-rated wind speed. Finally, Sect. 5 presents the conclusions of this paper.

2 FOWT Model

In this paper, the full nonlinear equations of motion for a 5-MW offshore wind turbine mounted on a barge have been studied. Four OWCs have been integrated inside the barge platform, as shown in Fig. 1, in order to reduce the oscillations in the system's states and generated power.

The full nonlinear equations of motion in time domain for the coupled floating wind turbine, support platform system and OWCs may be described as follows:

$$M_{ij}(q,u,t)\ddot{x}_j = f_i(q,\dot{q},u,t) \tag{1}$$

where M_{ij} (kg) expresses the inertia mass elements and x defines the states of the system. u describes the control inputs. The external forces of aerodynamic loads on the blades and nacelle, hydrodynamic forces on the platform, elastic

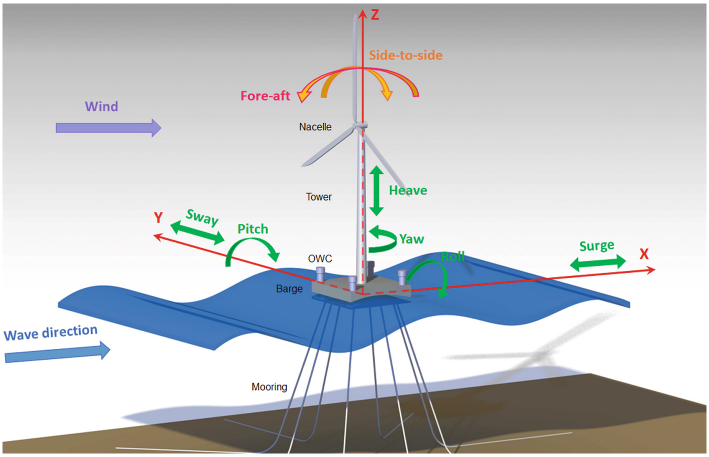


Fig. 1. Barge-type FOWT equipped with four OWCs.

and servo forces and Power Take Off (PTO) have been described as f_i on the right hand-side of Eq. 1.

The equation of motion in frequency-domain may be described as [1]:

$$I_{FOWT}(\omega)\ddot{q} + B_{FOWT}(\omega)\dot{q} + K_{FOWT}q = \mathbf{f}_{FOWT}(\omega) + \mathbf{f}_{PTO}(\omega) \qquad (2)$$

where I_{FOWT}, B_{FOWT} and K_{FOWT} stand for the inertia elements, damping components and stiffness matrix, respectively. $\mathbf{f}_{FOWT}(\omega)$ expresses the hydrodynamic force, viscous drag and aerodynamic loads. $\mathbf{f}_{PTO}(\omega)$ denotes the load caused by the PTO equipment and ω denotes the wave frequency. The term q in Eq. 2 is described as [11]:

$$q = \begin{bmatrix} surge & sway & heave & roll & pitch & yaw & fore\text{-}aft & side\text{-}to\text{-}side \end{bmatrix} \qquad (3)$$

The inertia elements of FOWT is described by:

$$I_{FOWT}(\omega) = A_{Hydro}(\omega) + M_{Platform} + M_{Tower} \qquad (4)$$

where $M_{Platform}$ (kg) is platform mass and M_{Tower} (kg) is tower mass and A_{Hydro} expresses the platform's added mass, may be calculated by the panel radiation program namely WAMIT.

The stiffness matrix K_{FOWT} may be defined as follows:

$$K_{FOWT} = K_{Hydro} + K_{Mooring} + K_{Tower} \qquad (5)$$

where K_{Hydro}, $K_{Mooring}$ and C_{Tower} describe the platform' hydrostatic restoring matrix, the mooring lines spring stiffness elements and the tower stiffness coefficients, respectively.

The damping coefficients can be described as:

$$B_{FOWT}(\omega) = B_{Hydro}(\omega) + B_{Tower} + B_{viscous} + B_{chamber} \tag{6}$$

where B_{Hydro} is platform's damping elements, B_{Tower} is damping matrix of the flexible tower and $B_{viscous}$ is the viscous drag. $B_{chamber}$ describes the the PTO's effect.

Thus, the equation of motion in frequency domain for the FOWT is obtained as follows, as described in Eq. 2:

$$I_{FOWT}(\omega)\,\ddot{\hat{q}} + (B_{FOWT}(\omega) + B_{PTO}(\omega))\,\dot{\hat{q}} + (K_{FOWT} + K_{PTO})\hat{q} = \mathbf{f}_{FOWT}(\omega) \tag{7}$$

In order to define the geometry of the platform, MultiSurf has been used. Figure 2a, 2b and 2c demonstrate three platforms including standard barge platform, closed OWCs-based barge platform and open OWCs-based barge platform, respectively.

3 Problem Statement

The FOWTs' Oscillations are undesirable because they have a negative impact on the system, consisting of stress on the system's structural components, a significant reduction in wave and wind energy harvest, and fluctuations in the generated power. Increased maintenance costs could also decrease the system efficiency. To tackle the problem, a switching control strategy has been proposed based on the RAOs. In this section, first the RAOs are evaluated, then the control strategy is described.

3.1 RAOs Evaluation

It is essential to use RAOs in order to assess the movement of FOWTs in different sea conditions. The procedure for plotting the RAOs for different states of the input-output system is explained in this section. Using MultiSurf, WAMIT, FAST and MATLAB and the following equation the RAOs can be achieved:

$$RAO = \frac{S_{xy}(\omega)}{S_{xx}(\omega)} \tag{8}$$

where $S_{xy}(\omega)$ and $S_{xx}(\omega)$ are the cross-spectral and auto-spectral densities of the wave elevation input $x(t)$ and the system's stats output $y(t)$, respectively.

The system states' RAOs with respect to the wave period are presented at below-rated wind speed of 8 m/s, showed in Fig. 3. Platform pitch and fore-aft states are significant because these states are the most altered modes by the wave and wind that are aligned with the surge mode. Oscillations occur in the platform pitch, showed in Fig. 3f for below-rated wind speed.

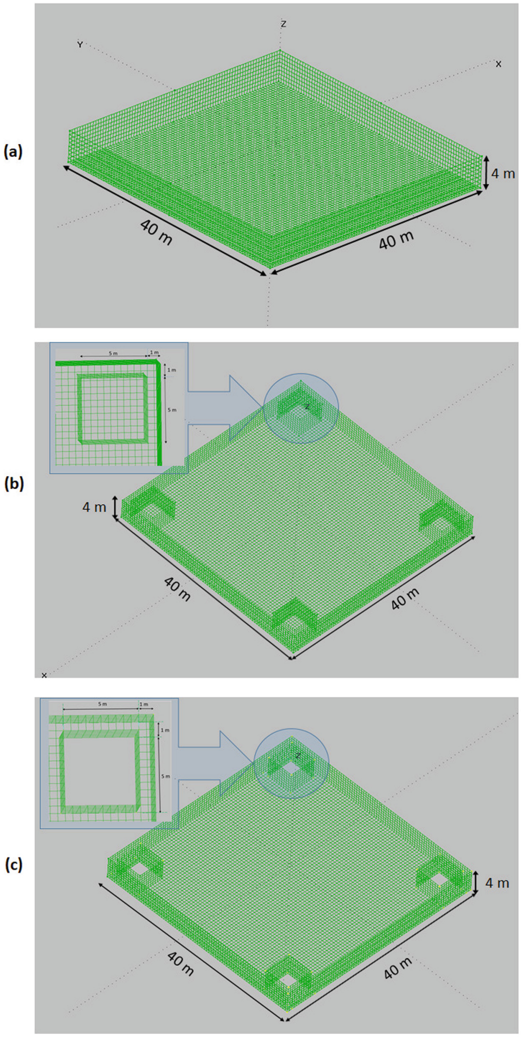


Fig. 2. Platforms' design for (**a**) Standard barge platform. (**b**) closed OWCs-based barge platform. (**c**) open OWCs-based barge platform.

3.2 Control Statement

A control method is introduced to reduce the oscillations in the FOWT's states as represented in Fig. 4. The proposed control method is based on the analysis of the platform pitch RAOs, illustrated in Fig. 3f at below-rated wind speed. As it is seen in the figure, the platform pitch RAO for the open OWCs-based barge platform crosses the platform pitch RAO for the closed OWCs-based platform at a wave period called switching point. The switching point is the wave periods 12.37 s for the wind speed of 8 m/s. Therefore, the switching controller opens the

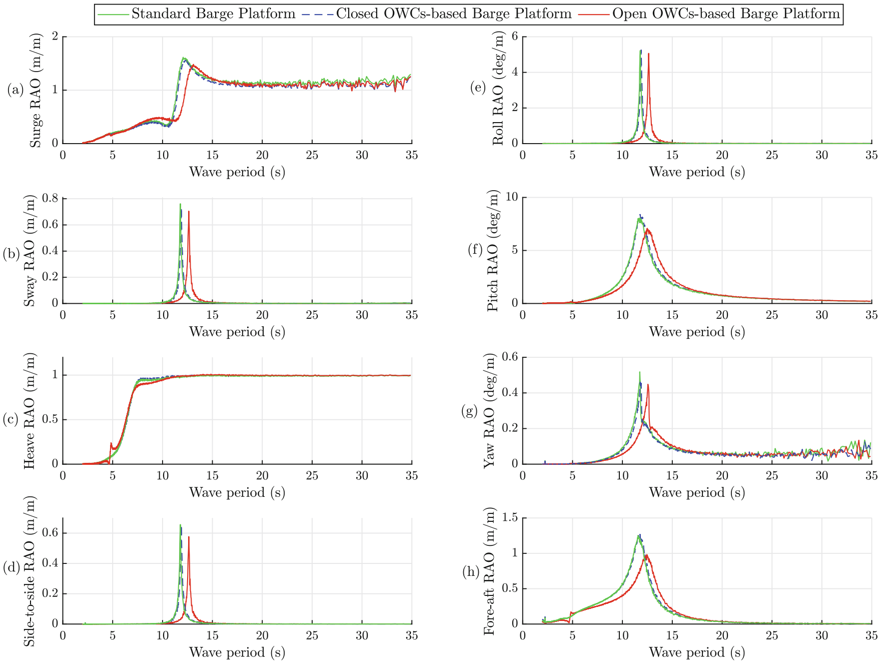


Fig. 3. RAOs at the below-rated wind speed of 8 m/s for (a) Surge. (b) Sway. (c) Heave. (d) Side-to-side. (e) Roll. (f) Pitch. (g) Yaw. (h) Fore-aft.

valves for waves with the periods lower than the switching point and closes the valves for waves with higher periods than the switching point. The equation for the switching technique is defined as follows:

$$u(T_w) = \frac{e^{(T_w - T_{sp})}}{e^{(T_w - T_{sp})} + 1} \tag{9}$$

where $u(T_w)$ represents a sigmoid function of the closing and opening valves' control input. T_w and T_{sp} express the wave period measured by a sensor and the switching point from platform pitch RAO, respectively.

In this article, blade-pitch angle is adjusted at zero to extracted the most energy from wind at below-rated wind speed. Also, variable-speed operation mode based on the generator torque manipulation have been employed in order to evaluate the performance of the switching control technique on the generator power output. The operation regions are divided into five modes including the regions 1, 1 $^1\!/_2$, 2, 2 $^1\!/_2$ and 3 (see Fig. 5). In region 2 (below-rated wind speed), the generator torque is proportional to the the filtered generator speed in order to keep an optimal tip-speed ratio. For more details of the variable-speed control of the generator, refer to [5].

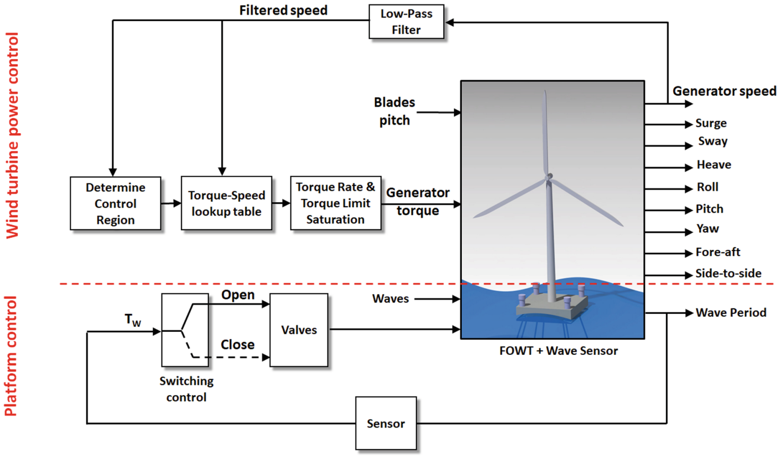


Fig. 4. Control scheme for the OWCs-based platform with wind turbine power control at below-rated wind speed.

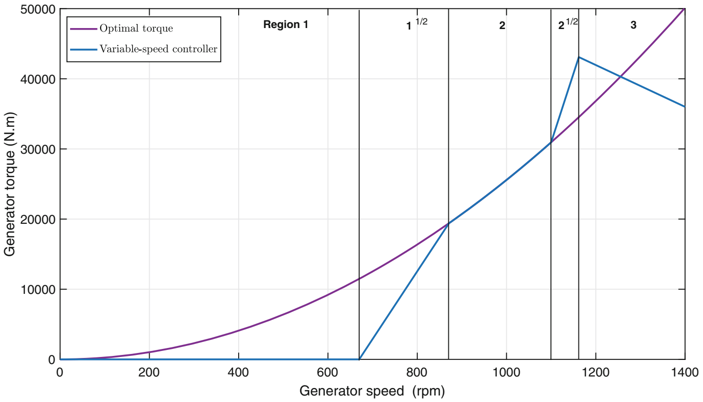


Fig. 5. Generator toque versus generator speed.

4 Results

To show the performance of the controlled OWCs-based barge platform compared to the uncontrolled standard barge platform, a time-domain simulation has been performed. The regular wave with an amplitude of 0.9 m has been considered, showed in Fig. 6a.

Based on the pitch RAO at the wind speed of 8 m/s, the switching controller acts to open the valves for the periods less than 12.37 s and to close the valves for the periods higher than 12.37 s. Here the OWCs' valves transition from closing to opening at 600 s.

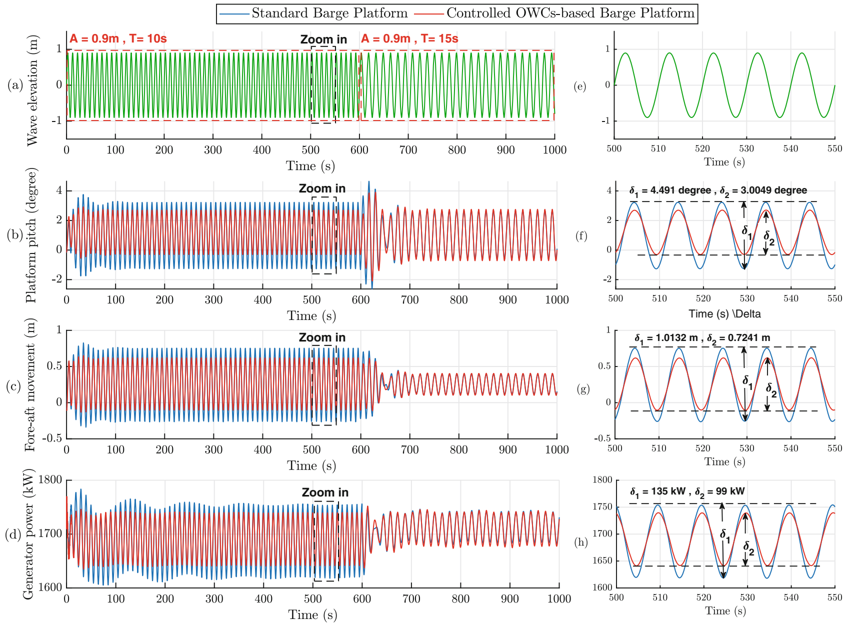


Fig. 6. First scenario at below-rated wind speed. (a) wave elevation. (b) platform pitch. (c) fore-aft movement. (d) generator power.

The standard barge platform and controlled OWCs-based barge platform are represented in blue and red respectively. As it can seen in Fig. 6b, the controlled OWCs-based barge platform pitches by 3.0049 °C while the standard barge platform pitches by 4.491 °C. This shows that the pitch oscillations in the controlled OWCs-based platform are drastically lower than the standard barge platform by 33% after transient time. Both controlled OWCs-based and uncontrolled standard barge platforms' pitch curves are almost identical with very slight difference after 600 s when the OWCs' controller manages to close the valves. Fore-aft motion, showed in Fig. 6c, shows the same behaviour as the platform pitch angles for both controlled OWCs-based and standard barge platforms.

Figure 6d indicates the generator power controlled by variable speed method. It is observed from the figure that there is a relationship between the platform pitch and fore-aft oscillations with generator power fluctuations. Before 600 s, the generator power in controlled OWCs-based barge platform fluctuates for 99 kW whereas the generator power in the standard barge platform fluctuates for 135 kW. This shows a 26.6% fluctuation reduction in the controlled OWCs-based barge platform, compared with the standard barge platform.

5 Conclusions

A control method was introduced to increase the stability of the FOWT by deduction of the oscillations in the platform and tower modes. The system was evaluated in various sea states and below-rated wind speed. The transition between opening and closing valves has been conducted by analysing the platform pitch RAOs. The platform pitch RAO provides the data for analysing the behaviour of the FOWT. To capture the wind energy, the blades' pitch were adjusted to zero degrees at below-rated wind speed and a variable speed controller was designed for the generator.

The results showed that the proposed switching control technique in OWCs' valves was able to decrease the oscillations in the system effectively. The performance of the controlled OWCs-based barge platform in terms of oscillations reduction was better, overall in different environmental conditions, compared to uncontrolled standard barge platform. Moreover, the generator power fluctuations decreased efficiently.

Acknowledgments. This work was supported in part by the MCIU/MINECO through the projects RTI2018-094902-B-C21 and RTI2018-094902-B-C22 (MCIU/AEI/FEDER, UE) and by the Basque Government through the project IT1555-22.

References

1. Aboutalebi, P., M'zoughi, F., Garrido, I., Garrido, A.J.: Performance analysis on the use of oscillating water column in barge-based floating offshore wind turbines. Mathematics **9**(5), 475 (2021)
2. Aboutalebi, P., M'zoughi, F., Martija, I., Garrido, I., Garrido, A.J.: Switching control strategy for oscillating water columns based on response amplitude operators for floating offshore wind turbines stabilization. Appl. Sci. **11**(11), 5249 (2021)
3. Bachynski, E.E., Moan, T.: Point absorber design for a combined wind and wave energy converter on a tension-leg support structure. In: International Conference on Offshore Mechanics and Arctic Engineering, vol. 55423, p. V008T09A025. American Society of Mechanical Engineers (2013)
4. Haji, M.N., Kluger, J.M., Sapsis, T.P., Slocum, A.H.: A symbiotic approach to the design of offshore wind turbines with other energy harvesting systems. Ocean Eng. **169**, 673–681 (2018)
5. Jonkman, J.M.: Dynamics modeling and loads analysis of an offshore floating wind turbine. University of Colorado at Boulder (2007)
6. Kosasih, K.M.A., Suzuki, H., Niizato, H., Okubo, S.: Demonstration experiment and numerical simulation analysis of full-scale barge-type floating offshore wind turbine. J. Marine Sci. Eng. **8**(11), 880 (2020)
7. Michailides, C., Gao, Z., Moan, T.: Experimental and numerical study of the response of the offshore combined wind/wave energy concept SFC in extreme environmental conditions. Mar. Struct. **50**, 35–54 (2016)
8. Muliawan, M.J., Karimirad, M., Moan, T., Gao, Z.: STC (spar-torus combination): a combined spar-type floating wind turbine and large point absorber floating wave energy converter-promising and challenging. In: International Conference on Offshore Mechanics and Arctic Engineering, vol. 44946, pp. 667–676. American Society of Mechanical Engineers (2012)

9. M'zoughi, F., Aboutalebi, P., Garrido, I., Garrido, A.J., De La Sen, M.: Complementary airflow control of oscillating water columns for floating offshore wind turbine stabilization. Mathematics **9**(12), 1364 (2021)
10. Palraj, M., Rajamanickam, P.: Motion control of a barge for offshore wind turbine (OWT) using gyrostabilizer. Ocean Eng. **209**, 107500 (2020)
11. Wamit, U.M.: V7. 2. Wamit. Inc., Massachusetts (2016)
12. Yang, J., He, E., Hu, Y.: Dynamic modeling and vibration suppression for an offshore wind turbine with a tuned mass damper in floating platform. Appl. Ocean Res. **83**, 21–29 (2019)

Control Tuning by Genetic Algorithm of a Low Scale Model Wind Turbine

Giordy Alexander Andrade Aimara[1](✉), Segundo Esteban San Román[2], and Matilde Santos[3]

[1] Faculty of Physics, UCM, 28840 Madrid, Spain
giordyan@ucm.es
[2] DACYA, Faculty of Physics, UCM, 28840 Madrid, Spain
sesteban@ucm.es
[3] Institute of Knowledge Technology, UCM, 28840 Madrid, Spain
msantos@ucm.es

Abstract. The continuous rise of wind energy makes it necessary to design controllers that make turbines more and more efficient. However, control designs are usually developed in simulation, which does not consider the many factors that affect control in a real turbine. An intermediate step, before testing them on turbines, is to check them on prototypes. In this paper, a first design of a laboratory scale model of a wind turbine (WT) is proposed, with the aim of testing different control algorithms. The model is built with commercial hardware and "ad hoc" circuits. Two control loops have been implemented; an external loop that commands the electric charge, and an internal loop that controls the pitch of the blades. The controllers have been tuned first experimentally, obtaining the best possible behavior by trial and error. Using genetic algorithms, the most optimal values for the controller are obtained, improving the response of the system. The operating and functional modes of a real WT have been also replicated on the model using a microcontroller programmed with the Arduino IDE input-output. Results obtained using optimized conventional controllers prove the correct performance of the prototype.

Keywords: Wind turbine · Control · Genetic algorithms · Scale model · Prototype · Arduino

1 Introduction

At present, wind energy is experiencing unprecedented development throughout the world. The recent appearance of floating wind turbines (WT) has increased expectations in energy production but has also created new challenges for control [1].

Control algorithms are becoming more sophisticated, to address these more complex problems [2]. These more advanced and intelligent control designs are usually developed in simulation, which does not consider the many factors that affect control in a real wind turbine (WT). An intermediate step, before testing them on turbines, is to check them

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 515–524, 2023.
https://doi.org/10.1007/978-3-031-18050-7_50

on prototypes. Besides, this approach makes it easier for the industry and the companies of the wind sector to incorporate the new solutions.

With this purpose, in this paper we describe a Low Scale Model (LSM) of a real wind turbine. This prototype, that includes significant improvements regarding a much simpler one previously proposed in [3], is equipped to test different control algorithms. The physical model of this first approach is shown in Fig. 1. It includes hardware in the loop and the dynamics of the system, scaled appropriately.

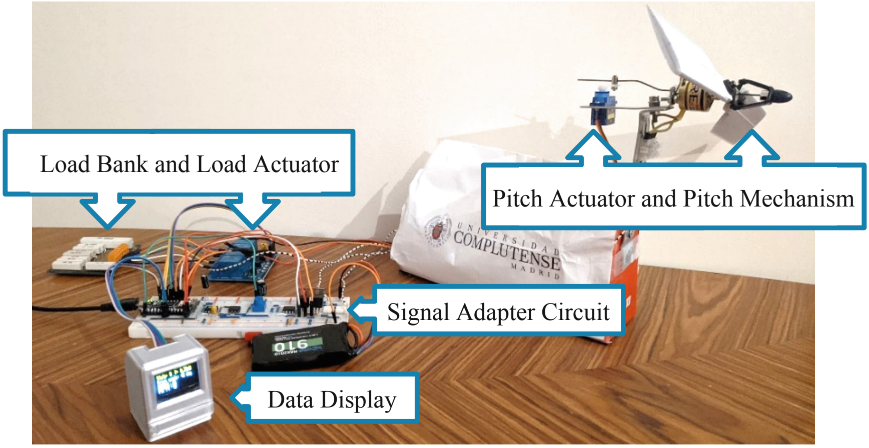

Load Bank and Load Actuator
Pitch Actuator and Pitch Mechanism
Signal Adapter Circuit
Data Display

Fig. 1. Scale model of a wind turbine.

This LSM has allowed us to observe some problems that appear when working with real systems that are not manifested in simulation. In addition, it has been proved that the tuning of the controller parameters by soft computing techniques, specifically genetic algorithms, provide solutions that work on the real prototype. The validation of different control solutions on the scale model may facilitate the implementation of any control solutions on medium-scale mill models or even on real wind energy converter devices [4].

The structure of the paper is as follows. Section 2 present the wind turbine model. In Sect. 3 the low scale model is described, explaining how each component has been built. Experimental results are analyzed in Sect. 4. Finally, conclusions and future works end the article.

2 Wind Turbine Modelling

2.1 Aerodynamic Model

The aerodynamic performance of a wind turbine is key to understand how energy is extracted [5]. The wind power P_{in} *[W]* at the input of the WT is given by:

$$P_{in} = 0.5\rho A U^3 \tag{1}$$

where ρ *[kg/m^2]* is the air density, A *[m^2]* is the area swept by the blades, and U *[m/s]* is the average wind speed. Taking into account the Betz's limit, the ratio of the kinetic power input, P_{in}, to the electrical power output, P_{out}, is the power coefficient Cp (2), which is a function of the blades pitch angle, θ *[°],* and the Tip Speed Ratio (TSR), λ.

$$C_p(\lambda, \theta) = P_{out}/P_{in} \tag{2}$$

The TSR is a dimensionless coefficient that relates the tangential speed of the tip of a blade and the wind speed.

$$\lambda = r\omega_t/U \tag{3}$$

where r *[m]* and ω_t *[rad/s]* are the radius and the angular velocity of the wind turbine, respectively. When the pitch angle increases, the power coefficient decreases, and consequently, the output power of the WT [6]. Even more, in the LSM it will be necessary to work witch large pitch angles to compensate the low wind speeds used. The effect of the power coefficient is replicated in the LSM with the pitch control action.

3 Wind Turbine Low Scale Model

The LSM is a small replica (scale ≈ 1/100) of an onshore wind turbine. Like in a real WT, it includes two control loops: an internal one that controls the pitch angle of the blades, and an external one to control the electrical load of the generator (see Fig. 2).

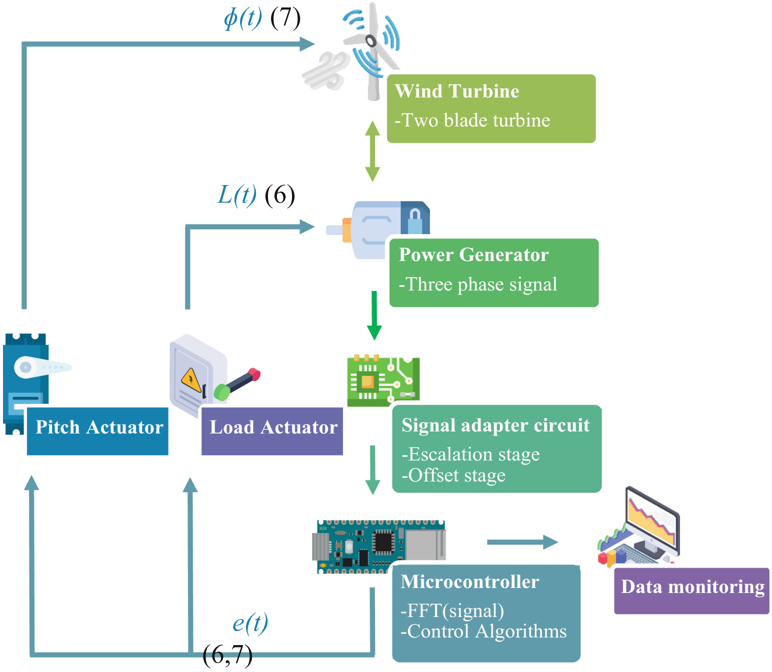


Fig. 2. LSM components and control loops.

The main components of the LSM are the following:

- Two built-in-the-house blades installed on a variable pitch mechanism.
- A radio-control (RC) brushless motor used as electric generator.
- A built-in-the-house signal adapter circuit.
- A microcontroller to monitor the system and to execute the control algorithms.
- The electric load actuator and the pitch actuator.

3.1 Blades and Pitch Mechanism

To emulate the performance of the wind turbine, specific blades have been designed and built with a 3D-printer. A symmetric aerodynamic profile, the NACA0012 (National Advisory Committee for Aeronautics) has been selected to capture more wind [7]. These blades have more inertia than RC commercial blades, adding more realism to the behavior of the prototype. As pitch actuator, an SG90 RC servomotor is used with a two-blade variable pitch mechanism. With this setup, an open loop set of experiments was be carried to characterize the C_p coefficient above an optimal pitch angle.

3.2 Electric Generator and Load Actuator

The electric generator is emulated by a RC brushless motor, which provides a three-phase signal at the output. The frequency of electric signal is directly proportional to the rotor angular velocity, and depends on the number of pairs of poles of the stator.

$$\omega_s = 60 f_g / N \tag{4}$$

Being ω_s the synchronism speed $[rpm]$, f_g *[Hz]* the frequency of the three-phase signal, and N, the number of pairs of poles.

This generator contains six pairs of windings facing each other on the stator, thus obtaining a synchronous speed ten times greater than the signal frequency. An actuator mechanism capable of increasing the power extracted for the same three-phase signal, and with it, the electro-magnetic torque, is designed. For this purpose, the load resistance of the generator terminals is decreased by means of a load bank. This device can supply electrical loads in order to test electrical supplies. Ceramic resistors are used to ensure proper power dissipation. Using a four-channel relay, switching between the built-in parallel resistors is performed, obtaining up to sixteen possible states.

3.3 Measurement System

The complete WT LSM includes a signal adapter circuit to safely process the signal provided by the generator. The microcontroller is in charge of reading the three-phase signal to obtain the revolutions of the turbine and to calculate the generated power.

Even more, the security of the microcontroller must be guaranteed. The signal adapter circuit is responsible for limiting the three-phase signal within the reading range of the microcontroller. To do so, firstly optocouplers that are able of electrically isolating the generator output from the measurement system are included. Next, operational amplifiers are included in an inverting amplifier configuration to regulate the amplitude of the signal based on the values of the built-in resistors. However, by working with an alternating signal centered on zero, it is necessary to add an offset level to the signal to center it in the reading range. This is achieved with an operational amplifier in non-inverting configuration, whose added continuous signal is obtained from the intermediate terminal of a potentiometer connected between power supply and ground. Finally, a high input impedance measure instrument has been designed with respect to the system to be measured.

3.4 Microcontroller

The Arduino Nano 33 IOT [8] has been selected for the LSM which, in addition to incorporating different wireless connections, is equipped with eight analog inputs and an IMU module that allows it to calculate the attitude of the system, a very interesting measurement for future Floating Offshore Wind Turbine (FOWT) implementation.

The adapted signal is read by the microcontroller using AD converters. Digital signals are real time processed using the Fast Fourier Transform (FFT) with a sample rate of 300 Hz and 1024 bits of the ADC. This way it is possible to calculate the frequency and amplitude of the signal and thus, the angular velocity and output power of the WT. Depending on the these values and the WT state, an operational logic is applied and the pertinent control algorithms is used.

Data obtained from the system are: attitude, angular velocity, blade pitch angle, load and output power. Data are sent through the port-serial connection to a selected IP address by WiFi-UDP protocol. For the representation of the mentioned data, the Matlab® Simulink software is used. To give the system greater autonomy, the information is also displayed on a small built-in LCD screen.

3.5 Load Control System

To control the rotational speed of the WT, it is necessary to control the electromagnetic torque τ_g *[Nm]* of the generator:

$$\tau_g = P_g / \left(\eta_g \eta_m \omega_g\right) \tag{5}$$

where P_g *[W]* is the power extracted by the generator, η_g the efficiency of the generator, η_m the efficiency of the mechanism, and ω_g *[rad/s]* the speed of rotation of the generator, equal to the turbine's in our case. This torque counteracts the torque generated by the wind on the blades, slowing down the turbine.

The microcontroller handles the switch between the different possible states until the electromagnetic torque is able to control the rotor revolutions depending on the state of the WT. In the corresponding regions, a PD control is applied to stabilize the output power. The proportional action reduces the output error (difference between rated speed and measured speed), while the differential action is responsible for compensating the inertia of the system.

$$L(t) = K_{PL} e(t) + K_{DL} T_D de(t)/dt \tag{6}$$

3.6 Pitch Control System

The pitch control is necessary to control the aerodynamic torque of the generator. It is implemented with a PID (7) regulator which, in addition to reducing the error in the state transitions, removes the error in the stationary state. The pitch control is accurate enough to stabilize the WT within a desired range.

$$\Phi(t) = K_{P\Phi}e(t) + K_{I\Phi}/T_I \int e(\tau)d\tau + K_{D\Phi}T_D de(t)/dt \tag{7}$$

3.7 Software Architecture

By a correct sequential coding it is possible to solve the absence of threads of execution of the single core processor. The software implemented onboard is available in [9]. When reading the signals, the frequency is calculated, the power generated, and if necessary, the gain of the scaling is adjusted. From this measured frequency, the speed of the rotor is obtained and thus, the different operating regions of the WT are detected. The control algorithms allow managing the necessary actuators to regulate the turbine angular speed. Finally, data are transmitted through different communication channels.

4 Experimental Results

The first step to carry out the experiments is to adjust the system around an optimal pitch angle, that will be set as reference for the pitch control. At that point, the highest power coefficient is obtained. Thus, the PD and PID controllers have been first tuned by trial and error (experimental tuning). Figure 3 corresponds to experimental results with target speed of 500 rpm. It can be seen how the values of the actuators are adapted to reach the target value (top graph, blue line), using the two independent control loops (pitch control, green line, and load control, red line) to tackle wind speed variations.

The operation regions of a typical wind turbine are shown in Fig. 3 in the experiment:

- *Region I:* The turbine remains stopped until the wind speed exceeds the starting speed. The largest possible pitch angle is set to overtake the static mechanical friction. In the same way, the smallest electromagnetic torque with the largest available load resistance is applied (state 0).
- *Region II:* Once the rotor starts rotating, the WT exponentially decreases the pitch angle towards the optimum. At the same time, it increases the state of charge to maximize the power extracted until reaching the nominal speed. In this region, only load control is applied.
- *Region III:* The pitch control to stabilize the output power of the WT is activated. The load control is disabled as long as the angular speed is kept within the established working range.
- *Region IV:* As the wind speed increases, if the working range is exceeded, the load control is activated again, keeping the pitch angle constant. In this case the load control is not able of limiting the speed of the turbine, therefore the emergency stop protocol is fired, placing the blades perpendicular to the wind direction.

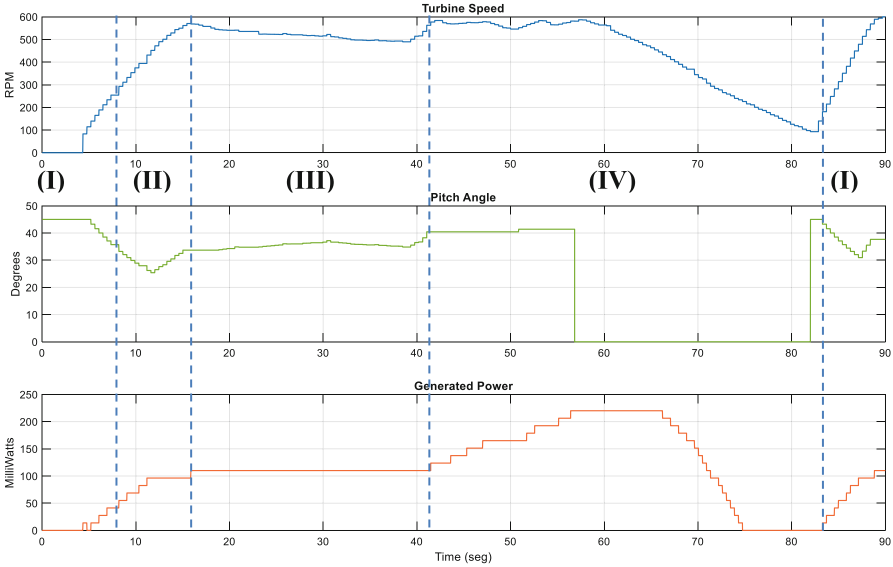


Fig. 3. Angular speed (rpm) (blue line), pitch angle of the blades (degrees) (green line) and generated power (orange line) for a variable wind speed.

5 Optimal Tuning by Genetic Algorithms

The tuning of the system control is performed with the Matlab optimization toolbox once the plant model has been identified, using experimental data. Genetic algorithms are applied based on the preservation of the best individuals (elitism), the mutation of the genes, and the individuals crossover to find the best individuals [10]. The pitch controller gains, [Kp,Ki,Kd], are selected as optimization variables. Then, an initial population of 30 individuals is randomly initialized with a uniform distribution. The optimization variables have been limited to a range close to that obtained in the initial tuning tests. Along 100 generations, the fitness function is evaluated.

The integral of time multiplied by the absolute error (ITAE) is used as cost function for a step input. It was chosen because this way, the error in the steady state at each simulation is considered and penalized. In this application, the simulation time is long enough to disregard the transient errors. The settling time of the experimental tuning is used as integration interval. In this way, it is possible to consider the possible solutions that give a smaller error in the stationary state in each simulation.

$$ITAE = \int_{0}^{t_s} t \cdot |e(t)| \cdot dt \tag{8}$$

At each generation, 5% of the best individuals directly survive to the next generation. Tournament is used to select the parents for crossover from a pool and 80% of the next population is generated using this operator. Small mutation probability is applied for offspring mutation, complaining feasibility restrictions. Figure 4 shows the evolution of the best individuals regarding the ITAE criterion.

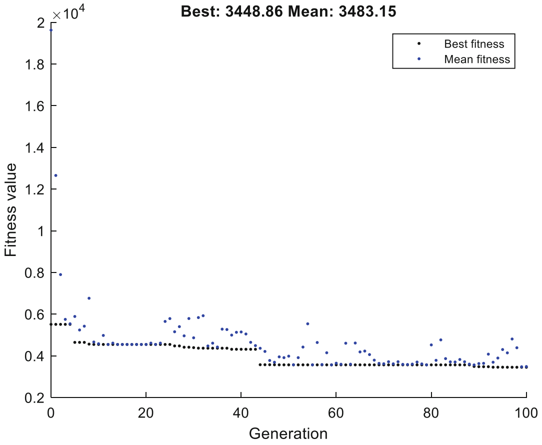


Fig. 4. Evolution of solutions for PID tuning parameters by genetic algorithm.

Once the number of generations is reached, the algorithm obtains the optimal values for the PID tuning parameters: Kp = 0.3, Ki = 0.0038, Kd = -0.0434. The gains obtained for the descrete controller tuned by trial and error were Kp' = 0.03, Ki' = 0.01, and Kd' = -0.01. Figure 5 shows how the system response with the controlled tune by GA presents a much shorter settling time for a step input and it does not have overshoot, in comparison with the PID tuned experimentally.

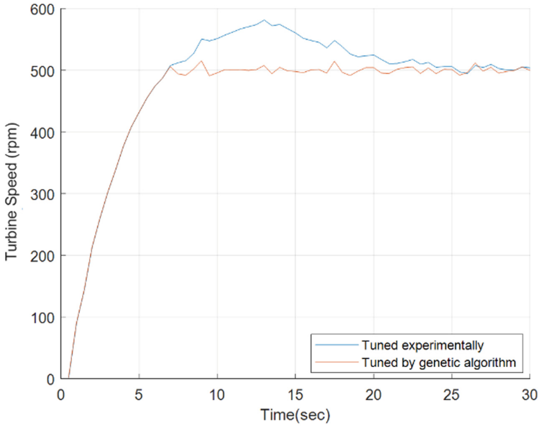


Fig. 5. Response of the wind turbine PID control with different tuning methods.

The real WT has certain inertia that slows down the response to the input signal. In addition, the system has to overcome a starting mechanical resistance. Increasing the action of the controllers in the real system would cause a much less stable response. Besides, the microcontroller requires a minimum sampling time of 0.5 s to measure the RPM and execute the control algorithms.

Even obtaining similar results, the optimization of the controller by means of genetic algorithms has made it possible to considerably reduce the overshoot of the system, at the same time that the settling time has been reduced.

6 Conclusions

In this work, the design of a laboratory scale wind turbine prototype is described. It is focused on the implementation of the necessary electronics and mechanics to be able to carry out closed-loop control tests of a WT. Two control loops have been implemented, an external one that controls the electrical load and an internal one that controls the pitch of the blades. In the testing experiments, the LSM switches between the typical operating modes of a WT and the response is consistent.

The WT has been identified in sufficient detail to tune controllers by genetic algorithms. It has been shown how the controllers tuned by the soft computing technique give a better system response than those adjusted experimentally based on the knowledge of the engineer.

Therefore, the prototype can be used to test more complex control algorithms. In addition, as the LSM is able of registering the position of the structure, it can be used to replicate floating wind turbines in a set-up that integrates the waves motion [11]. These implementations are part of the futures works along with improvements in wireless communication and testing other optimization techniques for control tuning [12]. The final goal is to develop a totally autonomous WT replica, with data collection uploaded to the cloud, that can be controlled telematically through the corresponding digital twin.

Acknowledgments. This work has been partially supported by the Spanish Ministry of Science and Innovation under the project MCI/AEI/FEDER number RTI2018–094902-B-C21.

References

1. Sierra-García, J.E., Santos, M.: Redes neuronales y aprendizaje por refuerzo en el control de turbinas eólicas. Revista Iberoamericana de Automática e Informática industrial **18**(4), 327–335 (2021)
2. Sierra-García, J.E., Santos, M.: Performance analysis of a wind turbine pitch neurocontroller with unsupervised learning, 2020, Complexity (2020)
3. Andrade, G.A., Esteban, S.: Modelo a escala de aerogenerador para control. In: Aitor J. Garrido et al. (eds.) Innovation and Lecture Notes In Control Engineering For Clean Energy Generation, pp. 53–58. Universidad del País Vasco, Bilbao (2021)
4. López, R., Santos, M., Polo, O., Esteban, S.: Experimenting a fuzzy controller on a fast ferry. In: Proceedings of the IEEE Int. Conf. on Control Applications CCA, vol. 2, pp. 1082–1087. IEEE (2002)
5. Alzayed, M., Chaoui, H., Farajpour, Y.: Maximum power tracking for a wind energy conversion system using cascade-forward neural networks. IEEE Trans. Sustain. Energy **12**(4), 2367–2377 (2021)
6. Sierra-García, J.E., Santos, M.: Lookup table and neural network hybrid strategy for wind turbine pitch control. Sustainability **13**(6), 3235 (2021)

7. Chavero-Navarrete, E., Trejo-Perea, M., Jáuregui-Correa, J.C., Carrillo-Serrano, R.V., Ronquillo-Lomeli, G., Ríos-Moreno, J.G.: Hierarchical pitch control for small wind turbines based on fuzzy logic and anticipated wind speed measurement. Appl. Sci. **10**(13), 4592 (2020)
8. ARDUINO Store, http://store.arduino.cc/products/arduino-nano-33-iot. Accessed 25 Feb 2022
9. GITHUB, TFGGiordyAlexander, https://github.com/GGiordy/TFG_Giordy_Alexander.git. Accessed 25 Feb 2022
10. Zotes, F.A., Penas, M.S.: Multi-criteria genetic optimisation of the manoeuvres of a two-stage launcher. Inf. Sci. **180**(6), 896–910 (2010)
11. Villoslada, D., Santos, M., Tomás-Rodríguez, M.: General methodology for the identification of reduced dynamic models of barge-type floating wind turbines. Energies **14**(13), 3902 (2021)
12. Torralba-Morales, L.M., Reynoso-Meza, G., Carrillo-Ahumada, J.: Tuning and comparison of design concepts applying Pareto optimality. a case study of cholette bioreactor. Revista Iberoamericana de Automática e Informática Industrial, **17**(2), 190–201 (2020)

Pitch-Based Wind Turbine Tower Vibration Damping Optimized by Simulated Annealing

Mikel Serrano[1], Jesús Enrique Sierra-Garcia[2](✉), Matilde Santos[3], and Giordy Alexander Andrade[4]

[1] Complutense University of Madrid, Madrid, Spain
mikeserr@ucm.es
[2] Electromechanical Engineering Department, University of Burgos, Burgos, Spain
jesierra@ubu.es
[3] Institute of Knowledge Technology, Complutense University of Madrid, Madrid, Spain
msantos@ucm.es
[4] Faculty of Physics, Complutense University of Madrid, 28040 Madrid, Spain
giordyan@ucm.es

Abstract. In this work we present a control architecture to reduce the vibration of the tower of a wind turbine (WT) using the pitch angle. The parameters of the active tower damping (ATD) control are optimized by the simulated annealing algorithm, which imitates the physical process of steel hardening. To validate the approach, a linearized model of a 5 MW turbine is used at 13 m/s of wind speed. Results show how the optimized active tower damping is able to dampen the oscillations and thus, the peaks and the amplitude of the vibrations are much smaller. This allows to increase the efficiency and reduce the fatigue of the WT.

Keywords: Wind turbine · Active tower damping · Pitch control · Vibrations reduction · Simulated annealing

1 Introduction

In the last decades, the wind energy demand has steadily increased, pushing the optimization of all the processes involved in the development of this renewable energy technology in order to improve efficiency and reduce maintenance costs [1]. The control algorithms are playing a key role in wind turbines (WT), not only to increment the energy production but also to reduce vibrations of the wind structure, hence decreasing the fatigue and extending the lifetime of the wind energy converter devices [2, 3].

In addition, diminishing vibrations make it possible to use lighter materials which reduces installation costs and facilitates operations [4].

This problem is also challenging from the control point of view. Structural control, that is, control oriented to mitigate oscillations, has been an active area of research over the last decades in civil engineering and more recently, it has been started to be applied to floating offshore wind turbines (FOWT) [5, 6].

Depending on the method used, structural control schemes can be broadly categorized into three main groups: passive, active, and semi-active control [7]. In the passive control

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 525–533, 2023.
https://doi.org/10.1007/978-3-031-18050-7_51

the parameters are constant, no external forces are required and therefore, there is no need to provide energy. In the semi-active control, the coefficients of the damping devices (usually springs and dampers) are tuned by closed-loop control algorithms in response to the dynamics of the structure. Finally, the active methods are more complex and require an actuating force. One of the actuators than can be used to reduce the vibration is the pitch actuator, that at the same time that is used to control the angle of the blades it can reduce the tower top acceleration. This is done in the fore-aft axis, as the moments presented here are much higher than in the side-side direction.

In this work, we present a control architecture to reduce the vibration of the tower of a wind turbine using the pitch angle control, an approach called Active Tower Damping (ATD), whose parameters are optimized by the simulated annealing (SA) algorithm, which imitates the physical process of steel hardening [8, 9]. Results show how the optimized ATD is able to damp the vibrations, reducing the peaks of the frequency system response as well and decreasing the amplitude of the vibrations.

The paper is organized as follows. In Sect. 2 the proposed control architecture is described. Section 3 explains how the optimization algorithm is used to tune the control strategy. Simulation results are shown and discussed in Sect. 4. The paper ends with the conclusions and future works.

2 Pitch-Based Wind Turbine Tower Vibration Damping

The ATD strategy adds a collective pitch offset to the blades to dampen the fore-aft tower frequency first mode. It is important to have in mind that the pitch angle is used to adjust the blade surface that faces the wind. When the pitch is reduced, the blade surface in contact with the wind grows and the wind turbine accelerates; and conversely, when the pitch angle increases, wind turbine rotor decelerates.

In addition to the acceleration of the rotor, the wind also exerts a drag force that produces a deflection in the tower. This is usually measured at the nacelle (tower top displacement). So, the idea behind ATD is that when the tower moves upwind, pitch is reduced, so that more wind is captured, the drag force grows, and tower oscillation is slowed down. On the contrary, when the tower is moving downwind, pitch is increased so that less wind is captured, the drag force decreases, and the tower speed is reduced.

Thus, in a WT, the vibrations in the nacelle can be mathematically modelled by (1).

$$F = M \cdot \ddot{x} + C(\theta) \cdot \dot{x} + K \cdot x \tag{1}$$

where x is the displacement of the tower top (m), $\dot{x}$ its derivative (m/s), and $\ddot{x}$ the acceleration (m/s^2), also denoted as NacX. F is the drag force (N), M is the mass of the WT (kg), K is the elastic constant (N/m), and C is a function that represents the damping coefficient that depends, among other factors, on the angle of the blades, θ (º). Therefore, by adjusting the pitch angle it is possible to act on the damping coefficient and in turn, in the vibrations of the turbine [10, 11].

In order to reduce the acceleration of the wind turbine tower, the control scheme in Fig. 1 has been implemented. The input of the wind turbine is the pitch angle, θ, and it is subjected to disturbances (dPitch). The output of the wind turbine is the acceleration of the nacelle (NacX), what also is subjected to disturbances (dNacX).

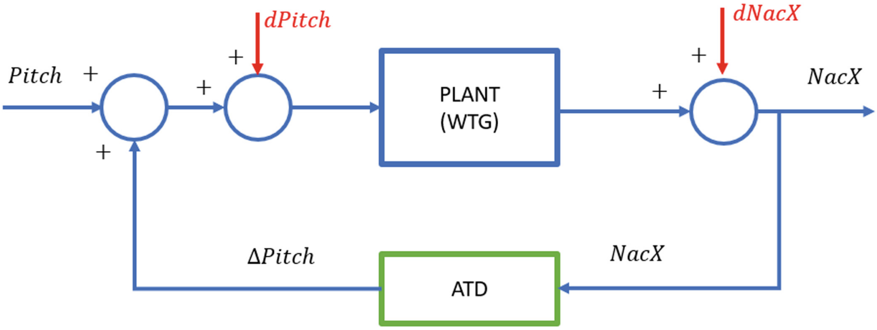


Fig. 1. Vibration damping strategy

The ATD receives the signal NacX and modifies the pitch signal in order to reduce the acceleration. This way, if the acceleration increases, the pitch angle and thus the damper coefficient also grow. This contributes to reduce the acceleration.

The ATD block consists of a low-pass filter in series with a Notch filter that dampens the 3P frequency of the blades. The 3P frequency is produced by the “shadow” caused by the passage of each blade on the tower, which ends up being coupled to the system. The purpose of the low-pass filter is to introduce an offset of –90° (approximately) to get the added pitch action by the ATD to be in phase with the acceleration of the nacelle. This offset is because although the acceleration is measured, the modification of the damper coefficient affects the rotor velocity, so that this measurement needs to be integrated. However, if an integration function were used directly, a very high gain would be added at low frequencies, which is the area where the first fore-aft frequency mode (1Fore-Aft) is located. Thus, a low-pass filter is selected to avoid this.

The expression (2) denotes the equation of this filter in the Laplace domain.

$$ATD(s) = \frac{\Delta Pitch(s)}{NacX\,(s)} = k_{LP} \cdot \frac{\omega_{npLP}^2}{s^2 + 2\xi_{pLP}\omega_{npLP}s + \omega_{npLP}^2} \cdot k_{Notch} \cdot \frac{s^2 + 2\xi_z\omega_{nz}s + \omega_{nz}^2}{s^2 + 2\xi_p\omega_{np}s + \omega_{np}^2} \tag{2}$$

where $[\mathrm{K_{LP}}, \mathrm{K_{Notch}}]$ are the gains of the low-pass and Notch filter, respectively; $[\xi_{pLP}, \omega_{npLP},]$ are the damping coefficient and the natural frequency of the low-pass filter, $[\xi_p, \xi_z]$ are the damping coefficient of the denominator and numerator of the Notch filter, $[w_{np}, w_{nz}]$ are the denominator and numerator natural frequencies of the Notch filter.

The aim of the control algorithm is to reduce the loads produced in the tower by the excitation of the 1Fore-Aft frequency. For this purpose, this oscillation is damped by means of the ATD.

3 Tuning of the Controller by Simulated Annealing

Simulated annealing (SA) is a global optimization metaheuristic search technique. In a wide search space, the main goal of this kind of methods is to discover an approximation to the optimal value of a function [12]. This global optimum is the best solution when no optimal value exists. The global optimum in this minimization problem will be the one for which the objective function has the smallest possible value in the search space [8, 9].

The process of annealing steel and ceramics, which includes heating and then gently cooling the material to modify its physical qualities, inspired the name and motivation of the Simulated Annealing (SA) optimization technique. The heat causes the atoms to raise their energy, allowing them to migrate away from their starting locations (a local energy minimum); the slow cooling allows them to recrystallize into lower-energy configurations (a global minimum).

As in other optimization techniques, the decision variables, the constraints and the cost function have to be defined.

3.1 Decision Variables

In this application, the decision variables to be optimized are the parameters of the filter implemented by the ATD, the set $S = \left[\mathrm{K_{LP}}, \xi_{pLP}, \omega_{npLP}, \mathrm{K_{Notch}}, \xi_p, \xi_z, w_p, w_z\right]$.

$\mathrm{K_{LP}}$: Gain of the low pass filter
ξ_{pLP}: Damping coefficient of the low pass filter
ω_{npLP}: Natural frequency of the low pass filter
$\mathrm{K_{Notch}}$: Gain of the Notch filter
ξ_p: Damping coefficient of the denominator of the Notch filter
ξ_z: Damping coefficient of the numerator of the Notch filter
w_p: Natural frequency of the denominator of the Notch filter
w_z: Natural frequency of the numerator of the Notch filter

3.2 Constraints

The constraints represent the limits that need to be met to find feasible solutions. In this case, it is necessary to ensure the following constraints.

- The phase obtained with the low-pass filter must be within a realistic range, for instance, between –80° and –100°, as the pitch can vary up to 90°. A good starting point is thus –90°
- To ensure stability, phase margin MF $> 30°$
- Gain margin MG $>$ 6d B though MG $>$ 4 dB may be enough in some cases.
- Maximum peak of the sensibility function $<$ 3 dB
- Natural frequencies cannot be negative
- Damping coefficients cannot be bigger than 1.
- The system must be stable

3.3 Auxiliary Functions

functions have been defined, that are the formalization of the previous requirements and constraints.

- ATDoff_WndTwr_gain: Gain of the WindSpeed_Vs_TwrBsMyt Bode with 1Fore-Aft deactivated. The WindSpeed_Vs_TwrBsMyt Bode rep-resents the loads at the tower base with respect to the wind reaching the wind turbine.
- ATDon_WndTwr_gain: Gain of the WindSpeed_Vs_TwrBsMyt bode with 1Fore-Aft active
- Gm_PlantATD_dB: Gain margin of the system
- Pm_PlantATD_deg: Phase margin of the system
- phase_ATD_rest_ok: Confirmation that the phase of the ATD in 1Fore-Aft is within limits.
- mag_S_ATD_gain: Maximum peak sensitivity at output with ATD enabled
- Gm_PlantATD_Min(=6): Minimum gain margin
- Pm_PlantATD_Min(=30): Minimum phase margin
- S_max(=3): Limit for maximum peak output sensitivity
- Stable: Despite checking the phase and gain margins, the Bode may not be entirely accurate and it is therefore necessary to check the stability of the system.

3.4 Cost Function

In order to most reduce the loads at the first mode frequency of the tower, the ATD must be optimized. The criterion is to reduce the gain of the Bode at the 1Fore-Aft frequency. The cost or objective function is made up of 5 sub-functions:

- F1: ATDon_WndTwr_gain
- F2: -Gm_PlantATD_dB. The minus sign is because the smaller value, the better.
- F3: -Pm_PlantATD_deg, with sign minus for the same reason as above.
- F4: if the phase of the ATD in 1FA meets the phase_ATD_rest_ok criterion, this sub-function is 0, otherwise it is 1, when 1 is the best value.
- F5: mag_S_ATD_gain
- F6: -Stable. The minus sign is because when the system is stable this variable is 1, and when it is unstable the variable is 0.

Sub-functions $1, 2, 3$ and 5 are normalized in the global objective function by dividing them by the maximum value they are expected to reach, so they all have a value within the same range, [0 1], as sub-functions 4 and 6 already do. After several tests, the maximum values of these variables have been found to be: $F1_{max} = 101, F2_{max} = 13, F3_{max} = 76, F5_{max} = 17$.

In addition, a weight is assigned to each sub-objective function, $[w1, w2, w3, w4, w5, w6]$. Finally, the final cost function is defined as (3).

$$CF = w1\frac{\mathrm{F1}}{F1\_max} + w2 \cdot \frac{\mathrm{F2}}{F2\_max} + w3\frac{\mathrm{F3}}{F3\_max} + w4 \cdot F4 + w5\frac{\mathrm{F5}}{F5\_max} + w6 \cdot F6 \quad (3)$$

3.5 Simulated Annealing Algorithm

The simulated annealing algorithm has been implemented in the following way:

1. Initialization of the decision variables, S. Initialization of L_k (epoch length) to L_1 (initial length), and T_k to T_1 (initial temperature). T_1 has been set to 6 and L_1 to 1.
2. Adjustment of L_k considering the temperature using Eq. (4), where f_L is set to 0.75. When temperature decreases, the epoch length grows.

$$L_k \leftarrow L_1 \cdot f_L \cdot \frac{\mathrm{T}_1}{\mathrm{T}_\mathrm{k}} \tag{4}$$

```
3. Repeat $L_k$ times:
    3.1 Generate a random neighborhood solution $S' \leftarrow S + N \cdot rand()$
    3.2 Evaluate the cost function $C(S')$
    3.3 Compute $\Delta \leftarrow C(S') - C(S)$
    3.4 If $\Delta < 0$ then $S \leftarrow S'$
    3.5 If $\Delta \geq 0$ then [if $rand() < e^{-\frac{\Delta}{T_k}}$ then $S \leftarrow S'$]
4. If stop condition is satisfied, then algorithm ends
5. Else
    5.1 Execute cooling function $\mathrm{T_k} \leftarrow \alpha \mathrm{T_{k-1}}$, where $\alpha$ is set to 0.85.
    5.2 If $\mathrm{T_k}$<$\mathrm{T_{min}}$ then
        5.2.1 if $reheat > reheatMax$ then the algorithm stops
        5.2.2 else $T_k = T_1$, $reheat \leftarrow reheat + 1$, go to step 3
    5.3 Else go to step 3
```

4 Simulation Results and Discussion

In order to study the improvement obtained with the ATD, in this section the Bode diagrams of the system with and without this control strategy are presented and analyzed. These results have been obtained using Matlab software. It is applied to a linearized model of an onshore 5MW NREL wind turbine at 13m/s of wind speed operation point.

Figure 2 shows the Bode diagram considering that the pitch angle is the input, and the output is the acceleration of the nacelle. The blue line indicates the results when the ATD is not active, the green one when the ATD is applied but the decision variables have not been optimized, and finally the red line shows the results when the ATD is applied with the optimum parameters. It is possible to observe how the ATD is able to damp the vibrations better, and the peak is also smaller. It is noteworthy to remark that the magnitude is represented in dB, if it would be represented in the linear scale the improvement would be even more noticeable. Although the oscillations at the desired frequency are reduced, the amplitude at other frequencies is slightly increased.

As expected, the improvement in the damping is also visible in the transient response. Figure 3 shows the comparison of the acceleration of the nacelle with and without ATD. The color code is the same as in Fig. 2. When the ATD is applied the peaks in the acceleration are smaller and, moreover, the amplitude of the vibrations is reduced much faster. Indeed, if the ATD is optimized the vibration is no perceptible from t = 15 s on, but if ATD is not active there are oscillations until t = 45 s. In this figure it is also noticeable the effect of the optimization. The amplitude decreases much faster when the ATD is optimized than if the initial parameters are used.

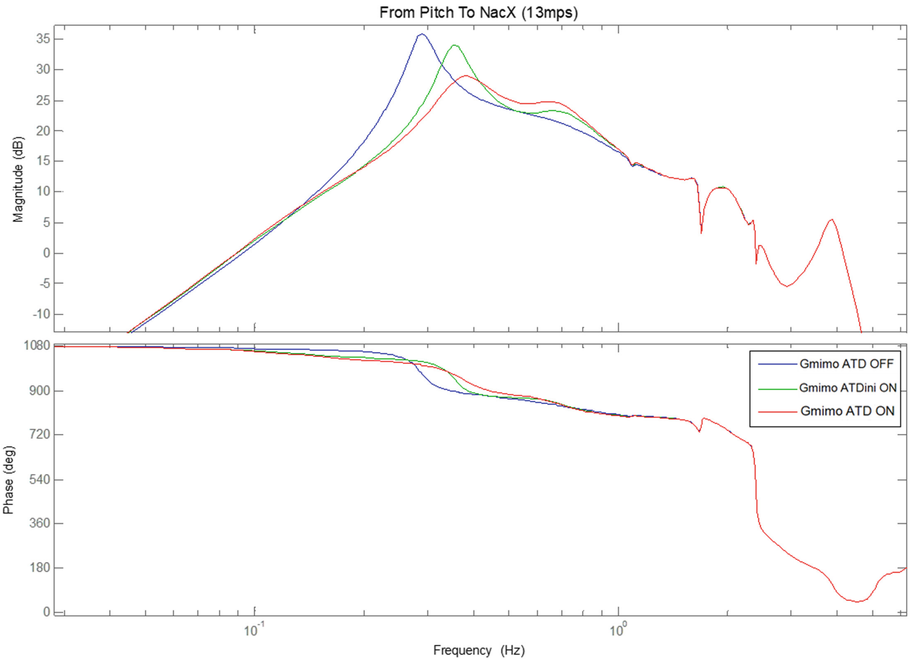


Fig. 2. Comparison of Bode diagrams when the ATD is applied.

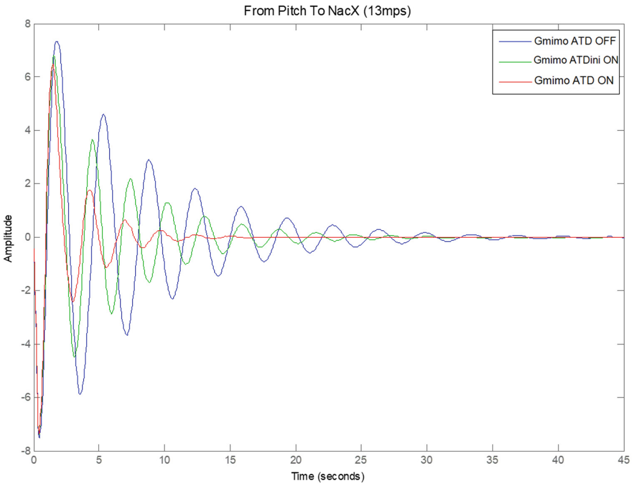


Fig. 3. Comparison of acceleration of the nacelle when the ATD is applied.

An important feature of the ATD is the ability to reject disturbances. To study this effect, it is interesting to study the Bode diagram of the system considering that the input is the disturbance in the acceleration, and the output is the acceleration of the nacelle. Figure 4 shows these results, with the same color code as in previous figures. Looking at this Bode plot of the sensitivity function, when ATD is optimized the first mode of the tower frequency is dampened. However, it also shows a lateral elongation to the right, which is slightly larger than the 3dB limit. Despite this, the strong reduction in 1FA compensates this gain increase. This small increase in the gain at higher frequencies is causing new frequencies to be excited, so care must be taken not to increase them too much.

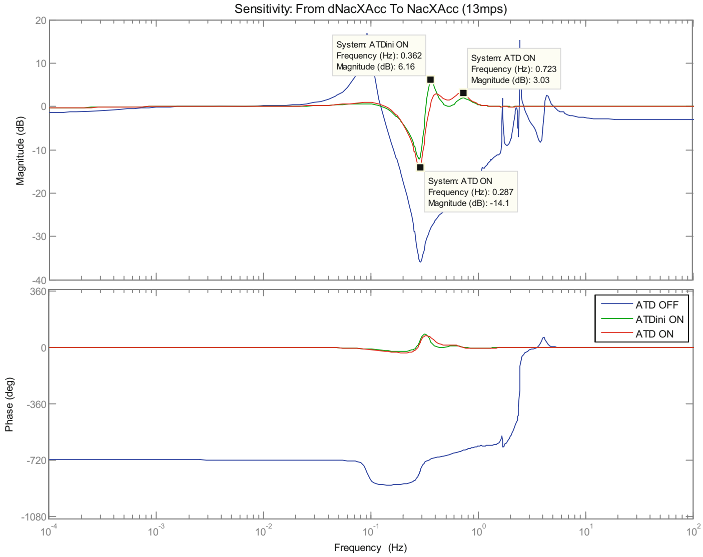


Fig. 4. Comparison of sensitivity to disturbances when ATD is applied

5 Conclusions and Future Works

The demand for wind energy has continued to increase unstoppably in recent decades. For its profitability, it is necessary to optimize the processes that intervene in the generation of energy, while at the same time optimizing the structures and extending the useful life. Control algorithms that enable efficiency gains can also be used to reduce wind turbine tower vibrations.

In this work, it has been shown how a pitch control strategy can reduce vibrations of the tower of a WT, and the role of the optimization of the parameters of control architecture has been highlighted. Specifically, pitch control has been used, optimizing its parameters through the heuristic technique of simulated annealing. The simulation results have allowed us to verify how this proposal improves the stability of the turbine.

As future works, different optimization algorithms for WT control could be used and compared, and it would be also desirable to test the proposal with a wind turbine prototype.

Acknowledgments. This work has been partially supported by the Spanish Ministry of Science and Innovation under the project MCI/AEI/FEDER number RTI2018–094902-B-C21.

References

1. Sierra-García, J.E., Santos, M.: Redes neuronales y aprendizaje por refuerzo en el control de turbinas eólicas. Revista Iberoamericana de Automática e Informática industrial **18**(4), 327–335 (2021)
2. Galán-Lavado, A., Santos, M.: Analysis of the effects of the location of passive control devices on the platform of a floating wind turbine. Energies **14**(10), 2850 (2021)
3. Sierra-García, J.E., Santos, M.: Lookup table and neural network hybrid strategy for wind turbine pitch control. Sustainability **13**(6), 3235 (2021)
4. Prasad, B.B., Duvigneau, F., Woschke, E., Juhre, D.: Wind turbine blade and generator test specimen for evaluating a passive vibration reduction concept based on granular materials. In: Proceedings of the ISMA (2020)
5. Lackner, M.A., Rotea, M.A.: Structural control of floating wind turbines. Mechatronics **21**(4), 704–719 (2011)
6. Yang, J., He, E.M., Hu, Y.Q.: Dynamic modeling and vibration suppression for an offshore wind turbine with a tuned mass damper in floating platform. Appl. Ocean Res. **83**, 21–29 (2019)
7. Tomás-Rodríguez, M., Santos, M.: Modelling and control of floating offshore wind turbines. Revista 623 Iberoamericana de Automática e Informática Industrial **16**(4), 381–390 (2019)
8. Delahaye, D., Chaimatanan, S., Mongeau, M.: Simulated annealing: from basics to applications. In: Handbook of Metaheuristics, pp. 1–35. Springer, Cham (2019)
9. Hernández-Vázquez, J.O., Hernández-González, S., Hernández-Vázquez, J.I., Jiménez-García, J.A., Hernández-Ripalda, M.D.: Análisis multi-objetivo del problema de asignación del buffer con meta-modelos de simulación y una metaheurística híbrida. Revista Iberoamericana de Automática e Informática industrial **19**(2), 221–232 (2022)
10. Leng, D., et al.: Vibration control of offshore wind turbine under multiple hazards using single variable-stiffness tuned mass damper. Ocean Eng. **236**, 109473 (2021)
11. Golnary, F., Tse, K.T.: Simultaneous active control of tower lateral vibration and power control of wind turbine: a novel multivariable approach. Energy Rep. **8**, 4233–4251 (2022)
12. Zotes, F.A., Penas, M.S.: Multi-criteria genetic optimisation of the manoeuvres of a two-stage launcher. Inf. Sci. **180**(6), 896–910 (2010)

Neural Networks Techniques for Fault Detection and Offset Prediction on Wind Turbines Sensors

Fabio Rodríguez[1](✉), William D. Chicaiza[2], Adolfo J. Sánchez[3], and Juan Manuel Escaño[2]

[1] Department of Applied Mathematics II, University of Seville, Seville, Spain
frodriguex@us.es

[2] Department of System Engineering and Automatic Control, University of Seville, Seville, Spain
{wchicaiza,jescano}@us.es

[3] Department of Mechanical, Biomedical, and Manufacturing Engineering, Munster Technological University, Cork, Ireland
adolfo.sanchezdelpozofernandez@mtu.ie

Abstract. Digital Twins (DT) are one of the basis of Industry 4.0. DTs are used to simulate physical assets and improve the efficiency and decision making of industrial production. DT models are usually fitted with data collected from their physical counterparts through sensor readings. The data quality of the information retrieved by sensors is one main problem when training and retraining DT models. Poor data quality leads to low-accuracy DT predictions. In this study, a methodology is proposed for fault detection and offset error prediction problems related to retraining the DT of two wind turbine systems. Wind turbines are of utmost importance in Industry 4.0, as the use of renewable energy reduces production cost and benefits the environment. Having time series data sets with sensor readings of two real wind turbines, machine learning techniques based on Recurrent Neural Networks (RNNs) with Long Short Term Memory (LSTM) layers were implemented for multiple sensor fault forecasting. High-precision models were obtained and experiments were designed and performed to test the effectiveness of the proposed multi-fault detection system. The strengths and weaknesses of the approach are presented, which shows the relevance of this methodology for the DT retraining process.

Keywords: Digital twin · Wind turbine · Sensors · Fault detection · LSTM networks

1 Introduction

The most widely deployed renewable energy sources are wind and photovoltaics, which represent 2/3 of this generation [1]. Wind turbines are a mechanism to obtain renewable energy from the wind force. Today, the use of wind turbines

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 534–543, 2023.
https://doi.org/10.1007/978-3-031-18050-7_52

is expanding in the search for a substitute for the energy obtained by fossil combustion, which has large consequences for the environment, by the energy obtained from renewable systems such as solar plants and wind turbines [2].

Industry 4.0 is the new paradigm for optimising production and efficiency in industry while saving energy and optimising the use of renewable energy. Digital twins (DT) [3] are one of the main parts of Industry 4.0 and enable the construction of virtual plants that help discover the behavioural problems of dynamic systems. DT needs to communicate with the real plant to update its parameters according to changes in the plants. The flow of data between the twin and the real plant allows for dynamic models perfectly adapted to the system.

Data are usually collected from sensors, so Fault Detection in sensors is of outstanding importance in industry, [4,5]. Sensor failures are frequent and can be caused by system failure, cyber attacks, or other reasons. Sensor failures cause an offset in its output values. Since DTs learn from the sensor outputs of a physical entity, the system must be capable of recognising incorrect offsets in sensors. A drift in a sensor can trigger the twin learning mechanism in an unwanted way, leading to a loss of accuracy in the operation of DT.

The problem of fault detection in DT involves real-time data analysis. Neural network techniques are popular for solving data-related problems. Specifically, Long-Short-Term Memory (LSTM) networks are a type of Recurrent Neural Network (RNN) capable of learning from time series and extracting knowledge related to, for example, the outlet of a sensor during time. LSTM networks are frequently used for Fault Detection problems. In [6], a combination of models based on convolutional and LSTM networks is used to detect faults in wind turbines. In [7], an innovative architecture of LSTM networks is combined with the detection and diagnosis of fast Fourier transform faults and continuous wavelet transform for rotating machinery. In [8], LSTM regression and classification architectures are used for fault region identification, fault type classification, and fault location prediction with high precision for a large-scale multi-machine power system.

In this research, we apply LSTM-based techniques to detect failures and predict offset errors in the output of wind turbine sensors. Machine learning models are fed historical data that correspond to two wind turbines. The data used are publicly available on Github[1]. In the literature, many researches have been implemented to detect faults in wind turbines, for example [6,9].

The intention and novelty behind this research is to design a methodology for fault detection in multiple sensors simultaneously to be installed on any sensor-based system, in this specific case a wind turbine system. The proposed Multi Failure Detection Process (MFDP) methodology is capable of detecting sensor failures simultaneously at any time during wind turbine operation. This idea has a wide application in Industry 4.0 or cybersecurity. The objective of this research is the specific problem of retraining DTs during the life cycle of physical industrial entities [10]. A similar approach was implemented in [11]

[1] https://github.com/fabio-rodriguez/wind-turbines-data-sets.

Table 1. Minimum, maximum, and mean values per variable in the data for wind turbine 1

Variable	Min	Max	Mean
EG kWh	0	732	155.94
WS m/s	0	26.80	7.32
RS rpm	0	14.73	8.92
RP kW	0	323	96.99
P kW	0	3070	941.75
NP °	0	359	225.57

for the retraining of the DT of a Solar Cooling Plant with good results using techniques such as neurofuzzy logic and RNN.

This research is organised as follows: Sect. 2 describes the data sets used, Sect. 3 defines the problem, Sect. 4 on board the network architecture used, Sect. 5 describes the proposed methodology, Sect. 6 exposes the experimental studies and analyses the results, and Sect. 7 explains the conclusion of the article.

2 Data Sets

In this article, a machine learning-based methodology is proposed for retraining of DTs during its lifecycle. To develop this study, data were collected from sensor readings of two real wind turbine systems installed for a factory. The data describe the operation of the systems for one year of operation. The data sets used contain information about the date, electricity generated (EG), wind speed (WS), rotor speed (RS), reactive power (RP), power (P), and nacelle position (NP) sensors at 10-min intervals.

Tables 1 and 2 present the range of values of each sensor throughout the year. In total, the data sets contain 52343 entries for the first wind turbine and 52284 entries for the second.

Having a total of 6 columns: EG, WS, RS, RP, P, and NP, a standardisation was performed on the data sets by subtracting the mean and dividing the standard deviation in each variable. Subsequently, a study of the principal components (PCA) was performed on the data. After applying PCA, the first 5 principal components were selected, which contain more than 95% of the variance in the data. After this work, the data sets were ready to be fed to the learning models, using the 80% of the data as the training set and the rest as the validation set. The training and validation sets were randomly split.

Table 2. Minimum, maximum and mean values per variable in the data for wind turbine 2

Variable	Min	Max	Mean
EG kWh	0	645	135.21
WS m/s	0	22.10	6.35
RS rpm	0	14.69	9.52
RP kW	−4	1219	461.33
P kW	−25	2476	811.88
NP °	0	359	220.69

3 Problem Statement

Using the data sets addressed in Sect. 2, this research addresses two main problems. Given the list of sensor outputs during a year:

1. Detection: Given new sensor outputs, the problem of evaluating which of the output values corresponds to sensor failures.
2. Prediction: Given the new sensor outputs, the problem of predicting the magnitude of the failure, i.e., predicting the offset between the received and expected output from the sensors.

The first problem is related to the prediction of failures in the system sensors, while the second problem refers to the estimation of the dimension of the hazard found. Both problems have the added difficulty that, given the new sensor outputs, faults can occur in any of them. Therefore, the system must be capable of detecting faults even in the event that every sensor is faulty.

The approach presented in this article is based on RNN with LSTM units that learn from historical sensor readings from wind turbines. Recurrent networks look at the past states of the wind turbines to predict the new states and determine if there are faults and their offsets. Therefore, for the correct use of this approach, the assumption must be made that fault-free states precede the states to predict.

4 LSTM Architectures

The heart of the proposed multifault detection methodology is the machine learning model used in the forecasting of the values of new sensor readings based on the historical information from previous readings. With the predicted values, fault detection and offset prediction are performed.

Figure 1 shows the architecture of the deep neural network used, which consists of four consecutive layers. The first two layers contain 40 LSTM units each, followed by two fully connected layers with 20 and 10 units, respectively, and a final output layer with 1 unit. The input layer has a shape of $t \times 5$, where

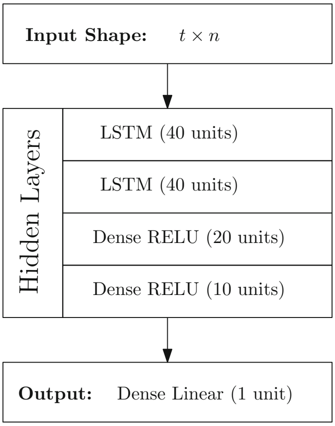


Fig. 1. Architecture of a machine learning RNN with LSTM layers

t represents the look-back value, that is, the number of states in the historical record to be examined to predict the current value; and five is the number of variables in each state, due to the application of PCA over readings in the data sets. All units in the third and fourth layers have the RELU activation function. The output layer has a linear activation function to make a numerical prediction of the correct value for the defective sensor.

The architecture combines LSTM layers to extract patterns from the time series conformed by the historical record of sensors, and feedforward ANN layers to make regression of the expected output value for the sensor at the current time. The architecture is very similar to the one used in [11], to detect failures in a solar cooling plant. This simple architecture is proposed to determine the time dependence of the wind turbine sensors' behaviour.

Six networks with this structure were built, one for each sensor. The networks were trained using Adam optimiser with parameters: $learning\,rate = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 1e-7$ and without weight decay. The learning process is stopped after 4000 epochs with $batch\_size = 128$ and $patience = 100\,epochs$. The Mean Absolute Error (MAE) measure was chosen as the loss function to minimise in the learning, having slightly better results than the Mean Squared Error (MSE) measure. The k-fold technique was used in the training process with $k = 4$ folds, keeping the one with the lowest MAE in the validation set.

With fitted models, new sensor output can be predicted and the difference between the output and predicted values defines whether the sensors are faulty and the offset of the failures. Therefore, the presented approach represents a valid solution for both problems addressed in this article, Detection and Prediction.

5 Methodology

This work proposes a methodology for multisensor fault detection and offset prediction in wind turbine systems. Figure 2 shows the workflow of this methodology, which is based on the following steps:

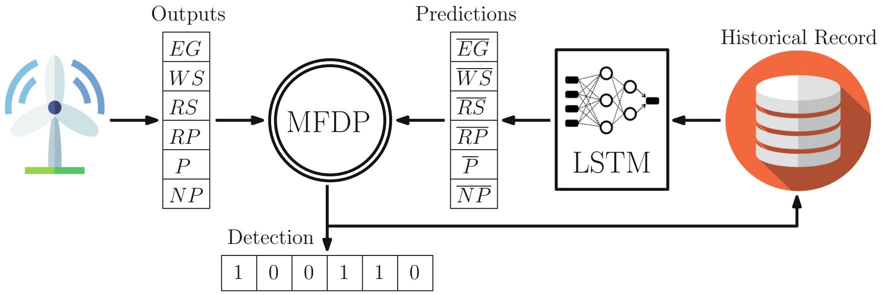


Fig. 2. Fault Detection workflow: wind turbines provide the sensor outlets, the RNN models with LSTM layers predict the sensor outlets given a historical record H. The Multi failure Detection Process (MFDP) compares the outlets with the predictions to detect fault in sensors, the system outputs the faulty sensors and updates the historical record with the new fault-free values.

1. The system has stored a set H with historical information corresponding to the t previous fault-free readings of the wind turbine sensor.
2. The wind turbine returns the sensor readings in a given instant of time, s_i.
3. Predictions are made for each sensor p_i based on the states in H.
4. Prediction offsets are obtained, $o_i = |p_i - s_i|$, and a decision is made about which of the sensors has a failure.
5. The set H is updated with the faultless sensor outputs, combined with the faulty sensor prediction. Updates are made in such a way that H always has the t last states, $|H| = t$.
6. The detection and the offset prediction are returned for each sensor.

Algorithm 1 shows in detail how the methodology can be implemented. The input of the process is the sensor output values, vector V, and the historical record of t previous fault-free sensor readings, set H. In the algorithm, the prediction and detection of each sensor, s_i, are carried out within a loop, lines 6–19. As a first step, the prediction model for each s_i is loaded on line 7. Each model is used to predict the output value of s_i based on the set H; obtain the predicted value $\bar{y}_i$, line 8. Later, in lines 9–10, the offsets o_i are calculated as the absolute value of the difference between the sensor outputs $y_i = V[s_i]$ and their respective predictions $\bar{y}_i$. In lines 12–18, fault detection is performed by comparing the offset values o_i with a certain threshold, the sensor is declared faulty (1) if o_i overcomes the sensor threshold; otherwise, it is declared fault-free (0). Threshold values are selected depending on each sensor values; see Sect. 5 for more information. On line 20, the set H is updated with fault-free values for future predictions. Finally, the algorithm returns the detection results represented by a binary array, the offset error for each sensor, and the updated historical record.

Algorithm 1: Algorithm pseudo code for the fault detection methodology

```
Input: $V, H$
Output: $D, O, H^{'}$
$sensors \leftarrow get\_sensors()$
$thresholds \leftarrow get\_thresholds()$
$detection \leftarrow list()$
$offsets \leftarrow list()$
$current\_state \leftarrow list()$
for $s_i$ in $sensors$ do
    $model \leftarrow get\_model(s_i)$
    $\bar{y_i} \leftarrow model.predict(H)$
    $y_i \leftarrow V[s_i]$
    $o_i \leftarrow |y_i - \bar{y_i}|$
    $offsets.append(o_i)$
    if $o_i < thresholds[s_i]$ then
        $detection.append(0)$
        $current\_state.append(y_i)$
    else
        $detection.append(1)$
        $current\_state.append(\bar{y_i})$
    end if
end for
$H^{'} \leftarrow update\_H(current\_state)$
return $detection, offsets, H^{'}$
```

6 Experiments and Results

This Section presents the experiments implemented to test the proposed methodology and the results in each case. All experiments were implemented using Python 3.9, on Windows 11. The code and data that support this article will be available for the sake of reproducibility after acceptance[2].

Using the architecture in Sect. 4, the networks were trained and the results in terms of mean squared error (MAE) in the validation set for each turbine are shown in Tables 3 and 4. The tables show that an increase in the number of previous states in H taken to predict sensor outlets at the current time produces an improvement in the MAE measure. This means that wind turbine behaviours are not random and follow a certain dependency on the previous states of the system. The results for $t = 30$, where $t = |H|$, are very accurate, having the mean MAE for each sensor under 10% of the mean of the sensor values throughout the data set.

An experimental case was developed to test the methodology as a whole. In the experiment, the model receives the historical record of $t = 30$ fault-free states of the system. Subsequently, the detection algorithm was used to predict all sensor outlets with simulated errors for one month. After every prediction,

[2] https://github.com/fabio-rodriguez/wt-dq.

Table 3. Prediction results in the validation set for wind turbine 1

LSTM	EG kWh	WS m/s	RS rpm	RP kW	P kW	NP °
t = 1	4.89 ± 8.13	0.44 ± 1.04	0.36 ± 0.88	4.16 ± 5.72	28.73 ± 48.57	3.25 ± 7.25
t = 5	4.83 ± 8.02	0.40 ± 0.89	0.32 ± 0.72	4.21 ± 6.10	27.66 ± 46.52	3.29 ± 7.23
t = 10	4.73 ± 8.01	0.37 ± 0.83	0.30 ± 0.66	3.85 ± 5.39	27.05 ± 45.35	3.26 ± 7.24
t =20	4.38 ± 9.02	0.32 ± 0.68	0.27 ± 0.59	3.59 ± 5.21	26.56 ± 45.06	3.15 ± 7.24
t = 30	4.10 ± 6.67	0.29 ± 0.53	0.25 ± 0.52	3.32 ± 4.34	23.83 ± 38.44	3.06 ± 7.20

Table 4. Prediction results in the validation set for wind turbine 2

LSTM	EG kWh	WS m/s	RS rpm	RP kW	P kW	NP °
t = 1	5.54 ± 7.17	0.38 ± 0.59	0.30 ± 0.45	26.38 ± 25.40	34.05 ± 39.62	3.11 ± 7.02
t = 5	5.69 ± 7.12	0.37 ± 0.57	0.29 ± 0.39	27.54 ± 26.75	34.44 ± 39.62	3.07 ± 7.04
t = 10	5.63 ± 7.44	0.35 ± 0.52	0.29 ± 0.41	26.58 ± 25.84	33.60 ± 41.24	3.05 ± 7.03
t = 20	5.24 ± 7.17	0.32 ± 0.41	0.27 ± 0.39	23.36 ± 22.30	30.24 ± 33.70	3.06 ± 7.03
t = 30	4.88 ± 6.16	0.30 ± 0.38	0.26 ± 0.35	21.18 ± 20.79	29.05 ± 33.54	3.04 ± 7.08

Table 5. Fault offsets and thresholds for wind turbine 1 sensors

Variable	Offset	Threshold	Accuracy
EG kWh	±39	±29.25	0.99
WS m/s	±2	±1.50	0.97
RS rpm	±3	±2.25	0.81
RP kW	±25	±18.75	0.98
P kW	±236	±177.00	0.99
NP °	±57	±42.75	1.00

the set H is updated with the union between the fault-free outputs and the predictions corresponding to the fault outputs. In this way, the historical record is updated step by step for the following predictions.

Several days from the data set of each turbine were selected as the experimental set. In the set, failures were randomly simulated with probability 50%, so about half of the entries were simulated as faults. Sensor failures in one entry were uniformly determined, and all sensors had the same probability of containing errors. Faults were generated at sensor outputs by adding or subtracting, with the same probability, their corresponding offsets. The offset errors per sensor are defined as the 25% of the mean value of the sensor outlets in the entire set; and the thresholds are defined as the 75% of the offsets, Tables 5 and 6.

The proposed methodology was applied each day in the experimental set without great long-term results. The detection algorithm returned high-accuracy results for the initial hours of the days; however, the detection results were not good for the rest of the hours.

Table 6. Fault offsets and thresholds for wind turbine 2 sensors

Variable	Offset	Threshold	Accuracy
EG kWh	±34	±25.50	0.99
WS m/s	±2	±1.50	0.89
RS rpm	±3	±2.25	0.95
RP kW	±116	±87.00	0.98
P kW	±203	±152.25	0.99
NP °	±56	±42.00	1.00

The weakness of this approach is that, when faults are detected in any sensor at initial hours, the historical record is updated with low error sensor predictions; however, these residual errors accumulate subsequently, decreasing prediction accuracy at the following hours. In real applications, this issue could be resolved by checking the sensor functionality in the event of detection of repeated faults.

Despite this, the models, in general, are very accurate. A second experiment was formulated for the entire validation set. Failures were simulated along the validation set in the same way as in the previous experiment. The same offset and threshold values per sensor were selected, Tables 5 and 6. In this experiment, the historical record used for each prediction was taken with fault-free states, removing residual errors. The detection precision in the validation set for each variable is presented in Tables 5 and 6. The tables show great results for the fault detection process, detecting more than 80% faults in all sensors and having almost perfect detection in many of them.

7 Conclusions and Future Work

In this work, a multiple failure detection and offset prediction methodology was proposed for the data quality problem in DT retraining of two wind turbines. This research is very important in Industry 4.0 as it helps to include and optimise the use of renewable energies.

The models presented in this article were based on RNN with LSTM layers for time series forecasting. Studying the parameter t, the number of states in the historical record, shows that the precision of the prediction increases, as does the parameter t, demonstrating the time dependence of the sensor values of the wind turbines. The detection results were very accurate for all sensors in most cases. However, the methodology showed a weak point related to the residual errors in the predictions. Despite this, the methodology is the foundation for future approaches with improved models and techniques.

In future approaches, a deeper study of the data sets should be developed, looking for properties to improve the prediction models. Furthermore, an outlier elimination process should be performed on the data sets. Many ideas can also be implemented to improve the prediction results of machine learning models, such as taking larger values of the parameter t for predictions or drastically increasing the number of units in the layers of machine learning architecture. Furthermore, an exhaustive comparison with other machine learning models and with similar techniques in the literature must be done.

Acknowledgment. The authors want to thank the European Commission for funding this work in the DENiM project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 958339.

References

1. I. E. A.: Global energy review 2021: Assessing the effects of economic recoveries on global energy demand and CO2 emissions in 2021. International Energy Agency, Technical report, April 2021
2. Panwar, N., Kaushik, S., Kothari, S.: Role of renewable energy sources in environmental protection: a review. Renew. Sustain. Energy Rev. **15**(3), 1513–1524 (2011)
3. Tao, F., Zhang, H., Liu, A., Nee, A.Y.C.: Digital twin in industry: state-of-the-art. IEEE Trans. Industr. Inf. **15**, 2405–2415 (2019)
4. Miljković, D.: Fault detection methods: a literature survey. In: Proceedings of the 34th International Convention MIPRO, pp. 750–755. IEEE (2011)
5. Park, Y.J., Fan, S.K.S., Hsu, C.Y.: A review on fault detection and process diagnostics in industrial processes. Processes **8**(9), 1123 (2020)
6. Xiang, L., Wang, P., Yang, X., Hu, A., Su, H.: Fault detection of wind turbine based on SCADA data analysis using CNN and LSTM with attention mechanism. Measurement **175**, 109094 (2021). Available via DIALOG. https://www.sciencedirect.com/science/article/pii/S026322412100124X
7. Jalayer, M., Orsenigo, C., Vercellis, C.: Fault detection and diagnosis for rotating machinery: a model based on convolutional LSTM, fast Fourier and continuous wavelet transforms. Comput. Ind. **125**, 103378 (2021). Available via DIALOG. https://www.sciencedirect.com/science/article/pii/S0166361520306126
8. Belagoune, S., Bali, N., Bakdi, A., Baadji, B., Atif, K.: Deep learning through LSTM classification and regression for transmission line fault detection, diagnosis and location in large-scale multi-machine power systems. Measurement **177**, 109330 (2021). Available via DIALOG. https://www.sciencedirect.com/science/article/pii/S0263224121003286
9. Miele, E.S., Bonacina, F., Corsini, A.: Deep anomaly detection in horizontal axis wind turbines using graph convolutional autoencoders for multivariate time series. Energy AI **8**, 100145 (2022). Available via DIALOG. https://www.sciencedirect.com/science/article/pii/S2666546822000076
10. Lim, K.Y.H., Zheng, P., Chen, C.H.: A state-of-the-art survey of digital twin: techniques, engineering product lifecycle management and business innovation perspectives. J. Intell. Manuf. **31**(6), 1313–1337 (2020)
11. Rodríguez, F., Chicaiza, W., Sánchez, A., Escaño, J.M.: Detection of anomalies in sensor data using neurofuzzy systems. In: IFSA-EUSFLAT 2021 jointly with AGOP, IJCRS, and FQAS, Bratislava, Slovakia. Book of abstracts (2021). ISBN 978-80-7599-269-7

Special Session on Pre-processing Big Data in Machine Learning

Errors of Identifiers in Anonymous Databases: Impact on Data Quality

Paulo Pombinho[1](✉), Luís Cavique[2], and Luís Correia[1]

[1] LASIGE, Faculdade de Ciências, Universidade de Lisboa, Lisboa, Portugal
pmimatos@fc.ul.pt, luis.correia@ciencias.ulisboa.pt
[2] Universidade Aberta, Lisboa, Portugal
luis.cavique@uab.pt

Abstract. Data quality is essential for a correct understanding of the concepts they represent. Data mining is especially relevant when data with inferior quality is used in algorithms that depend on correct data to create accurate models and predictions. In this work, we introduce the issue of errors of identifiers in an anonymous database. The work proposes a quality evaluation approach that considers individual attributes and a contextual analysis that allows additional quality evaluations. The proposed quality analysis model is a robust means of minimizing anonymization costs.

Keywords: Data pre-processing · Anonymized data · Data quality

1 Introduction

Data quality is crucial in business and governmental usage to allow for a correct analysis and subsequent decisions to be made appropriately. However, a dataset with quality problems can imply high costs both on economic and social levels, with the possibility of erroneous decisions to be made when looking at incorrect data [1, 2].

The use of personal data following the GDPR [3] implies that datasets are often anonymized. Anonymization creates a challenge for evaluating the quality of the data since it prevents the identification and confirmation of errors, making it impossible to use another dataset to identify which values are correct.

These limitations make traditional metrics inadequate for assessing the quality of anonymized data [4]. Current research has focused more on anonymization procedures than data prospecting and quality assessment. In addition, if an improper anonymization procedure is performed, the potential usefulness of data prospecting can be ruined even by marginal privacy gains [5]. The essential factor for the success of data science projects is the use of the correct dimensions and their treatment, being the most time-consuming part of these projects [6].

Data pre-processing is essential for dealing with complexity in Business Analytics. However, most scientific and corporate literature proposes solutions with many synonyms depending on the methodology or tool used to answer these challenges. Data Pre-processing also appears with Data Wrangling or Feature Engineering designations.

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 547–556, 2023.
https://doi.org/10.1007/978-3-031-18050-7_53

Different designations are also usual for each pre-processing stage, and data collection is often used as a synonym for data selection. Moreover, data cleaning appears as data cleansing or data quality. Four stages were chosen, and the pre-processing data stages are sequenced in the following pipeline:

Data Collection → Data Quality → Data Transformation → Data Reduction.

In what follows, Sect. 2, Data Collection, presents the data enrichment, types of errors, and noisy identifiers. Section 3, Data Quality, enumerates the different quality assessments used. In Sect. 4, we discuss the computational results of applying the pre-processing. In Sect. 5, we conclude this study.

2 Data Collection

We instantiate the pipeline applied to a dataset with over twenty million student enrollment data entries to better analyze each step. This data has been anonymized following the General Data Protection Regulation [3], with all students identified only by a number. Since this dataset is used as input to data mining algorithms, we must ensure that the data we have is as accurate as possible to minimize the noise given as input to the algorithms and maximize the quality of the modeling and predictions made.

In this section, we will describe the data selection and data cleaning processes.

2.1 Data Enrichment

The database used consists of information on student enrollments. Each entry concerns a student's enrollment in a specific school year with information detailing the characteristics of the enrollment and its outcome. It has the following relevant dimensions:

- School year (e.g., 2008–2009)
- Grade (e.g., 12$^{\text{th}}$ grade)
- Nature (e.g., public or private)
- Geographic Information
- Enrolment event
- Age and gender

Since we want to model the student flow, we need to correctly perceive how a student's path through the education system changes and what events are associated with these changes. We analyze how a student's enrollment changes in sequential years and add an attribute to characterize the student's state. We consider the path of a student as a sequence of states with three different types of state inputs and three types of outputs (Fig. 1):

- Dropout: there is no information about the student in the next school year,
- Retention: the student is in the same grade in the following school year,
- Pass: the student is enrolled in a higher grade in the following school year,
- Ingress: there is no information about the student in the previous school year.

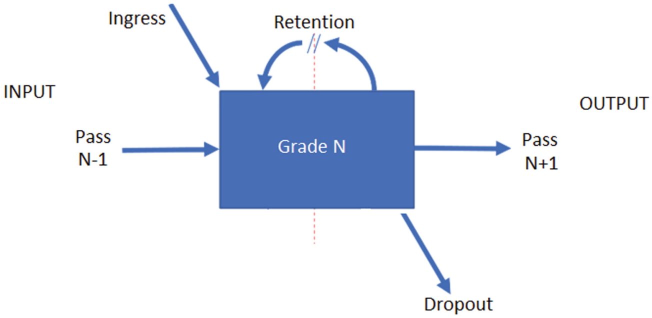


Fig. 1. Types of input/output enrolment events.

2.2 Errors of Identifiers

Identifier problems are derived from the anonymization procedures and can sometimes produce problems in the data. Since we have no ground truth to compare the results, these problems are more challenging to detect and correct than non-anonymized data. Identifier problems can result in severe problems if we analyze the relationship between different entries of the same real-world entity. Suppose the anonymization process has introduced noise in the primary/foreign keys of the data. In that case, we may find that a single entity has multiple data entries collected at certain time intervals in datasets consisting of time series.

Figure 2 shows how incorrect identifiers in two sequential entries can create a wrong and potentially severe deviation in the perception of reality. Figure 2 (a) shows how overlapping identifiers in two different students may produce an incorrect reading of

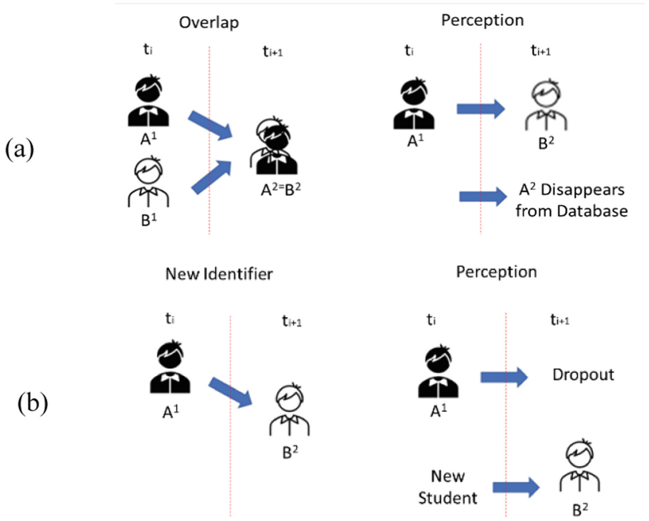


Fig. 2. (a) overlapping of two different students' identifiers (b) incorrect creation of a new identifier

reality. Instead of two different students passing grades, the overlap of the first student over the second will produce an incorrect result where student A will, in the next year, transition to the grade year of student B instead of its own. In contrast, student B will be shown as dropping out of school. Likewise, Fig. 2 (b) gives an example of how the attribution of a new identifier to an existing one can also cause issues. In this instance, student A will be perceived as dropping out in the next year, while a new student will be shown enrolling for the first time in the next year.

Since, in some instances, these errors create patterns that can be mistaken for normal data, they are not easily detected. Therefore, there is a need to use a set of different metrics that, in their combination, can allow us to detect these types of problems.

3 Data Quality

In this paper, we propose the combination of different types of analysis that allow for more robust classification of each data entry quality and the different data dimensions themselves. We adapt the nomenclature proposed in [7–9] and consider four types of individual quality assessment:

Completeness evaluates whether all attributes of an entry are populated, or values are missing in some attributes.

Consistency assesses whether the information contained in the entry's attributes must be normalized during the ETL (Extract, Transform, Load) processing phase. Alternatively, it also assesses if entered data was not formatted according to the rest of the dataset.

Uniqueness identifies the entries as duplicated if, for example, two entries in the same period are conflicting.

Accuracy measures the accuracy of the attribute values of a given entry and allows the identification of entries with incorrectly populated attributes. For example, let us consider the accuracy of a numeric attribute. We can identify scenarios where the value is invalid because it is outside the domain of possible values, such as negative values when only positive values are valid. Regarding the longitudinal assessment of the dataflow, we consider an accuracy assessment that analyses three different contextual dimensions:

I. Attribute variations – evaluates situations where a change in the value of an attribute in two consecutive time points of the same entity is greater than expected;
II. Value variation – performs a calculation that evaluates the probability of a value of an attribute considering the other attributes of the duplicate entry;
III. Attribute Incompatibility – compares, in the duplicate entry, the different attributes for which it is known that there is a correlation in search of incompatible values.

In the following subsections, we will describe the calculation of the individual quality metrics and the use in calculating the overall quality for each entry.

3.1 Quality Metrics

We calculate four different data quality metrics using the types defined in the previous section. These range between 0 and 1; '0' represents zero quality and '1' a perfect quality.

To calculate the **completeness** metric, we used information regarding the missing attribute count. The summation argument uses an Iverson bracket[1] to count the number of missing attributes, in entry a, about the total number of attributes, M. (Eq. 1).

$$Completeness = 1 - \frac{1}{M}\sum_{i=1}^{M}\left[a_i \, is \, missing\right] \tag{1}$$

Data **consistency** is also classified, similarly to the completeness, using the count of attributes with consistency issues over the total number of attributes, M (Eq. 2).

$$Consistency = 1 - \frac{1}{M}\sum_{i=1}^{M}[a_i \, is \, not \, consistent] \tag{2}$$

Uniqueness is classified according to the identification of duplicate entries if the identifier is in more than one entry at the same time data point and, in case this happens, evaluating how different the duplicate entries are (Eq. 3).

$$Uniqueness = \begin{cases} 1, & notduplicated \\ 0.8, & duplicate \, with \, minor \, differences \\ 0, & duplicate \, with \, major \, differences \end{cases} \tag{3}$$

The **accuracy metric** evaluates the accuracy and probability of specific attribute values. As indicated earlier, we classify an attribute as invalid if the value is outside the domains of that attribute. However, this analysis can be improved by using semantic information of expected attribute values, considering other contextual information, and allowing a classification with different degrees of probability (Eq. 4). V is the value of the attribute, V_{max} and V_{min} are then valid maximum and minimum values, and VP_{max} and VP_{min} are the respective maximum and minimum values likely for the current context of the entry.

$$Accuracy\_Type\_I(V) = \begin{cases} 1, & VP_{min} < V < VP_{max} \\ \frac{V - V_{min}}{VP_{min} - V_{min}}, & V_{min} < V < VP_{min} \\ 1 - \frac{V - VP_{max}}{V_{max} - VP_{max}}, & VP_{max} < V < V_{max} \\ 0, & V\langle V_{min} \vee V \rangle V_{max} \end{cases} \tag{4}$$

This metric may calculate how likely a student's age is, given his enrolled grade. The first component of the equation gives maximum quality to ages inside the valid, and expected, intervals, while, on the opposite, ages outside the valid values will have zero quality. Simmilarly, the second and third components of the equation evaluate the quality of valid age value, but below or above the likely values interval, respectively.

Similarly, an attribute can be classified using information about the variation of the current V value (obtained comparing with the previous data point of the same entity) and

[1] Iverson bracket https://en.wikipedia.org/wiki/Iverson_bracket.

its comparison with the minimum (most probable) and maximum variations possible, respectively, $\mathrm{V}_{\mathrm{min}}$ and $\mathrm{V}_{\mathrm{max}}$ (Eq. 5).

$$Accuracy\_Type\_II(V_a) = 1 - \frac{|V| - |V_{min}|}{|V_{max}| - |V_{min}|} \tag{5}$$

Like the previous metric, the value variation can be used to assess, for example, the change in grades of a specific student, allowing the detection of abnormal jumps in the student's path in the education system. This way expected variations of only one school year will have maximum quality while higher variations will decrease the quality value.

Finally, the accuracy of attributes heavily influenced by others can also be assessed by comparing pairs of attributes, searching for values that should not coexist, using a two-argument function (Eq. 6).

$$\mathrm{Accuracy\_Type\_III}(V_1, V_2) = \begin{cases} 1, V_1 \; compatible \; with \; V_2 \\ 0, V_1 \; incompatible \; with \; V_2 \end{cases} \tag{6}$$

The compatibility metric can identify situations where two attributes are incompatible; for example, a student enrolled in early grades but a modality exclusive of higher education grades.

After the calculations of the different types of accuracy metrics, the overall accuracy is defined as the average of its different components in all M attributes (Eq. 7).

$$Accuracy = \frac{\sum_{i=1}^{M} Acc\_I_i + \sum_{i=1}^{M} Acc\_II_i + \sum_{i=1}^{M} Acc\_III_i}{3M} \tag{7}$$

3.2 Aggregated Data Quality

The quality assessment metric described in the previous section allows us to understand each entry and its reliability in each dimension. Each metric helps identify potential problems with the data. However, because more severe data problems are expected to simultaneously create quality problems in multiple dimensions (such as problems that arise from inconsistencies with identifiers), we use the combination of the different metrics to compute the overall quality for each entry in the database.

For each entry, the Aggregated data Quality (AQ) is obtained as the weighted sum of all metrics previously described and is classified as a value from 0 (no quality) to 1 (best quality) (Eq. 8). All weights are positive, and their sum is equal to 1.The use of weights on each metric allows us to emphasize which metrics are most important for a specific dataset or task. If all metrics are of equal importance then each weight will have equal value.

$$\begin{aligned} AQ_i &= w_{Comp}.Completness_i + w_{Cons}.Consistency_i \\ &+ w_{Uniq}.Uniqueness_i + w_{Acc}.Accuracy_i \end{aligned} \tag{8}$$

The data quality allows us to perform initial filtering of data entries that have a quality below some threshold and recognize identifiers associated with potential identification problems and report them to be fixed. In the analysis performed, and described in the next section, we used a weight of 0.4 for the accuracy and 0.2 for the other three metrics.

4 Discussion

4.1 Data Cleaning

The proposed metrics were applied in the dataset described in Sect. 3, in which it was possible to point out problems arising from identifier errors. These could be found by comparing attributes between different time points and verifying the incompatibility of existing values, namely in (i) duplicate entries (referring to the same time point but with different attributes), (ii) inconsistencies in the attributes that defined an age, and (iii) in attributes with too high variation between successive time points.

After calculating the metrics, the data with different cutting lines were extracted, and the number of entries filtered in each case was checked (Table 1).

Table 1. Percentage of removed lines by criteria.

Criteria	Removed lines (%)
AQ $\geq$ 0.4	0.00%
AQ $\geq$ 0.5	0.20%
AQ $\geq$ 0.6	0.22%
AQ $\geq$ 0.7	0.25%
AQ $\geq$ 0.8	0.56%
AQ $\geq$ 0.9	11.04%

We chose to filter all data with an AQ below 0.7 since it allowed to remove only 0.25% of the data, corresponding to less than 40,000 entries.

4.2 Impact of the Anonymization on the Data Quality

In the preliminary analysis of the filtered data, despite the small number of entries withdrawn, it was possible to verify a significant decrease in entries with serious errors.

The errors considered to be more serious, such as entries with attributes with values with a low probability (Accuracy_type_I), which corresponded to improbable ages, considering the grade the students were enrolled in, allowed filtering out 98% of the existing problems (Table 2). Moreover, the entries with inconsistent data duplication were reduced by 46%. The remaining metrics had smaller weights and, as such, caused a smaller number of filtered data. For example, the existence of many entries with unfilled attributes and the lower weight used for the completeness metric resulted only in a slight decrease in the number of entries with this problem.

Evaluating the impact of each metric separately helps understand what type of errors are present in the database allowing the evaluation of potential solutions to solve them. For example, almost all the consistency errors present in the preliminary analysis could be solved by using several scripts that normalized some attributes in the database that were identified as having problems.

Table 2. Types of errors with occurrences and percentage removed.

Type of Error	Lines with Error (%)	Errors Removed (%)
Completeness	2.25%	0.38%
Uniqueness	0.02%	46.24%
Accuracy Type I	0.23%	97.67%
Accuracy Type II	0.33%	7.36%
Combined Errors	0.36%	67.98%

However, problems caused by identifier errors derived from the anonymization procedures are harder to identify as they affect only a smaller subset of the data. Moreover, as these types of errors deal with the whole entry and not only one of its attributes, they will trigger inconsistencies across several of the affected entries attributes, which will, in turn, affect multiple metrics simultaneously.

If, for example, two students' data are mixed due to an overlapping identifier, there will probably be inconsistencies in several attributes. In this type of scenario, the students' data could, for instance, show a jump in a curricular year, an age that would not match the curricular year she is enrolled in, a wrong teaching modality for the students' course or even a duplicate enrollments in the same school year. For this reason, it is precious to be able to detect these problems.

The fact that anonymization issues create more significant problems can be used to identify them as entries with several combined errors and, thus, a lower quality.

Using the aggregated data quality allowed us to identify most identifier problems. With aggregated data quality, it was possible to reduce, by almost 68%, the number of entries that had a combination of errors.

The analysis presented gives only a notion of how this type of quality assessment can be used to improve the data without cutting too many entries. However, performing a more detailed analysis is essential, comparing different weights and cutting lines and the effective gain in each case. The number of entries with errors could have been reduced further by using a higher cutting line, resulting in a more substantial number of filtered entries. Therefore, it is necessary to conduct an in-depth evaluation to find the best compromise between the most significant quality gain and the smaller number of cut entries.

It is also important to highlight that appropriate weight values are essential to obtain quality values relevant to the intended type of use. Consequently, the weight values need to consider the data derived from the preliminary assessment and the type of errors that need to be resolved.

5 Conclusions

This paper discusses the impact of GDPR and anonymization on data quality. Anonymization procedures introduce new errors that must be taken into account. Our proposal uses an aggregation of distinct types of quality assessment. In addition, it allows the calculation of global data quality. It uses weights to adjust the importance of each metric used, better to reflect the objectives of the case under study.

Unless filtered, identifier problems derived from anonymization procedures can significantly impact the creation of valid models, making the process complicated because they can produce incorrect readings of the data, mixing entries from different entities.

The proposed approach calculates an individual data quality metric that assesses how reliable a specific entry is regarding its accuracy, consistency, completeness, and uniqueness.

Concerning future work, the objective is to evaluate how the proposed quality analysis tools can improve predictions made by data mining algorithms by comparing filtered data sets with different quality thresholds and unfiltered sets.

We will also compare datasets with heterogeneous quality with homogeneous quality datasets to understand how the existence of inputs with very diverse quality influences the overall quality and explore the addition of a measure that evaluates this variation.

The proposed metrics use an internal evaluation, considering only the existing data in the database. It is essential to add an assessment that also allows a comparison between the overall values of the data set and those made available from external sources, such as institutions that provide statistical information.

Finally, the application of GDPR has hidden costs that must be considered. As anonymization is unavoidable, the proposed quality analysis model is a robust means of minimizing these costs.

Acknowledgments. The authors would like to thank the FCT Projetct of Scientific Research and Technological Development in Data Science and Artificial Intelligence in Public Administration, 2018–2022 (DSAIPA/DS/0039/2018), for its support, and also acknowledge support by BioISI (UID/MULTI/04046/2103) and LASIGE Research Unit (UIDB/00408/2020, UIDP/00408/2020) center grants.

References

1. Batini, C., Scannapieco, M.: Data and information quality: dimensions, principles and techniques. Data-Centric Systems and Applications, Springer (2016). https://doi.org/10.1007/978-3-319-24106-7
2. Heinrich, B., Hristova, D., Klier, M., Schiller, A., Szubartowicz, M.: Requirements for data quality metrics. J. Data Info. Quality 1936–1963 9 (2), 1–32 (2018)
3. GDPR, General Data Protection Regulations: Regulation (EU) 2016/679 of the European parliament and of the council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, OJ L 119, 4.5.2016, pp. 1–88 (2016)
4. Fletcher, S., Islam M.Z.: Quality evaluation of an anonymized dataset. In: 22nd International Conference on Pattern Recognition, 3594–3599 (2014)
5. Brickell, J., Shmatikov, V.: The cost of privacy: destruction of data-mining utility in anonymized data publishing. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 70–78 (2008)
6. Domingos, P.: A few useful things to know about machine learning. Commun. ACM **55**(10), 78–87 (2012)
7. Pipino, L., Lee, Y., Wang, R.: Data quality assessment. Commun. ACM **45**(4), 211–218 (2002)

8. Cai, L., Zhu, Y.: The challenges of data quality and data quality assessment in the big data era. Data Sci. J. **14**(2), 1–10 (2015)
9. Sidi, F., Panahy, P., Affendey, L., Jabar, M., Ibrahim, H., Mustapha, A.: Data quality: a survey of data quality dimensions. In: Proceedings of the 2012 International Conference on Information Retrieval & Knowledge Management, pp. 300–304 (2012)

Feature-Aware Drop Layer (FADL): A Nonparametric Neural Network Layer for Feature Selection

Manuel Jesús Jiménez-Navarro[1](✉), María Martínez-Ballesteros[1],
Isabel Sofia Sousa Brito[2], Francisco Martínez-Álvarez[3],
and Gualberto Asencio-Cortés[3]

[1] Department of Computer Science, University of Seville, 41012 Seville, Spain
{mjimenez3,mariamartinez}@us.es
[2] Department of Engineering, Polytechnic Institute of Beja, Beja, Portugal
isabel.sofia@ipbeja.pt
[3] Data Science and Big Data Lab, Pablo de Olavide University, 41013 Seville, Spain
{fmaralv,guaasecor}@upo.es

Abstract. Neural networks have proven to be a good alternative in application fields such as healthcare, time-series forecasting and artificial vision, among others, for tasks like regression or classification. Their potential has been particularly remarkable in unstructured data, but recently developed architectures or their ensemble with other classical methods have produced competitive results in structured data. Feature selection has several beneficial properties: improve efficacy, performance, problem understanding and data recollection time. However, as new data sources become available and new features are generated using feature engineering techniques, more computational resources are required for feature selection methods. Feature selection takes an exorbitant amount of time in datasets with numerous features, making it impossible to use or achieving suboptimal selections that do not reflect the underlying behavior of the problem. We propose a nonparametric neural network layer which provides all the benefits of feature selection while requiring few changes to the architecture. Our method adds a novel layer at the beginning of the neural network, which removes the influence of features during training, adding inherent interpretability to the model without extra parameterization. In contrast to other feature selection methods, we propose an efficient and model-aware method to select the features with no need to train the model several times. We compared our method with a variety of popular feature selection strategies and datasets, showing remarkable results

Keywords: Feature selection · Neural network · Classification · Regression

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 557–566, 2023.
https://doi.org/10.1007/978-3-031-18050-7_54

1 Introduction

Neural networks have shown to perform well in supervised learning tasks in a variety of domains such as healthcare, time-series forecasting, artificial vision, etc. Their capability in unstructured data, such as images or text, has been demonstrated in the literature and real applications. However, structured data modeling with neural networks is evolving rapidly and achieving competitive results [1].

Although neural networks produce competitive results, some well-known issues remain: curse of dimensionality, performance limitations, noise data or lack of interpretability are some examples. Feature selection is an effective technique that reduces the impact of these issues by selecting a subset of features [10].

Because feature selection reduces the number of features, the resources consumed by the neural network are considerably reduced as it requires less memory, less time to fit and less inference time. This is especially useful nowadays, with the increasing number of data sources and the application of feature engineering.

Feature selection methods can be divided into two groups: model-free methods and model-aware methods. Model-free methods employ data analysis to determine the importance of features. This type of method is very efficient, but does not consider the model in the selection. Model-aware methods analyze the importance of the features learned by the model. However, these methods can involve more processing time, but the selected features can be related to the model.

Metaheuristics-based is a model-aware selection method which must fit a model several times consuming a prohibitive amount of resources. In most of the cases, the selection is limited, resulting in a suboptimal feature selection, which might impair the performance, efficacy and interpretability of the model.

Most feature selection approaches incorporate some or several hyperparameters such as a threshold to select the best ranked features or a predefined number of iterations in the search space. Such types of methods are called parametric. Poor hyperparameter selection can turn out to be suboptimal selection, whereas optimal hyperparameter selection implies repeating the selection method using more computation time.

In this paper, we propose a feature selection method for neural networks optimized by backpropagation. We consider a scenario with limited resources in which we can sacrifice some efficacy to reduce the number of features. The feature selection method is implemented in the neural network as an additional layer, hereinafter called FADL, after the input layer. The proposed method is a nonparametric model-aware selection method that adds an inherent interpretability to the model without extra parameterization. This interpretability is understood by knowing which features the model uses once it is trained.

We compared our method with three different model-free feature selection methods called: Linear, Correlation and Mutual information [4]. All the selected model-free methods establish a score to the features which is used to determine which ones are relevant. We use these methods in the comparison because they are usually faster than model-aware methods as it must optimize just the threshold to determine which features are relevant. In addition, model-free methods scale well with when the number of features increase. On the contrary, model-aware methods may need to explore the feature space to obtain the best selection. Consequently, when the feature space increases, the resources needed increase exponentially.

In contrast to other feature selection methods, our method obtains competitive results more efficiently since it does not need to train the model several times. Although, our method is scalable as the feature space exploration is made during the training time. These methods are compared in five classification datasets and four regression datasets, obtaining a considerable improvement in the effectiveness of regression datasets.

The remainder of this paper is organized as follows. Section 2 describes the related work and the key differences from our work. In Sect. 3, the methodology and each of its components are explained. In Sect. 4, the experimentation is explained including the experimental settings, results and discussion. Finally, in Sect. 5, we indicate some conclusions from the method and future work.

2 Related Works

Feature selection is a well-known technique in the literature, and there exist numerous methods that can be applied to almost any neural network structure. In this paper, we will focus on feature selection methods developed for neural networks. There are two types of methods used to select features in neural networks: meta-models and custom regularization. Meta-model methods are model-aware feature selection while custom regularization methods are model-free method.

In Verikas et al. [15], Khemphila et al. [7] and [11] a meta-model is used to select the features. This method uses a workflow that finds the optimal feature selection for the model. To measure the efficiency of selection, the model must be trained and evaluated for each iteration of the workflow. Lui et al. [8] use an ensemble of models that require training of several models that increase the training time and memory needed. Tong et al. [14] use a genetic algorithm in which individuals are the model, consuming a lot of memory for numerous populations. The computation can be prohibitively expensive with numerous features or large models. For that reason, this approximation is beyond the scope of this paper.

J. Wang et al. [16] and A. Marcano-Cedeño et al. [9] are included in the custom regularization type. In this group of methods, the optimization in the training phase is modified on the basis of some assumptions, such as the distribution of input data. The weights are regularized based on some selected heuristic which can result in suboptimal results. One example is Cancela et al. [2] where a non-convex regularization is used which may difficult the convergence of the model. These types of methods are efficient, but assumptions can affect the convergence of the model. In this paper, any strong assumption is used in the model in order to obtain a general purpose method that does not difficult the convergence of the model in order to minimize the training time.

3 Methodology

3.1 Description

In this section, the formal definition of FADL is explained. Each tensor dimensions are defined as the superindex that excludes the batch dimension.

Let $X^{MxD} \in \mathbb{R}$ be the input taken from the dataset, where M represents the past window size for the time-series and D represents the number of features. Note that in all datasets except time-series the past window size M will always be 1. Let W_L^{1xD} be the weights used in FADL with always 1 past window size and D features. Let H be the Heaviside function that binarizes the input element by element, resulting in a tensor with the same shape as the input.

Formally, FADL can be defined as:

$$FADL(X^{MxD}) = H(W_L^{1xD}) \circ X^{MxD}, \tag{1}$$

where $\circ$ represents the Hadamard product. Moreover, the Heaviside function serves as a gate that indicates if the feature is selected:

$$H(W_L^j) = \begin{cases} 1, \ if \ W_L^j >= 0.5, & (2) \\ 0, \ if \ W_L^j < 0.5, & (3) \end{cases}$$

where W_L^j represents the weight associated to X_j, j $\in$ [1..D], in W_L^{1xD}.

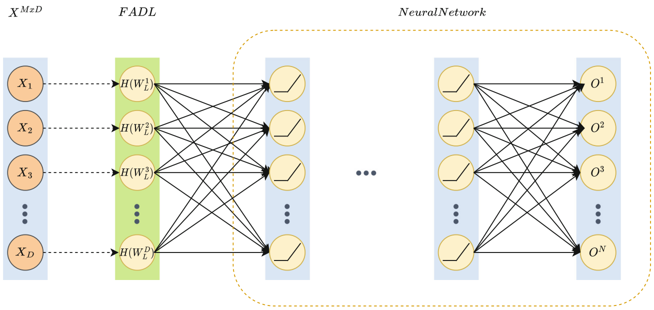


Fig. 1. The proposed FADL with a fully connected neural network ($M = 1$ is assumed). Note that dashed lines represent the element-wise multiplication and solid lines represent the usual fully connected multiplication with weights. Additionally, O_i, $i \in [1, H]$, represents the output of the neural network.

Figure 1 shows FADL applied to a fully connected neural network. Note that the Heaviside function outputs the selection mask, which determines the relevant features of the neural network for the problem. Hadamard product is used between X^{MxD} and $H(W_L^{1xD})$, setting unnecessary features in X^{MxD} to zero.

Backpropagation is used to train FADL as an additional layer in the neural network. When a weight is set to zero, the corresponding feature contribution in the forward step is zero. The feature selection mask is related to the knowledge modeled by the neural network, since the FADL weights are trained as an ordinary layer.

We will study the impact of FADL in two extreme scenarios. On the one hand, if all weights in FADL are positive, FADL would not have any impact on the efficiency of the neural network. On the other hand, if all weights in FADL are negative, that means that the neural network considers all features irrelevant and have no impact on the target.

Note that any of the previous scenarios is desirable. The first scenario does not consider any selection, and FADL serves no purpose other than to notify us that all features are useful for the model. The regularization term described in the following subsection is used to avoid this scenario. The second scenario is not desired as this means that no informative features have been selected. The second situation is uncontrollable since it might be caused by external factors such as a poor data gathering procedure, a poor model configuration, or even the problem is not well modeled using a neural network approach.

3.2 Weight Initialization and Regularization

In FADL, weight initialization may be thought of as the initial hypothesis of the influence of the features. Large positive values introduce a bias, indicating that the features must be present. As a consequence, the model would need more time to change the sign of the feature values. The same logic can be applied to lower negative weight values. In our case, any strong bias is imposed on the initial weights. The weights are set to positive, but close to zero, to facilitate the change from positive to negative weights without extra time. For this reason, the weights were initialized as 0.01.

Weight regularization is another crucial element in FADL that can be thought as a loss tolerance and avoid undesirable scenarios. The regularization used in FADL is Lasso regression over the feature selection mask. Note that we are not interested in regularizing the weights of the FADL, but the feature selection mask result of the Heaviside function. Regularization penalizes the selection of the feature with a constant value independent of the weight value. In this way, the fewer features in the feature selection mask, the higher penalization is added.

Penalization helps avoid previously described undesirable scenarios. We mentioned the extreme scenario in which the feature selection mask selects all features. In this case, the penalization is maximum and would be avoided. The second undesirable scenario is when the n features can explain the target with the same loss as the m features, where $n > m$. Without regularization, any option would be selected, as the loss introduced by both are the same. However, with regularization, the option with fewer features would be preferred.

4 Experimentation

4.1 Datasets

The datasets were chosen from a variety of disciplines where feature selection is important. Healthcare, natural disaster impact, fault detection, and air pollution prediction are just a few examples of these application sectors.

BreastCancer dataset [17] contains 10 features of a digitized image of a fine needle aspirate of a breast mass obtained from clinical cases between 1989 and 1991. The goal is to classify benign and malignant tumour.

Heart dataset [12] contains 19 features obtained from the Behavioral Risk Factor Surveillance System (BRFSS) in U.S. territories through telephonic interviews. The goal is to predict the existence of any coronary heart disease (CHD) or myocardial infarction (MI).

The Cancer Genome Atlas dataset (TCGA) [5] contains 20531 gene expression levels through RNA-Sequencing technique using illumina HiSeq platform. The dataset contains an extraction of gene expression from patients with different types of tumors. The goal is to predict the existence of different types of tumors: Breast invasive carcinoma (BRCA), Kidney renal clear cell carcinoma (KIRC), Colon adenocarcinoma (COAD), Lung adenocarcinoma (LUAD) and Prostate adenocarcinoma (PRAD).

Earthquake dataset [3] contains 38 features collected through surveys by Kathmandu Living Labs and the Central Bureau of Statistics in Nepal. The goal is to predict the level of damage to buildings caused by the 2015 Gorkha earthquake on three levels: low, medium, and high.

WaterPump dataset [13] contains 39 features from waterpoints across Tanzania, collected from Taarifa waterpoints dashboard, which aggregates data from the Tanzania Ministry of Water. The goal is to predict the operating condition of a waterpoint, which can be functional, functional but needs repair, or non-functional.

Torneo dataset [6] contains four pollutants and three meteorological features obtained by sensors in the area of Torneo (Seville) in a hourly basis. The dataset has been divided into four different data sets, each with one pollutant: CO, NO_2, O_3, and PM_{10}.

Table 1 represents the characteristics of each dataset. Except for the features that represent the identifiers, the categorical features were encoded using one-hot encoding. Any missing values in the dataset have been replaced by the most common value for that feature. Except for categorical data, all data has been normalized to improve the convergence of the neural network. Data have been separated into train, valid and test with 70%, 10% and 20% of the data in each dataset, respectively. In the case of all Torneo datasets, the time-series has been transformed to forecast one step ahead for the target, including all features from the past 24 h. In the train/valid/test split, the older records are in the train split, while the new records are in the test split.

Table 1. Number of features, number of instances, sequence size, number of targets, and type of problem for each data set.

	#Features	#Instances	Sequence size	#Targets	Type
BreastCancer [17]	10	570	1	2	Classification
Heart [12]	19	319,796	1	2	Classification
TCGA [5]	20,531	802	1	5	Classification
Earthquake [3]	38	260,602	1	3	Classification
Waterpump [13]	39	59,401	1	3	Classification
TorneoCO [6]	7	4,017	24	1	Regression
TorneoNO_2 [6]	7	4,017	24	1	Regression
TorneoO_3 [6]	7	4,017	24	1	Regression
TorneoPM_{10} [6]	7	4,017	24	1	Regression

4.2 Experimental Settings

The chosen fully-connected model consists of an input layer, an output layer and two hidden layers, one with up to half the number of features and the other with up to a quarter of the number of features. The selected activation function is relu in all hidden layers and, in the output layer, softmax and linear activation function are selected for classification and regression tasks, respectively. The neural network was trained for 100 epochs with Adam optimizer using a learning rate of 0.001 and early stopping with 10 epochs maximum. The model was chosen because it was simple, efficient and all-purpose. The purpose of our paper is to focus on feature selection rather than tuning the hyperparameters of the model.

Our proposed method is compared to four different strategies: no selection (NS), correlation feature selection (Corr), linear feature selection (Linear) and mutual information feature selection (MI). No selection technique is used as a baseline since it does not perform any selection. Correlation feature selection analyzes the relationship between the features and the target in order to provide a score. Linear feature selection requires fitting a linear model and calculating a score to each feature based on the coefficients of the model. Mutual information selection [4], assigns a score to each feature based on its dependence on the target.

As the proposed feature selection methods are parametric, we need to set a threshold to select the relevant features. To make the selection, the features were ranked according to their score. Then, we select the features whose sum of scores is greater than a threshold. The selected threshold is in the range [60%, 95%] in steps of 5%. Ranges below 60% are not considered due to computation limitations. Note that feature selection is done to all time steps in torneo datasets, as feature selection based on time is outside the scope of this paper.

The metrics were separated into two categories: efficacy metrics and performance metrics. The performance metrics measure the number of features selected by the method and time used to select the features plus the time used to train the model. Note that in the no selection strategy and in FADL, the model is trained just once, as there is no threshold optimization. The Mean Squared Error is used as an efficacy metric in regression, while the F1 score is used in classification. As the parametric methods need to optimize the threshold, the result with best efficacy and performance is used for each dataset. However, the parametric feature selection methods need to execute the model several times to obtain the optimal threshold. For that reason, to perform fair performance comparisons between the methods, the sum of times for each threshold is used.

4.3 Results and Discussion

In this section, we analyze our method in terms of both effectiveness and performance perspectives, both for classification and regression tasks. Specifically, Fig. 2 shows the effectiveness achieved by a neural network trained using the proposed FADL, in addition to the rest of benchmark methods, both for classification (F1-Score) and regression (Mean Squared Error) datasets.

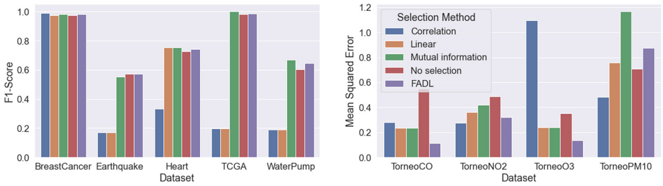


Fig. 2. Efficacy results for classification (F1-Score) and regression (MSE).

In classification datasets, Mutual information method achieves the best efficacy in Heart, TCGA and Waterpump datasets; Correlation method achieves the best efficacy in BreastCancer and FADL method in Earthquake dataset. In general, the best feature selection methods for classification are mutual information and FADL as they obtain good results in almost all datasets. Additionally, mutual information and FADL obtain better results than No selection except in Earthquake dataset where only the FADL obtains the same results as No selection. Note than Correlation and Linear methods obtain poor results in Earthquake, TCGA and WaterPump due to any of the thresholds in the range explored being adequate for such datasets.

In regression datasets, the best selection methods are Correlation and FADL. In TorneoCO and TorneoO_3 datasets, there is a great improvement in FADL compared to other methods. In TorneoNO_2 dataset, the best method is correlation but has little difference with FADL. In TorneoPM_{10} dataset, all selection methods perform poorly except for correlation. In general, most of the methods improve the no selection approach except for TorneoPM_{10} where the selection seems to be harder compared to other regression datasets.

Figure 3 shows the selected features for the best threshold and the sum of training time. Analyzing the number of features, feature selection methods that

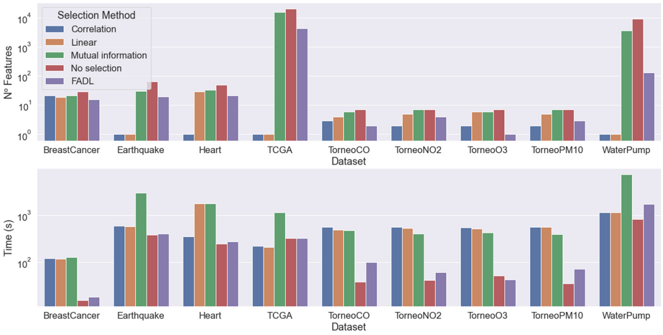


Fig. 3. Number of features and training time for each feature. Note that the y-axis is on a logarithmic scale.

perform poorly in terms of efficiency select just one feature. Apparently, the score function used in each selection method assigns a great value to one feature that represents at least 95% of the total sum of scores. In classification datasets, the FADL is the method which, obtaining good efficacy results in all datasets, selects fewer features than other methods with similar efficacy results. In regression, correlation method obtains fewer features in two datasets and FADL in the remaining regression datasets.

In terms of execution time, all methods obtain an execution time similar to or greater than no selection. The methods that achieved poor efficacy obtained less execution time because the selection was not adequate. In cases where the selection was performed adequately, FADL obtains better execution times than any other selection method. In general, no selection approach obtains fewer times and FADL shows little increment except for regression datasets and Waterpump.

5 Conclusions and Future Works

In this paper, we proposed a nonparametric model-aware feature selection with stable results. The FADL provides interpretability for neural networks, indicating the relevant features used by the model.

The FADL achieved the best execution times, obtaining a competitive selection specially in TorneoCO and TorneoO3 datasets where the efficacy was greatly improved, the number of features reduced, and the execution time was considerably lower than the other selection methods studied. Furthermore, as the results showed, FADL is the most stable method in terms of efficacy and performance metrics.

We propose expanding the interpretability to another dimension as time in future work, revealing not only the significant features but also the relevant time steps. Furthermore, we try to address the problem with features that are important in only a few circumstances, and some features are only significant at particular times. Because there are so few of them, they cause such little error and can be ignored.

Acknowledgements. The authors would like to thank the Spanish Ministry of Science and Innovation for the support under the project PID2020-117954RB and the European Regional Development Fund and Junta de Andalucía for projects PY20-00870 and UPO-138516.

References

1. Borisov, V., Leemann, T., Seßler, K., Haug, J., Pawelczyk, M., Kasneci, G.: Deep Neural Networks and Tabular Data: A Survey (2021)
2. Cancela, B., Bolón-Canedo, V., Alonso-Betanzos, A.: E2E-FS: An End-to-End Feature Selection Method for Neural Networks. CoRR (2020)
3. Nepal Earthquake Open Data (2015). http://eq2015.npc.gov.np/#/
4. E-Shannon, C.: A mathematical theory of communication. ACM SIGMOBILE Mob. Comput. Commun. Rev. **5**(1), 3–55 (2001)
5. Fiorini, S.: UCI Gene Expression Cancer RNA-Seq (2016)
6. Gómez-Losada, A., Asencio-Cortés, G., Martínez-Àlvarez, F., Riquelme, J.C.: A novel approach to forecast urban surface-level ozone considering heterogeneous locations and limited information. Environ. Model. Softw. **110**, 52–61 (2018)
7. Khemphila, A., Boonjing, V.: Heart disease classification using neural network and feature selection. In: Proceedings - ICSEng 2011: International Conference on Systems Engineering (2011)
8. Liu, B., Cui, Q., Jiang, T., Ma, S.: A combinational feature selection and ensemble neural network method for classification of gene expression data. BMC Bioinform. **136**, 10 (2004)
9. Marcano-Cedeño, A., Quintanilla-Domínguez, J., Cortina-Januchs, M.G., Andina, D.: Feature selection using sequential forward selection and classification applying artificial metaplasticity neural network. In: IECON Proceedings (Industrial Electronics Conference) (2010)
10. Miao, J., Niu, L.: A survey on feature selection. Proc. Comput. Sci. **91**, 12 (2016)
11. Monirul Kabir, Md., Monirul Islam, Md., Murase, K.: A new wrapper feature selection approach using neural network. Neurocomputing. **73**(16), 3273–3283 (2010)
12. Pytlak, K.: Personal Key Indicators of Heart Disease (2020). www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease
13. Taarifa: Water pump (2022). https://taarifa.org/
14. Tong, D., Mintram, R.: Genetic algorithm-neural network (GANN): a study of neural network activation functions and depth of genetic algorithm search applied to feature selection. Int. J. Mach. Learn. Cybern. **1**, 75–87 (2010)
15. Verikas, A., Bacauskiene, M.: Feature selection with neural networks. Pattern Recogn. Lett. **23**(11), 1323–1335 (2002)
16. Wang, J., Zhang, H., Wang, J., Pu, Y., Pal, N.R.: Feature selection using a neural network with group lasso regularization and controlled redundancy. IEEE Trans. Neural Netw. Learn. Syst. **32**(3), 1110–1123 (2021)
17. Wolberg, W.H., Nick Street, W., Mangasarian, O.L.: UCI Breast Cancer Wisconsin (Diagnostic) (1995)

Classification Methods for MOBA Games

Marco A. Peña-Cubillos[1], Alejandro Villar-Ruiz[1],
Antonio J. Tallón-Ballesteros[2(✉)], Yaoyang Wu[3], and Simon Fong[3]

[1] International University of Andalusia, Huelva, Spain
[2] Department of Electronic, Computer Systems and Automation Engineering,
University of Huelva, Huelva, Spain
antonio.tallon.diesia@zimbra.uhu.es
[3] Department of Computer and Information Science, University of Macau, Taipa,
Macau SAR, China

Abstract. The rise of the sports industry, which over time has increased in popularity along with machine learning and the possibilities for improving upon previously known and used methods, can serve many future predictions and benefits. This paper proposes a methodology to feature sorting in the context of supervised machine learning algorithms. A new perspective on machine learning by using it to predict outcomes with a database of the popular moba game Dota2, which consists of a large volume of data that was collected and analyzed. The reported results are concerned with three machine learning models with two significant metrics such as F-measure and Accuracy.

1 Introduction

Sports have been around us for as long as we can remember, with the documented history going back thousands of years. Over the time has remained but has evolved so that new modalities are applied or acquired to improve the sport, has become an important part of today's cultural society. There have been meetings for people to compare their skills and encourage healthy competitiveness, but many raise the eternal debate of what exactly is a sport, what it embodies and what it does not. For some, sporting is an activity involving physical exertion and competition, for others the latter suffices.

With the technological advances of the last decade and the impact they have on our activities today, every year it becomes more accessible to better technology. Thanks to machine learning we can do research in different areas, such as image analysis, weather prediction, facial recognition, among others. Every day we see technology in all our activities, from transportation to communication. In relation to e-sports and prediction other methods are still preferred, however, some previous research has been done and has generated good results.

Further level has been little applied in the world of video games, and pattern recognition [2] has a positive impact on the development of strategic analysis to make decisions to achieve the objectives in the world of video games.

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 567–574, 2023.
https://doi.org/10.1007/978-3-031-18050-7_55

This paper applies feature selection procedures for classification to obtain a reduced feature space that is fed into the data mining [18] stage to build a decision model. In particular, the performance of the original data and the proposed reduction are analyzed in order to shed light on the the recommendation of whether or not to perform a prior feature selection method with certain classifiers that differ in the way they represent knowledge.

The rest of this paper is as follows. Section 2 reviews the data preprocessing methods, concepts about feature selection and presents the motivation of this work. Section 3 describes the proposal and experimental design. Then, Sect. 4 and show the results. Lastly, Sect. 5 raises the discussion and further research lines.

2 Attribute Selection

Attribute selection, also widely known as feature selection, is a process which selects those attributes that are most relevant to explain the model [7]. Plainly speaking, attribute selection is an important step before any data mining activities, because this step distinguishes the valuable attributes and useless ones in the dataset, and thereby extract only meaningful data while leaving out the redundant part [11]. There are many ways to categorize between attribute selection methods, such as supervised and unsupervised [5], categories of attribute ranking [10], attribute subset selection and extended attribute subset selection [16], etc. In this manuscript, we categorize the Attribute selection methods based on the relationship with the classifier model, which presents us with the following 4 categories: filter, embedded, wrapper [4], and hybrid. On the one hand, we have the filter methods, which do not take into account the induction algorithm as model evaluation, but use correlation or consistency as metrics, among others. Simply speaking, filter method is selecting subsets based on a user-specific variance threshold. On the other hand, wrapper methods use the algorithms as evaluation metric. For example the classic Sequential Forward Selection (SFS) [9] which selects the features iteratively and repeatedly according to the optimal performance of the classifier in each iteration. A famous example of embedded method is the LASSO [19]. Embedded method is known to execute the selection process and training process simultaneously, namely to embed a search procedure into the learning process [20]. And the hybrid method [1] is essentially absorbing the advantages from both wrapper and filter methods, which with its continuous nature, it minimizes the problem of high variance. A taxonomy is presented in Fig. 1.

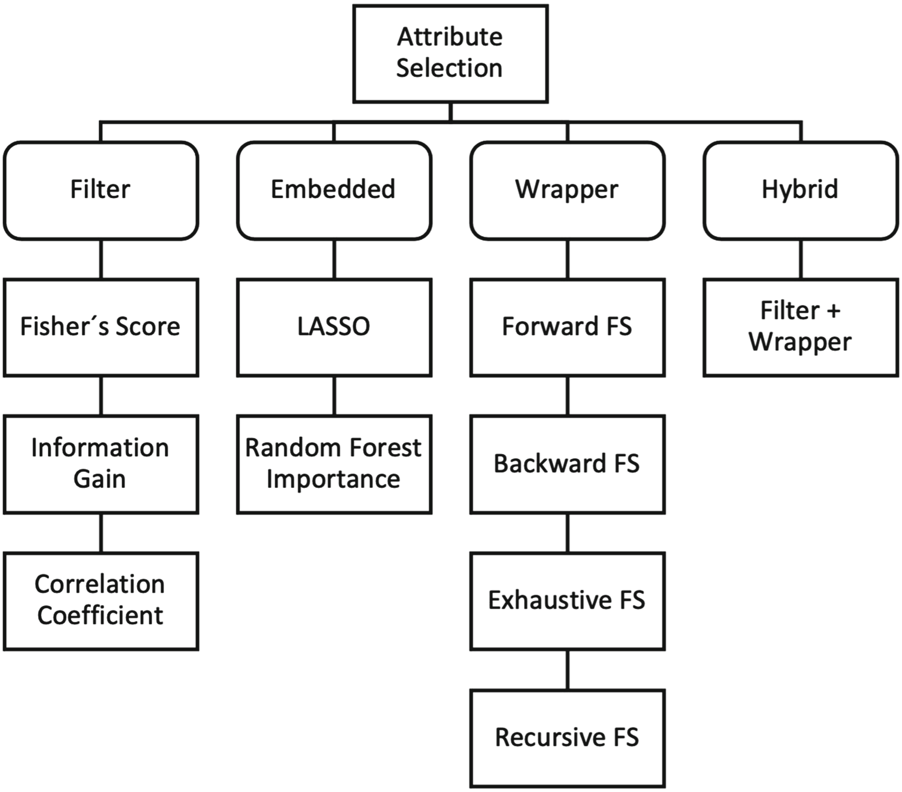


Fig. 1. Taxonomy of attribute selection

The objective is to gain descriptive capacity of the model by reducing the number of irrelevant or redundant attributes but their starting point is different. Focusing on the feature subset selection [8] methods, the most important ones are Correlation-based Feature Selection (CFS) [3] and Consistency-based feature selection (CNS) [17]. For this research we have decided to implement both, since they start from different points, have few parameters and have given a good performance within the supervised [13] machine learning area. CFS is based on attribute-class and attribute-attribute correlation. This algorithm [6] selects a group of features that best explain the class, but, at the same time, these features must have a low correlation with the rest of the attributes to avoid duplicity. In order to apply CFS, a correlation matrix has to be calculated and a heuristic search to find a good subset of features. CNS uses consistency as a basis. For this purpose, it calculates the inconsistency rate, according to which a subset of attributes is inconsistent if there are at least two instances with the same values of its attributes although with different classes. CNS seeks a set of attributes, which according to distance, can be identified as the most explanatory of a class [12].

There are many advantages to these methods, but the disadvantages are the fact that, since they are iterative algorithms, the attributes that are evaluated first are more likely to be among the list of attributes that best explain the model.

3 The Proposed Methodology and Experimental Design

The objective of this paper is to compare the performance of different attribute selection data preprocessing techniques.

The first scenario is Raw, which is the raw data. The second scenario is CFS, resulting from applying Correlation-based feature subset selection. And the third scenario is CNS, resulting from applying Consistency-based feature selection.

The following three classifiers will be applied to the scenarios described above. First, Naive Bayes is a probabilistic machine learning classification algorithm based on the classical Bayes theorem, which assumes that the events are independent. Secondly, C4.5 [14], a classification algorithm through decision trees, using information entropy. And finally, Logistic Regression, uses a multinomial logistic regression model to predict the classes. And the following metrics will be used for comparing the different algorithms:

- F-measure or F-score: is a balanced mean between precision and recall. Precision is a ratio between, true positive and the total amount of predicted positive, and recall is a ratio between, true positive and the total amount of real positive.
- Accuracy [15]: is one metric for evaluating classification models. It is calculated dividing the total of correct predictions by the total number of predictions. It is one of the most widely used metrics, but its disadvantage is that it does not perform well when classes are unbalanced.
- Kappa statistics: Cohen's Kappa, is a measure of concordance, which compares the observed concordance in a set of data with that which could occur by chance alone.

The objective is to evaluate if these attribute selection techniques improve the classification capacity or not. The procedure for this study is as follows:

(a) First, the data set is splitted into training and test sets; (b) second, we apply the attribute selection methods (CFS and CNS) to the training set; (c) third, we used the above classification algorithms for the following three scenarios: raw training set, training set processed with CFS and training set processed with CNS; and (d) fourth, we compare whether the application of attribute selection methods indeed improves the classification performance, or at least has the same train classification performance than the raw training set.

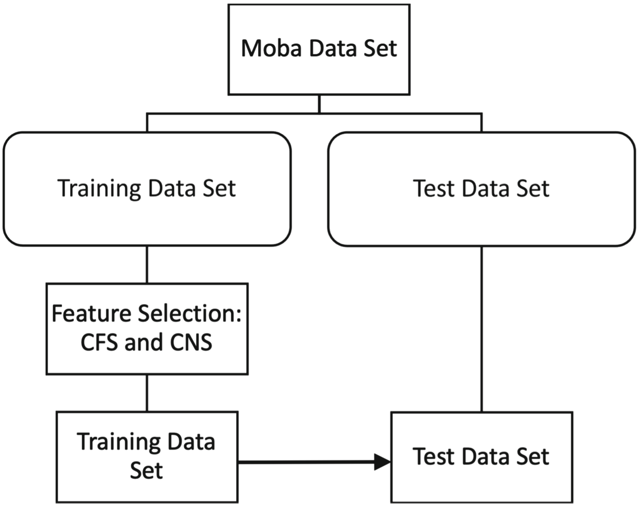


Fig. 2. Proposed methodology: Feature selection

Table 1 provides information about the data on a sample of MOBA games.

Table 1. Distribution of instances in classes: a sample of MOBA games

	Raw	Classification subset	Classification consistency
Attribute train	117	31	44
Attribute test	117	117	117
Class	2	2	2
Train instances	46325	46325	46325
Test instances	10294	10294	10294
Total instances	56619	56619	56619

4 Results

The test results of the experiments are shown in the following tables, each table corresponding to the three metrics used in our study. Tables show the results of the application of the classification algorithms to the three scenarios described above. The results are shown in columns for every classifier.

Table 2 depicts the results in three scenarios: Raw one, CFS and CNS, in the first, second and third rows, respectively. For the other classifiers, the proposal is able to reduce the feature space but the accuracy results have slightly decreased. Concerning the accuracy, as represented in Table 3 the conclusion is similar.

Table 2. Test F-measure

	Naive Bayes	C4.5	Logistic regression
Raw	0.594	0.532	0.592
CFS	0.580	0.555	0.580
CNS	0.567	0.533	0.566

Table 3. Test accuracy

	Naive Bayes	C4.5	Logistic regression
Raw	59.597	53.225	59.413
CFS	58.393	55.877	58.364
CNS	57.149	53.380	57.081

Table 2 shows the results of F-measure, we can see how in general the results of the application of the different classifiers to the CFS and CNS scenarios are practically identical to the results of the Raw scenario. Even in the CFS scenario, applying the C4.5 classifier improves the raw scenario. Therefore, with fewer attributes, results close to the full data treatment are achieved. Also note that the CFS scenario performs better than CNS in all cases.

In Table 3, we see the Accuracy table results as in the F-measure table, the results are equivalent. The raw scenario is the best performance with the exception of CFS and CNS using the C4.5 classifier.

Finally, Table 4 shows Cohen's kappa metric, it continues the same pattern as in the previous metrics the C4.5 classifier performed better for the CFS and CNS scenarios than the Raw scenario.

Table 4. Test Cohen's kappa

	Naive Bayes	C4.5	Logistic regression
Raw	0.182	0.058	0.179
CFS	0.156	0.105	0.156
CNS	0.129	0.156	0.128

5 Discussion and Further Research Line

This paper presented an approach to feature selection under the umbrella of supervised machine learning [6] problems in classification. The empirical study comprised a dataset of 56619 patterns, 117 attributes and 3 labels, binary and multiple class data set, with the purpose to verify the best classifier for the dataset. The results reported the accuracy in the test set measured as the percentage of the number of hits. The computational time is negligible both in the data preparation phase, as well as the data mining stage. There has been continuous research effort for optimizing the process of attribute selection for various different objectives, datasets and scenarios. We presented in this paper that for the same dataset, different preprocessing methods matching with different classifiers produces various performances and results. Knowing the capabilities and characteristics of the attribute selection methods and also their compatibility with different classifiers would be valuable information for future research of improving classification performance.

Acknowledgements. This work has been partially subsidised by the project US-1263341 (*Junta de Andalucía) and FEDER funds.*

References

1. Afef Ben Brahim and Mohamed Limam: A hybrid feature selection method based on instance learning and cooperative subset search. Pattern Recogn. Lett. **69**, 28–34 (2016)
2. Cui, L., Bai, L., Wang, Y., Philip, S.Y., Hancock, E.R.: Fused lasso for feature selection using structural information. Pattern Recogn. **119**, 108058 (2021)
3. Doshi, M.: Correlation based feature selection (CFS) technique to predict student performance. Int. J. Comput. Netw. Commun. **6**(3), 197 (2014)
4. Elavarasan, D., Vincent, D.R., Kathiravan Srinivasan, P.M., Chang, C.-Y.: A hybrid CFS filter and RF-RFE wrapper-based feature extraction for enhanced agricultural crop yield prediction modeling. Agriculture **10**(9), 400 (2020)
5. Huang, S.H.: Supervised feature selection: a tutorial. Artif. Intell. Res. **4**(2), 22–37 (2015)
6. Jordan, M.I., Mitchell, T.M.: Machine learning: trends, perspectives, and prospects. Science **349**(6245), 255–260 (2015)
7. Langley, P.: Selection of relevant features in machine learning: defense technical information center. In: Institute for the Study of Learning and Expertise (1994)
8. Liu, W., Liu, S., Gu, Q., Chen, X., Chen, D.: FECS: a cluster based feature selection method for software fault prediction with noises. In: 2015 IEEE 39th Annual Computer Software and Applications Conference, vol. 2, pp. 276–281. IEEE (2015)
9. Nakariyakul, S., Casasent, D.P.: An improvement on floating search algorithms for feature subset selection. Pattern Recogn. **42**(9), 1932–1940 (2009)
10. Narendra, P.M., Fukunaga, K.: A branch and bound algorithm for feature subset selection. IEEE Trans. Comput. **26**(09), 917–922 (1977)
11. Olafsson, S., Li, X., Shuning, W.: Operations research and data mining. Eur. J. Oper. Res. **187**(3), 1429–1448 (2008)
12. Onan, A.: A fuzzy-rough nearest neighbor classifier combined with consistency-based subset evaluation and instance selection for automated diagnosis of breast cancer. Expert Syst. Appl. **42**(20), 6844–6852 (2015)
13. Quinlan, J.R.: Induction of decision trees. Mach. Learn. **1**(1), 81–106 (1986)
14. Ross Quinlan, J.: C4. 5: Programs For Machine Learning. Elsevier, Amsterdam (2014)
15. Salguero, A.G., Medina, J., Delatorre, P., Espinilla, M.: Methodology for improving classification accuracy using ontologies: application in the recognition of activities of daily living. J. Ambient. Intell. Humaniz. Comput. **10**(6), 2125–2142 (2019)

16. Tallón-Ballesteros, A.J., Cavique, L., Fong, S.: Addressing low dimensionality feature subset selection: ReliefF(-k) or extended correlation-based feature selection(eCFS)? In: Martínez Álvarez, F., Troncoso Lora, A., Sáez Muñoz, J.A., Quintián, H., Corchado, E. (eds.) SOCO 2019. AISC, vol. 950, pp. 251–260. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-20055-8_24
17. Tallón-Ballesteros, A.J., Ibiza-Granados, A.: Simplifying pattern recognition problems via a scatter search algorithm. Int. J. Comput. Methods Eng. Sci. Mech. **17**(5-6), 315–321 (2016)
18. Tan, P.-N.: Introduction to Data Mining. Pearson Education India (2018)
19. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. R. Stat. Soc. Ser. B (Methodol.) **58**(1), 267–288 (1996)
20. Wang, S., Tang, J., Liu, H.: Embedded unsupervised feature selection. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 29 (2015)

Feature Ranking for Feature Sorting and Feature Selection, and Feature Sorting: FR4(FSoFS)∧FSo

Antonio J. Tallón-Ballesteros[1](✉), Alba Márquez-Rodríguez[2], Yaoyang Wu[3], Paola Santana-Morales[2], and Simon Fong[3]

[1] Department of Electronic, Computer Systems and Automation Engineering, University of Huelva, Huelva, Spain
antonio.tallon.diesia@zimbra.uhu.es

[2] University of Huelva, Huelva, Spain

[3] Department of Computer and Information Science, University of Macau, Taipa, Macau SAR, China

Abstract. This paper introduces the application of feature ranking with a twofold purpose: first it achieves a feature sorting which becomes into a feature selection procedure given that a threshold is defined to only retain features with a positive influence with the class label (feature sorting and feature selection: FSoFS), second the feature subset is optimised using feature ranking to get a promising order of attributes (FSo). The supporting paper introduced a single stage: Feature ranking for feature sorting and feature selection, FR4(FS)^2 with the shortcoming that for some high-dimensional classification problems the pre-processed data set did not obtain a more accurate classification model than the raw classifier. The results mean that it deserves to consider feature ranking in a couple of sequential stages with different goals: i) feature sorting, accompanied by feature selection, and ii) also alone

1 Introduction

Data mining (DM) and Analytics may take advantage of including any data preparation procedure from the feature perspective. DM is often used to analyze and extract useful patterns and information from datasets for prediction and decision making. The stage of data pre-processing in Machine Learning can also be demanding in the cost of any data-driven model [1]. It is an essential part that have to be done before the creation of any model and prediction.

Depending on different dataset requirements, machine learning techniques can be categorized as the taxonomy in Fig. 1 shows [15]. Supervised learning requires large amount of labelled data as input and also as a method of evaluating model accuracy when output is produced, hence the term 'supervised'. Unsupervised learning on the other hand refers to the model's manipulation of unllabeled data including clustering and reducing dimension, therefore discovering hidden patterns without human intervention, hence the term 'unsupervised'.

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 575–583, 2023.
https://doi.org/10.1007/978-3-031-18050-7_56

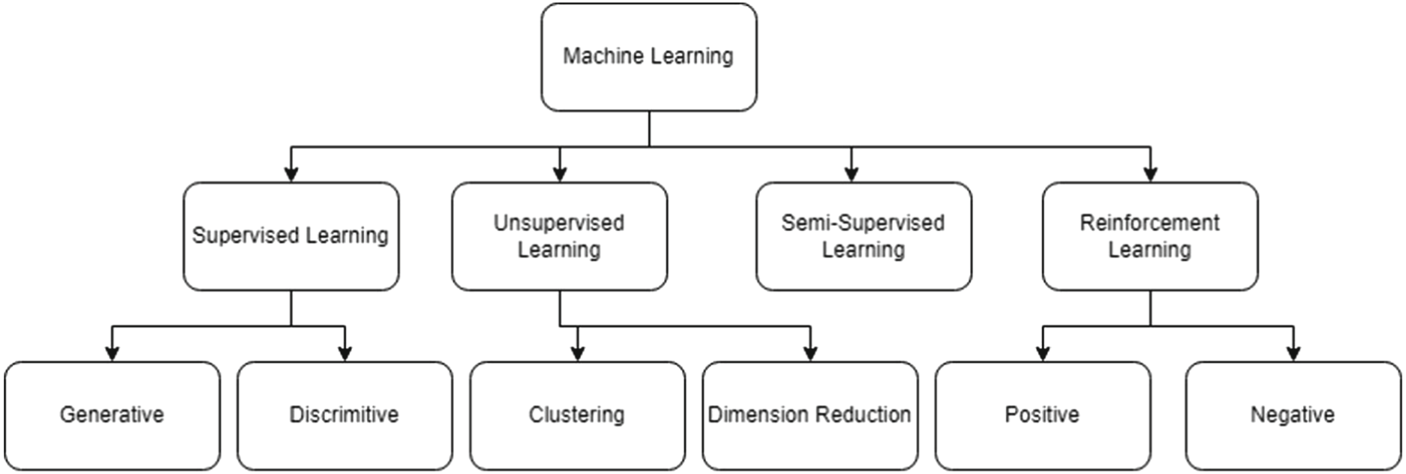


Fig. 1. Taxonomy of Machine Learning techniques

Semi-supervised learning is a combination method absorbing advantages from both supervised and unsupervised learning, and reinforcement learning is a reward-oriented method which reinforces agents takes optimal actions to achieve maximum reward.

There is a great deal of data preparation techniques facing many different issues like the quite high-volume in terms of instances, features, outliers or even missing values [6]. Data can be treated using any of the plethora of available methodologies, although there are only a few which encompass the data preparation stage such as Data Mining Methodology for Engineering applications (DMME) [3]. From all the possibilities, DMME is the most up-to-date approach and hence Society 5.0 requires any almost mandatory prior data-preprocessing before dreaming the modelling of any application with the prediction goal [28]. Evolutionary Computation [29] is a very convenient approximate path to create a high quality solution in time [18]. A myriad of strategies are able to fine-tune the parameters of any initial data transformation such as integer programming [7] or grid search [8] concerning the historical data which may be interpolated to unknown instances. Wide learning [2] is a very novel approach for the machine learning tasks.

This paper introduces a methodology which applies feature ranking not only for feature sorting and feature selection but also for an additional step of feature sorting acting like an optimiser of the feature space arrangement. Particularly, the performance of optimised percentage is analyzed with certain classifiers which differ in the way to represent the knowledge. The remaining of this paper is organized as follows. Section 2 reviews a few concepts about feature selection. Section 3 describes the proposed approach. Then, Sect. 4 depicts the experimental process, reports and discusses the results. Lastly, Sect. 5 draws some conclusions.

2 Feature Selection

Feature Selection (FS) aims at building simpler and more comprehensible models, improving data mining performance, and preparing clean, understandable data [10]. Less is more is a classical statement in the FS context [11]. Computational intelligence includes different categories such as Neural Networks and Genetic Algorithms. FS algorithms may be divided into stochastic feature selection and non-stochastic feature selection [23]. One way to get a stochastic FS approach is adding a genetic or evolutionary component into the FS procedure Nowadays, it is very common to listen that Deep Learning does automatically do FS, although the idea in this scenario is to do a Feature Extraction process which may generate any new feature whereas in Feature Selection the idea is to choose a subset feature space. Clearly, Neural Networks do not include any Dimensionality Reduction and filters have continued being used to retain only relevant attributes [27]. Additionally, in the last decade of previous century a work by L. Milne tried to measure the contribution of any feature for the neural network-based machine learning prediction model [13]. Important contributions

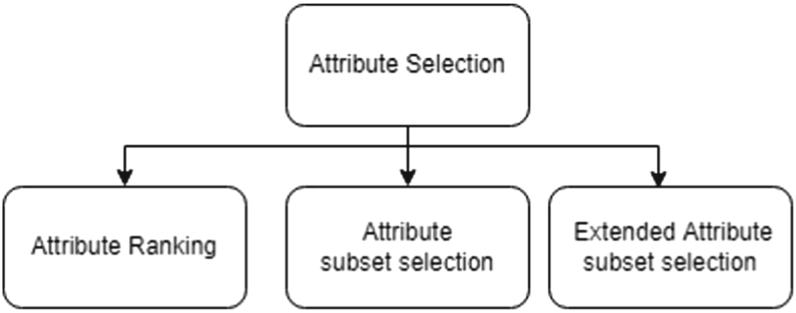

Attribute Selection
Attribute Ranking
Attribute subset selection
Extended Attribute subset selection

Fig. 2. Main attribute selection approaches

on surrogate models for FS have been done in the previous year [5]. Previously, stochastic approaches using Particle Swarm Optimisation has been proposed jointly with surrogate decision-making learners [14]. There are many points of views to characterise the FS algorithms [10]: supervision perspective (supervised, semi-supervised and unsupervised FS) and selection strategy perspective (filter, embedded, semi-wrapper and wrapper methods [10,21]). Depending on the final solution of a FS procedure, attribute ranking, attribute subset selection and extended attribute subset selection approaches may be done as represented in Fig. 2. Attribute ranking requires a way to determine a cut-off (CO) if the objective is to reduce the feature space; if not the outcome is an ordering of the feature space which has been called feature sorting [25]. Attribute subset selection obtains a subset of features and hence there is no need to determine a CO. There are two ways to determine the CO: setting a numerical value for it or using a symbolic value in terms of a percentage of retained features ([19] and [12]) or a nominal value for it (ReliefF(-k)) [22]. Extended attribute subset selection applies a FS procedure based on subsets (FSbS) and then on the non-selected feature is conducted a further FSbS stage [22].

3 The Proposed Methodology

Dimensionality Reduction (DR) is one of the major tasks in data preprocessing. DR literally means to reduce the dimensions of the given data. This approach can help to cleanse the data from redundant or misleading information, as well as minimise complexity, improve processing efficiency and reduce computational cost of the ML process. Feature Selection (FS) is one of the possible approaches to reduce the dimensionality of the data.

Santana Morales et al. [19] introduced an approach called Feature Ranking for Feature Sorting and Feature Selection: FR4(FS)^2 from which this contribution extends. There, the presented framework kept in the space of features only the potentially beneficial ones selected by GainRatio (GR) [17] or InfoGain in an independent way with the primary hypothesis that a 90% of the features with an individual positive incidence may leverage to simplify the model without degrading the classification performance. This contribution does a further step and tries to optimise the characteristic space given that the order of the attributes may play an important role in classification.

According to all the above-mentioned the new proposal has been called Feature ranking for feature sorting and feature selection, and feature sorting: FR4(FSoFS)∧FSo. The words feature sorting are related to a methodology proposed three years ago in HAIS 2019 [25].

4 Experimentation, Results and Discussion

Table 1 describes the classification test-bed for the experiments. As cross validation technique, the hold-out with stratification has been chosen with 75% and 25% for the training and testing sets, respectively. All the attributes are numeric and the instances' distribution in classes is well-balanced and there is no need to apply any kind of data rebalancing [20] prior to the data pre-processing stage [9].

Table 1. Description of the binary classification data sets

Database	# Instances	# Attributes
Arcene	200	10000
Dexter	600	20000
Average	400	15000

GainRatio and InfoGain have been used as data pre-processing techniques under the umbrella of vertical data selection [24] in the first stage and ReliefF in the second stage.

Classification methods are strategies used to predict the response variable of a database. In the experiments two different classifiers have been used: C4.5 and PART under the Weka Framework, version 3.8.0 [4], which is suitable tool for Big Data [16]. As implementation of the C4.5 algorithm proposed by Quinlan the J48 classifier has been applied. It is a classifier based on decision trees [19]. PART is a decision making approach which follows a rule-based system. To evaluate those classifiers and its effectiveness on the different databases two metrics have been used: test accuracy and ROC Area. The first of these, represents the percentage of observations that are correctly classified as its name suggests. The second metric, ROC Area, represents the degree or measure of separability. It is a value between 0 and 1 and the closer it is to 1 the better our model is. Concerning the hyper-parameter adjustment we did a preliminary experimental design using a 5-fold cross validation; for J48 the tuned parameters -as in [26]- where Confidence factor (C) and Minimum number of instances per leaf (M), for that the possible values or range was 0.150, 0.175, 0.200, 0.225 and 0.250 and 2–10, respectively and according to the average accuracy the best values were 0.250 and 2 for C and

M. For PART classifier, the hyper-parameters where Binary Splits (S), Pruned (U) and numFolds (F) with grid-search value of [True|False] for the two first ones and 2–5 for the last one and afterwards the best values where False, False and 3, respectively.

Table 2 reports the test results for Arcene problem. For GainRatio, the proposal is convenient given that some improvements take place. The situation for InfoGain is different since only in the case of PART the results remain unaltered. The proposed approach in conjunction with PART is a good tandem. For C4.5 in one out of the two shown scenarios any enhancement happen.

The test results for Dexter database have been represented in Table 3. Now, GainRatio is able to improve the classification ability along with C4.5. On the other way round, InfoGain takes advantage only of the proposal application with PART for accuracy metric. Globally, bearing in mind both datasets the number of Wins/Ties/Losses with the new proposal is 4/0/3 which represents that the new approach may be interesting. Concerning the pre-processing computational times, it is important to remark that ReliefF is slower than InfoGain and ReliefF, although, in any case, no more than one hour is required for the second stage whereas for the first stage about one minute is required. ReliefF tries to get a more convenient arrangement in order to predict with unseen data more accurately.

Table 2. Test results for Arcene's dataset

Dataset	Data preprocessing approach	Accuracy		AUC	
		C4.5	PART	C4.5	PART
Arcene	GainRatio	76	72	0.513	0.4373
	GainRatio+ReliefF	78	74	0.5557	0.48
	InfoGain	76	80	0.513	0.5902
	InfoGain+ReliefF	72	80	0.4318	0.5902

Table 3. Test results for Dexter's dataset

Dataset	Data preprocessing approach	Accuracy		AUC	
		C4.5	PART	C4.5	PART
Dexter	GainRatio	85.3333	86.6667	0.7067	0.7333
	GainRatio+ReliefF	87.3333	85.3333	0.7467	0.7067
	InfoGain	88	85.3333	0.76	0.7067
	InfoGain+ReliefF	87.3333	86	0.7467	0.7067

5 Conclusions

This paper introduced a new approach to feature selection upon filter-based feature ranking which optimises the feature sub-space arrangement obtained in the first stage which runs feature ranking with a twofold purpose: a) feature sorting, which is constrained to features with a tentatively positive incidence, and hence feature selection and b) feature sorting. For C4.5 classifier, the tandem GainRatio+ReliefF is outstanding. The case of PART is different since there is not a clear approach given that in some situations and for certain metrics GainRatio or InfoGain followed by ReliefF may get better results. It may be though that the GainRatio order may be optimised with ReliefF although the arrangement comparison between InfoGain and ReliefF is more divergent. As further work, it is planned to assess the problem in multi-class classification problems and real-word datasets. Anyway, the done experiments are concerned with some challenging tasks from NIPS 2003.

Acknowledgements. This work has been partially subsidised by the project US-1263341 (*Junta de Andalucía) and FEDER funds.*

References

1. Ayele, W.Y.: A toolbox for idea generation and evaluation: machine learning, data-driven, and contest-driven approaches to support idea generation. arXiv preprint arXiv:2205.09840 (2022)
2. Cardoso, R.P., Hart, E., Kurka, D.B., Pitt, J.: WILDA: Wide Learning of Diverse Architectures for Classification of Large Datasets. In: Castillo, P.A., Jiménez Laredo, J.L. (eds.) EvoApplications 2021. LNCS, vol. 12694, pp. 649–664. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-72699-7_41
3. Huber, S., Wiemer, H., Schneider, D., Ihlenfeldt, S.: Dmme: data mining methodology for engineering applications-a holistic extension to the crisp-dm model. Procedia CIRP **79**, 403–408 (2019)
4. Janošcová, R.: Mining big data in weka. In: 11th IWKM, Bratislava, Slovakia (2016)
5. Jiang, Z., Zhang, Y., Wang, J.: A multi-surrogate-assisted dual-layer ensemble feature selection algorithm. Appl. Soft Comput. **110**, 107625 (2021)
6. Kuhn, M., Johnson, K.: Data Pre-processing. In: Applied Predictive Modeling, pp. 27–59. Springer, New York (2013). https://doi.org/10.1007/978-1-4614-6849-3_3
7. Lapucci, M., Pucci, D.: Mixed-integer quadratic programming reformulations of multi-task learning models. Math. Eng. **5**(1), 1–16 (2023)
8. Kee, G., et al.: Smart robust feature selection (soft) for imbalanced and heterogeneous data. Knowl.-Based Syst. **236**, 107197 (2022)
9. Li, J., Yaoyang, W., Fong, S., Tallón-Ballesteros, A.J., Yang, X.-S., Mohammed, S., Feng, W.: A binary pso-based ensemble under-sampling model for rebalancing imbalanced training data. J. Supercomput. **78**(5), 7428–7463 (2022)
10. Li, J., et al.: Feature selection: a data perspective. ACM Comput. Surv. (CSUR) **50**(6), 1–45 (2017)
11. Liu, H., Motoda, H. Computational methods of feature selection. CRC Press (2007)
12. Merchán, A.F., Márquez-Rodríguez, A., Santana-Morales, P., Tallón-Ballesteros, A.J.: Feature ranking merging: Frmgg. application in high dimensionality binary classification problems. In: Mathur, G., Bundele, M., Tripathi, A., Paprzycki M. (eds.) ICAIAA 2022. AIS. Springer (2023)

13. Milne. L.: Feature selection using neural networks with contribution measures. In: Yao, X. (ed.) AI '95 CONFERENCE-, pp. 571–571. World Scientific (1995)
14. Nguyen, H.B., Xue, B., Andreae, P.: Pso with surrogate models for feature selection: static and dynamic clustering-based methods. Memetic Comput. **10**(3), 291–300 (2018)
15. Nicolas, P.R.: Scala for machine learning. Packt Publishing Ltd. (2015)
16. Pramanik, P.K.D., Pal, S., Mukhopadhyay, M., Singh, S.P.: Big data classification: techniques and tools. In: Applications of Big Data in Healthcare, pp. 1–43. Elsevier (2021)
17. Quinlan, J.R.: Induction of decision trees. pp. 81–106 (1986)
18. Rodríguez-Molina, A., et al.: Exploiting evolutionary computation techniques for service industries. In: Evolutionary Computation with Intelligent Systems, pp. 153–179. CRC Press
19. Santana-Morales, P., Merchán, A.F., Márquez-Rodríguez, A., Tallón-Ballesteros, A.J.: Feature ranking for feature sorting and feature selection: Fr4(fs)^2. In: de Vicente, J.M.F., Álvarez Sánchez, J.R., de la Paz López, F., Adeli, H. (eds.) Bio-inspired Systems and Applications: from Robotics to Ambient Intelligence - 9th International Work-Conference on the Interplay Between Natural and Artificial Computation, IWINAC 2022. LNCS, Part II, vol. 13259, pp. 545–550. Springer, Cham (2022)
20. Slowik, A., Bottou, L.: On distributionally robust optimization and data rebalancing. In: Camps-Valls, G., Ruiz, F.J.R., Valera, I. (eds.) International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28–30 March 2022, Virtual Event. Proceedings of Machine Learning Research, vol. 151, pp. 1283–1297. PMLR (2022)
21. Tallón-Ballesteros, A.J., Riquelme, J.C., Ruiz, R.: Semi-wrapper feature subset selector for feed-forward neural networks: applications to binary and multi-class classification problems. Neurocomputing **353**, 28–44 (2019)
22. Tallón-Ballesteros, A.J., Cavique, L., Fong, S.: Addressing low dimensionality feature subset selection: ReliefF(-k) or extended correlation-based feature selection(eCFS)? In: Martínez Álvarez, F., Troncoso Lora, A., Sáez Muñoz, J.A., Quintián, H., Corchado, E. (eds.) SOCO 2019. AISC, vol. 950, pp. 251–260. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-20055-8_24
23. Tallón-Ballesteros, A.J., Correia, L., Cho, S.-B.: Stochastic and non-stochastic feature selection. In: Yin, H., Gao, Y., Chen, S., Wen, Y., Cai, G., Gu, T., Du, J., Tallón-Ballesteros, A.J., Zhang, M. (eds.) IDEAL 2017. LNCS, vol. 10585, pp. 592–598. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68935-7_64
24. Tallón-Ballesteros, A.J., Correia, L., Leal-Díaz, R.: Attribute Subset Selection for Image Recognition. Random Forest Under Assessment. In: Sanjurjo González, H., Pastor López, I., García Bringas, P., Quintián, H., Corchado, E. (eds.) SOCO 2021. AISC, vol. 1401, pp. 821–827. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-87869-6_78
25. Tallón-Ballesteros, A.J., Fong, S., Leal-Díaz, R.: Does the order of attributes play an important role in classification? In: Pérez García, H., Sánchez González, L., Castejón Limas, M., Quintián Pardo, H., Corchado Rodríguez, E. (eds.) HAIS 2019. LNCS (LNAI), vol. 11734, pp. 370–380. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29859-3_32
26. Tallón-Ballesteros, A.J., Riquelme, J.C.: Data mining methods applied to a digital forensics task for supervised machine learning. In: Muda, A.K., Choo, Y.-H.,

Abraham, A., N. Srihari, S. (eds.) Computational Intelligence in Digital Forensics: Forensic Investigation and Applications. SCI, vol. 555, pp. 413–428. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-05885-6_17

27. Tallón-Ballesteros, A.J., Riquelme, J.C., Ruiz, R.: Filter-based feature selection in the context of evolutionary neural networks in supervised machine learning. Pattern Anal. Appl. **23**(1), 467–491 (2019). https://doi.org/10.1007/s10044-019-00798-z
28. Wahab, S.A.: Moving towards society 5.0: A bibliometric and visualization analysis. In SOCIETY 5.0: First International Conference, Society 5.0 2021, Virtual Event, p. 93. Springer Nature (2022)
29. Zhan, Z.-H., Shi, L., Tan, K.C., Zhang, J.: A survey on evolutionary computation for complex continuous optimization. Artif. Intell. Rev. **55**(1), 59–110 (2021). https://doi.org/10.1007/s10462-021-10042-y

Special Session on Tackling Real World Problems with Artificial Intelligence

Introducing Intelligence to the Semantic Analysis of Canadian Maritime Case Law: Case Based Reasoning Approach

Bola Abimbola[1], Qing Tan[2], and José Ramón Villar[1](✉)

[1] University of Oviedo, Oviedo, Spain
{U0285018,villarjose}@uniovi.es
[2] Athabasca University, Athabasca, Canada
qingt@athabascau.ca

Abstract. The use of machine learning and semantic analysis in case law is the new trend in modern society. Case Based Reasoning tools are being used to analyze texts in courts to make and predict judicial decisions which are designed to base the outcomes of current court proceedings from past and or learning from the mistakes to make better decisions. Because of the accuracy and speed of this technology, researchers in the justice system have introduced Machine Learning to optimize the Case-Based Researching approach. This paper presents a study aimed to critically analyze semantic analysis in the context of machine learning and proposes a case-based reasoning information retrieval system. It will explore how CBR-IR is being used to improve legal case law information retrieval. The study covers the importance of semantic analysis. The study will discuss limitations and recommendations for improvement and future research. The study recommends that it is necessary to conduct further research in semantic analysis and how they can be used to improve information retrieval of Canadian maritime case law.

1 Introduction

Intelligence Information retrieval refers to the process of accessing case documents that are related to a given case. The process is important as it helps in linking related information and thus ensuring ideal solutions are met in every case. There are several approaches that are used in retrieving these data. For instance, Information extraction which uses natural language processing, machine learning techniques, rule-based, and case-based approach. However, to deliver the much-needed value, it is critical to ensure that all case and court processes are efficient, effective, and equitable. These approaches are used to retrieve the previous case with respect to the current case. Machine learning, semantic learning, and computers favor state legislatures and all other courts in the land by allowing all the stakeholders to design and develop equitable, effective, and efficient legal systems to serve justice for all [1]. Therefore, the purpose of this study is to identify and evaluate how machine learning and semantic learning integration into the legal domain is done by focusing on the analysis of legal texts and their impact on the legal system.

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 587–595, 2023.
https://doi.org/10.1007/978-3-031-18050-7_57

2 Problem Statement

One of the major problems for courts in the modern environment is the inability to optimally analyze quantitative legal data. Inadequate data acquisitions by courts today are a major problem that hinders decision making and litigation of cases. With limited information, courts are unable to have consistency between inputs and outputs. Today, courts are unable to normalize inputs and outputs, implying that they end up making uninformed decisions. Inconsistencies between inputs and outputs can be linked to the approaches that are used to gather information. Legal institutions rely on existing data or newly generated data to make predictions. Therefore, it is imperative for courts to ensure that these sources are reliable and gathered from people who are affected and who will benefit from the designed predictions. The presence of different types of data can pose a challenge in data analysis and generation of reliable predictions. The vast amount of data requires courts to constantly rely on similarities and differences in legal data, which can significantly aid generate optimal predictions. Because modern courts operate in a highly dynamic and competitive environment, they have to address the needs of all populations, which prompts them to adopt approaches that can help improve customization [11]. Also, these changes lead to changes in ML algorithms and thus, establishment of input and output inconsistencies.

Legal reasoning (LR) is a key component in legal practice, LR are methods that lawyers use to apply laws to facts to answer legal questions, it is mainly based on data gathered from previous cases. It is safe to imply that the verdict of cases can be made easier or difficult depending on the available legal data [20]. In the research [20], the authors argue that the effectiveness and efficiency of legal processing depends on how data is stored and retrieved. In this context, analysis of legal cases has been a challenge in the past because of the inability to store and retrieve data as needed. In the past, courts relied on hard copies and manual retravel of data, which was time consuming, predictable, and exposed to errors.

In most countries today, legal data is retrieved manually or by the use of syntax [12]. Using these methods to analyze cases have numerous problems, which inhibits the abilities of judges to analyze and identify cases. The use of manual methods and syntax methods are associated with generation of inadequate and less useful information. The problem of effective and efficient data retrieval and storage is significant in the legal domain because it is primarily on information. Besides, information gathered in the legal domain is important because it defines the survivability of society. Because of the inefficiency and ineffectiveness, the justice system is unable to provide justice where it is deserved. The soundness of judicial verdict depends on the quality of information gathered by legal experts. Another dimension of this problem is that judicial verdicts are based on the information gathered from past cases. As time passes, the volume of the information in past cases increases, making it a challenge for legal experts to analyze the data.

3 Related Work

The current literature shows that machine learning, information technology, and semantic learning have been used in the court system to enhance the quality of services provided [9]. For example, machine learning and semantic learning allow text creation, storage, and retrieval, which have become more accessible and an essential part of the legal process [2]. Aside from the hearing capacity, judges need to create composed rulings, judgment, and purposes behind the choices they ceaselessly make [10]. After the typewriter era, judges were forced to write decisions in longhand, and secretaries would then type the same out in typescript [8].

Additionally, machine learning and semantic learning have improved access to the law. In most countries and jurisdictions, the applicable law is found in different sources, including statutes, law reports, and customary laws [3]. In the past legal documents were kept as paper copies stored in filing cabinets and folders, but now it is possible access digital copies over the internet, it is easy to access and use the existing information to make critical decisions.

The use of ML to aid in semantic analysis of texts, phrases, and language of law data has helped in quantitative analysis of legal data [17]. The progress in artificial intelligence, language processing, and machine learning are linked to rapid evolution in algorithms and data-based practices [13]. An evolution in algorithms has helped courts to make informed predictions through provision of important insights and knowledge about [14]. Because of the need to offer accessibility to court cases and decisions, and the need to promote equality, machine learning is used to find similarities and differences in litigation judgements, which improves court outcomes [15].

4 Machine Learning and Sentimental Analysis

Semantic is a branch of linguistics that focuses on exploring the meaning of sentences and words. Understanding words and sentences help understand language. Semantic analysis helps in the comprehension of forms and how they interact with each other. In machine learning, semantic analysis is used to determine the significance of syntax in a program. It is used to verify whether software declarations are correct. It is used to determine if a code in a program is accurate.

Machine learning and semantic analysis are becoming an essential part of the modern court system. One of the easier parts of electronic court innovation is utilizing an advanced camera or scanner to take exhibit and show it on the screen [4]. This underlines how machine learning and semantic learning have revolutionized the modern legal and court systems [5]. The use of advanced cameras and scanners in the legal system shows that machine learning and semantic analysis to retrieve Canadian maritime law cases are improving and changing the Canadian legal and court system. However, future studies should focus on the challenges of adequate application and use of machine learning, information technology, and semantic learning in the court system as the court system moves towards efficiency and effectiveness.

According to [18], employment of semantic retrieval technology will enhance information quality retrieval. The work suggests a technological framework that will help users retrieve information by providing them with relevant case-based and semantic retrieval documents. Users will key in their query terms in their normal language and the system conduct analysis on them. [19] argues that comparing ideological query representation against the conceptual representations database aid in choosing the match within a close range. [19] further contends that users can use natural language or a relevant document or Boolean query to search-related documents.

The suggested sets of concepts for geographical-related information retrieval included a mixture of quantitative and qualitative geometric data, including sparse coordinate and topological relations information representing the geographical place's footprints. Geographical categories classified places and then connected to non-geographical cases grouped by conceptual ranking. According to [20], the goal was to combine Euclidean and hierarchical distance analysis to establish a geographical distance evaluation. Another proposal was on query advancement approach that employed advanced dependent topologies to bring queries nearer to document collection characteristics and user's preferences [20]. It aimed at linking each concepts' case and class with a feature vector to modify these ideas to the terminology and document collection utilized. The concepts and their related feature vectors processed results after identifying user's query on the search engine [20].

According to [21], concepts established in the Spirit web project were employed to support document retrieval that was geographically suitable to users' request, where query advancement used domain and spatial concepts. A concept network deduced from concepts from the original query words served as a knowledge base for modifying query advancement [20]. The conceptual query advancement quality relied on the quality of concept network. The purpose of a concept network was to match original query words which resulted in the development of other concepts and queries terms.

While most concepts categories used the WordNet as a controlled vocabulary to advance the query, the approach suggested by [20] combines the advantages of statistical methods and concept use. For query advancement, field concepts were utilized as controlled terminology. The primary presupposition is that users create a query concurrently illustrating an issue they are trying to address. The CBR approach uses other related cases to address an information retrieval request [22]. A solution is acquired by providing several links related to the user's query. A case-based approach contains a set of data that defines and provide information about other data. The model by [20] offers various intelligent query advancement and processing benefits.

5 Case-Based Reasoning Information Retrieval System (CBR-IR)

Traditional IR systems recovered information without defining any user's specific field of interest. As such, the system provided massive data that was not important to the user. [23] demonstrates how to employ concepts effectively included in various multidisciplinary fields comprising of different terminologies aiming goal to improve the browsing outcome quality for extensive search systems.

One strength of Case-Based Reasoning (CBR) systems is the ability to reason about a problem case and perform highly intelligent problem-solving, such as the generation of legal arguments or detailed operational plans [26].

Query searching using concepts is a promising and unique approach in the retrieval process. Users do not need to know the documents implementation; their focus is on the conceptual searching level. Domain concept is helpful for query advancement by increasing the input terms with the appropriate domain ontologies. WordNet adds meronyms, homonyms, and synonyms to the index terms, making the indexing stage effective [28]. There exist two problems in utilizing the concept–based approach. The first problem is using keywords to extract semantic concepts. Its main problem is determining relevant concepts that identify documents and determine the language used in user queries. It is essential to avoid matching and connecting inappropriate concepts and disregard appropriate concepts. The second problem is document indexing. Field ontology is established, and property and concept relationship in the professional field is described.

Nevertheless, query expansion contains some built-in dangers. Thesaurus has been used in information retrieval to identify the linguistic entities and synonymous expressions semantically the same. A query drift can occur due to query ambiguity providing information that is irrelevant to the user. For instance, the term windows could mean Microsoft Windows OS (operating system) or the actual house windows. The system should employ domain concepts to solve the problem. Not every tokenized term should be set for expansion. Query expansion process should replace the terms in the domain concepts with the original user terms and their related domain concepts.

A concept-based method utilizes concepts from a specific domain and CBR approach with various metadata containing relevant documents defines a case. A case bases act as a piece of document information to examine the query process and retrieve information from appropriate documents in the digital library [24]. It aimed at improving concept-based information retrieval by integrating.

domain ontology, case-based reasoning process, and traditional information retrieval process [24]. The model proposed by [24] employs concepts to expand queries and integrates textual and case-based closeness to recover a set of information for relevant documents to give users several document recommendations options. The steps in Fig. 1 are as follows:

Step one: matching a new case using other cases in the case dB – The CBR-IR system performs a CBR analysis of inputs. It then uses the analyzed results to generate a text-based document retrieval system. [28].

Step two: recover the closest matching case from the past cases' library - Defines categories and classes for a ranking system in this step. This is used to generate a standard query of these texts' top N terms or top D pairs of terms. The CBR-IR module checks large number of cases in the case dB. Full texts of the court's opinions and cases selected from the CBR module's Case dB. CBR system used a similarity measure that is based on a generalized weighted distance metric.

Step three: reuse the recovered case to address the existing issue - the system sorts the retrieved documents into an initial ordering of cases relevant to the problem case. The categorization is done on the bases of "on-point". The model of relevance and on-pointiness used are then used in CBR-IR style system settings. The cases selected are

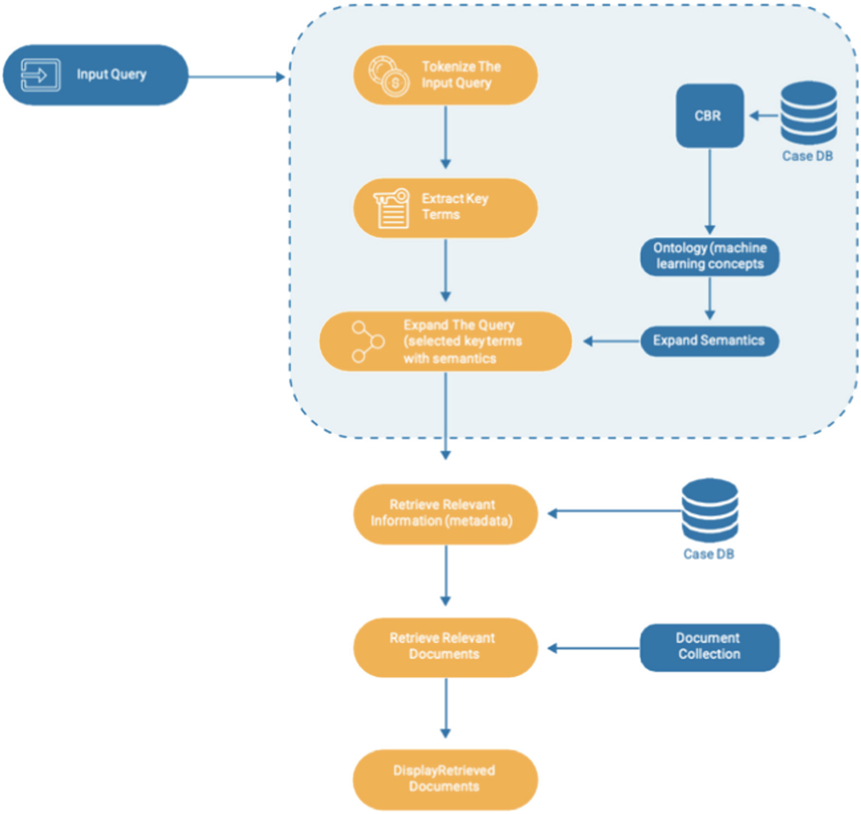

Input Query
Tokenize The Input Query
Extract Key Terms
Expand The Query (selected key terms with semantics
CBR
Case DB
Ontology (machine learning concepts
Expand Semantics
Retrieve Relevant Information (metadata)
Case DB
Retrieve Relevant Documents
Document Collection
DisplayRetrieved Documents

Fig. 1. The conceptual system diagram of the CBR-IR system

passed through the preprocessing steps, it is presented to the “on-point” classifier to retrieve similar previous cases from the CBR-IR. [28].

Step four: re-evaluate and modify the suggested solution if needed - The on-pointiness model sorts the unique cases. The sorting generates a partial ordering in which Case A is more “on-point” than Case B (if the set of applicable dimensions A shares with the problem case are more than those of case B and the problem case. Maximal cases in this ordering are called most on-point cases. The result of sorting these instances is virtualized in a “claim lattice.“ (See Fig. 2 for an example.)

Step five: rename the final result as a new case - These CBR analysis results are represented in a “claim lattice” that uses a hybrid CBR-IR system to select unique cases from the claim lattice. For instance, the most on-point cases and maximal cases in the on-point ordering. The texts associated with the unique cases are passed to a modified version of the relevance feedback (RF) mechanism of the “INQUERY IR” system [27].

Apache Lucene provides all the relevant interfaces for query, index engines and analyses the overall text format content, such as txt, pdf, docx, and web page data. The functions can be easily embedded into various applications to implement full text retrieval functions. Comparing Lucene to other text retrieval system, it provides faster

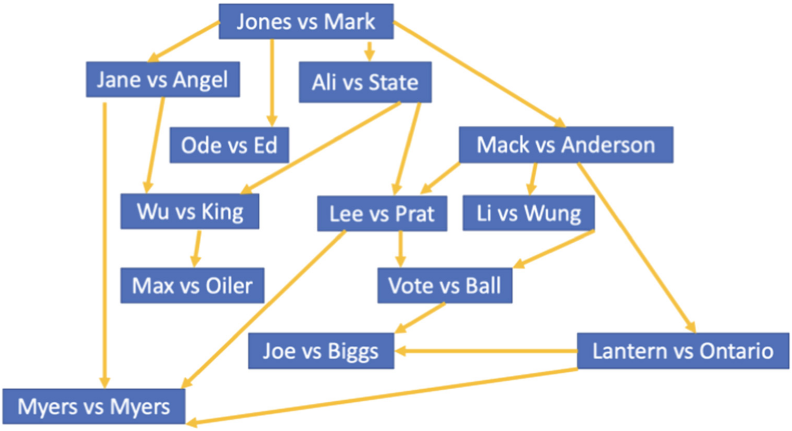


Fig. 2. Claim lattice for Jones vs Mark

speed and better performance repository and data retrieval without degrading the system performance and provides steady indexing across database in data center and files in various formats. [30] Lucene core data is encapsulated into classes by defining a file index format, and then program processes to form a complete information search engine. Lucene's has three main parts: External Interface, Index Core and Infrastructure Package. Index Core is the direct focus when manipulating file indexes [29].

Case similarity evaluation is done for the subject and author attributes [25]. Statistical IR methods used the Apache Lucene to measure Title attribute content. In Lucene, the Boolean technique and vector space model were used to determine the relevancy of a particular document to the user's request.

6 Conclusion

Semantic analysis is an important area of study today because courts today strive to achieve equality and optimal decision making by adopting technologies. The ability to understand information that courts gather is vital because it helps make effective predictions and improve performance. Semantic analysis can be used to address technical challenges that are faced in machine learning. Semantic analysis can be used to address problems associated with limited data, presence of different data types, and inconsistencies in AI predictions and real-life solutions.

The research has demonstrated that the already developed domain-specific ontology with Machine Learning can be efficient for query advancement. Many researchers have used semantic retrieval technology using concepts. It helps solve problems that lack semantics in traditional retrieval technology. Using concepts in ontology enhances search results. Expanding a query aims at minimizing document or query mismatch by adding related phrases and terms to the relevant documents set.

This research has introduced Machine Learning to the semantic analysis module in the proposed Case-Base Reasoning Information Retrieval (CBR-IR) system. The future research will further study and develop the CBR-IR by implementing the machine learning techniques with semantic analysis in the information retrieval system.

Acknowledgement. This research has been funded by the SUDOE Interreg Program -grant INUNDATIO-, by the Spanish Ministry of Economics and Industry, grant PID2020-112726RB-I00, by the Spanish Research Agency (AEI, Spain) under grant agreement RED2018–102312-T (IA-Biomed), and by the Ministry of Science and Innovation under CERVERA Excellence Network project CER-20211003 (IBERUS) and Missions Science and Innovation project MIG-20211008 (INMERBOT). Also, by Principado de Asturias, grant SV-PA-21-AYUD/2021/50994.

References

1. Venkateswarlu Naik, M.: Building a legal expert system for legal reasoning in specific domain-a survey. Int. J. Comput. Sci. Mach. Learn. **4**(5), 175–184 (2012)
2. Mosweu T. L., Mosweu, O.: Electronic court records management systems: a review of literature in selected African countries. Mousaion: South African J. Info. Stud. 36(4) (2019)
3. Pratiwi, S.J., Steven, S., Permatasari, A.D.P.: The application of e-court as an effort to modernize the justice administration in indonesia: challenges & problems. Indonesian J. Adv. Legal Serv. **2**(1), 39–56 (2020)
4. Imamnazarovna, N.: Aspects of legal regulation of electronic document and electronic document circulation in business. Am. J. Political Sci. Law Criminol. **02**(11), 8–14 (2020)
5. Procopiuck, M.: Machine learning and time of judgment in specialized courts: What is the impact of changing from physical to electronic processing? Gov. Inf. Q. **35**(3), 491–501 (2018)
6. Levitt, H.M., Morrill, Z., Collins, K.M., Rizo, J.L.: The methodological integrity of critical qualitative research: Principles to support design and research review. J. Couns. Psychol. **68**(3), 357–370 (2021)
7. Epp, M., Otnes, C. C.: High-quality qualitative research: getting into gear. J. Serv. Res. 109467052096144 (2020)
8. Mészáros, P. E.: The evolution of electronic administration and its practice in judicial proceedings. Pravni vjesnik 34(3–4) (2018)
9. Hasan M. I., Mia, B.: Initiation of virtual court system during COVID-19 pandemic and e-judiciary: challenges and way forward. Daengku: J. Human. Soc. Sci. Innov. 1(1), 8–17 (2021)
10. Gupta, L., Gadiwala, S.: Coping with the Coronavirus Disease-2019 pandemic: a giant leap towards digital transformation in academic research. Indian J. Rheumatol. 16(2) (2021)
11. Tae, K., So, S.: Machine-learning-based deep semantic analysis approach for forecasting new technology convergence. Technol. Forecast. Soc. Change **157**, 120095 (2020)
12. Medvedeva, M., Vols, M., Wieling, M.: Using machine learning to make decisions of the European court of human rights. Artif. Intell. Law. **28**, 237–266 (2019)
13. Huang, X., Zanni-Merk, C., Cremilleux, B.: Enhancing deep learning with semantics: an application to manufacturing time series analysis. Procedia Comput. Sci. **159**, 437–446 (2019)
14. Sarker, H.: Machine learning: algorithms, real-world applications and research directions. SN Comput. Sci. **2**, 160 (2021)
15. Ebietomere, E.P., Ekuobese, G.O.: A semantic retrieval system for case law. Appl. Comput. Syst. **24**(1), 38–48 (2019)

16. Cao, J., Wang, M., Li, Y., Zhang, Q.: Improved support vector machine classification algorithm based on adaptive feature weight updating in the Hadoop cluster environment. PLoS ONE 14(4): e0215136 (2019)
17. Babacar, G., Dezheng, Z., Aziguli, W.: Improvement of support vector machine algorithm in big data background. Math. Probl. Eng. (2021). https://doi.org/10.1155/2021/5594899
18. Baoxian, J.: Application of intelligent information retrieval for big data oriented brain science. Adv. Eng. Res. 66 (2018)
19. Joby, D.: Expedient Information retrieval system for web pages using the natural language modeling. J. Artif. Intell. Cap. Netw. **2**(2), 100–110 (2020)
20. Afuan, L., Ashari, A., Suyanto, Y.: A study: query expansion methods in information retrieval. J. Phys. Conf. Ser. **1367**, 012001 (2019)
21. Tehseen, R.: Semantic Information retrieval: a survey. J. Inf. Technol. Softw. Eng. 08(04) (2018)
22. Lin, K.-S.: A case-based reasoning system for interior design using a new cosine similarity retrieval algorithm. J. Inf. Telecommun. **4**(1), 91–104 (2020)
23. On-Piu Chan, J.: Digital transformation in the era of big data and cloud computing. Int. J. Intell. Inf. Syst. **9**(3), 16 (2020)
24. Anaissi, A., Goyal, M., Catchpoole, D.R., Braytee, A., Kennedy, P.J.: Case-based retrieval framework for gene expression data. Cancer Inform. **14**, 21–31 (2017)
25. Devi, M.U., Gandhi, G.M.: Scalable information retrieval system in semantic web by query expansion and ontological-based LSA ranking similarity measurement. Int. J. Adv. Intell. Paradig. **17**(1/2), 44 (2020)
26. Kolodner, J.: Case-Based Reasoning. Morgan Kaufmann, San Mateo (1993)
27. Daniels, J. J., Rissland E. L.: A case-based approach to intelligent information retrieval. In: SIGIR'1995, Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Seattle, Washington, USA, July 9–13 (1995) (Special Issue of the SIGIR Forum)
28. Feuillâtre, H., et al.: Similarity measures and attribute selection for case-based reasoning in transcatheter aortic valve implantation. PLOS ONE (2020). https://doi.org/10.1371/journal.pone.0238463
29. Jian, W.: Design and implementation of campus network search engine based on Lucene. J. Hunan Inst. Eng. (2012)
30. Youzhuo, Z., Yu, F., Ruifeng, Z., Shuqing, H., Yi, W.: Research on Lucene based full-text query search service for smart distribution system (2020)

Case-Based Reasoning for the Prediction of Flash Flood

Enrique Fernádez[1], José Ramón Villar[1](✉), Alberto Navarro[2], and Javier Sedano[2]

[1] University of Oviedo, 33005 Oviedo, Spain
{uo257742,villarjose}@uniovi.es
[2] Instituto Tecnoógico de Castilla y León, Burgos, Spain
{alberto.navarro,javier.sedano}@itcl.es

Abstract. The term flash-flood refers to the sudden raise in the water levels in a basin due to an abrupt change in the weather conditions. Early detection of flash-floods reduces the harm that they can produce in the infrastructure or even in preventing human losses. Up to now, the studies focus on the dynamics of the basins, determining how the water levels would be in a considered scenario. However, nothing have been done concerning the online prediction of flash-floods. This research focuses on this topic, proposing a Case-Based Reasoning tool to cope with the estimation of the water levels on a basin based on the current basin conditions and the weather forecast. Furthermore, this CBR tool has been designed to work in different basins provided enough data is available, either from past experiences or from simulation. This research is being designed, developed on two real basins, one from Spain and one from France; however, the experimentation has only been addressed with realistic data from the Venero-Claro basin in Spain. Expectancy is that the performance of the CBR tool will perfectly mimic the decision making of the public safety experts.

1 Introduction

Climate change and the abrupt changes in the weather conditions are inducing more and more flash-floods all around the world. As a consequence, there are studies to evaluate the current scenarios to assess the risk of flooding; this is call Risk Assessment. Risk assessment identifies those areas of a basin that are susceptible to flooding with the aim to define efficient management policies [1,15], to design complementary infrastructures in order to mitigate the effects of these flash-floods [13] or to design new infrastructure that avoids the flooding [12].

Nevertheless, there are new aspects that have not received enough attention from the research community yet. One of these is the on-line assessment of flash-floods to aid the public safety experts in the decision making process to tackle these floods in the short time. Solving this latter application, that is, aiding the public safety experts in the flash-flood detection, needs the use of data not only from the basin geo-morphological and hydrological information, but also

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 596–605, 2023.
https://doi.org/10.1007/978-3-031-18050-7_58

information concerning the current scenario -for instance, the water level of the rivers at certain points, the amount of rain in the last relevant period, etc.- together with weather forecast information.

This study propose a solution for this problem; to our knowledge, this is the first study focusing on the on-line flash-flood prediction. The idea is to gather information from the current basin, weather forecast and from the available sensory system; this information is used to estimate the height of water at any point in the basin. The height of water can be superposed to a map of the basin, providing information about what geographical points in the basin are in danger of a sudden flood.

To do so, this study proposes Case-Based Reasoning (CBR) as the intelligent paradigm that will extract the data and information of an scenario and will merge the maps from the different retrieved and relevant cases. The outcome of the CBR is the above-mentioned map containing the height of the water for each single point in the basing. Together with a web application, a novel on-line flash-flood tool can guide the public safety teams in their decision making.

The structure of this research is as follows. Next Section discuss the related work section concerning flash-flood risk assessment. Section 3 describes the first prototype that has been developed, while Sect. 4 shows the results obtained for a real basin in Spain. Finally, the conclusions from this research are drawn.

2 Related Work

The majority of the studies in the literature deal with the concept of risk assessment. Risk assessment identifies those areas of a basin that are susceptible to flooding. In doing so, several different Machine Learning (ML) techniques and Artificial Intelligence methods have been employed. However, numerical approaches have been used as well [14], but always in off-line studies of a basin due to the time needed for the simulations.

Considering unsupervised ML, some studies proposed clustering as an strategy that, given a set of risk indexes, labels certain GIS -Global Information System- position into certain risk group. For instance, [2] propose merged the improved analytic hierarchy process method, and an integration of iterative self-organising data analysis and maximum likelihood (ISO-Maximum) clustering algorithm; with these method, each possible position was assigned with a risk index that reflects its geo-morphological and geographical characteristics.

Supervised techniques, such as Decision Trees or Support Vector Machines, have been employed as well for the risk assessment. In [10] rule-based decision tree and the combination of frequency ratio and logistic regression statistical methods were proposed for flood susceptibility mapping at Kelantan, Malaysia. Firstly, a set of relevant features were calculated from the GIS and remote sensing data; up to 10 conditional factors were proposed. Then, the ML and the combined statistical methods determined the risk of each position in a map. Similar study has been also analysed in [6], comparing Random Forest and Boosted Regression Trees.

Neural networks have been also applied in this risk assessment [4,5]. Again, a set of features were extracted from GIS positions to train and test the neural model. When deploying the model, each GIS position is analysed and the set of features is calculated; afterwards, the neural model estimates the risk of the position. In [11], a similar procedure but considering Support Vector Machines was proposed.

Multi-criteria decision making has been also used in the evaluation of the risk assessment. In [3], the authors proposed Analytical Hierarchy Process to generate the decision making process and to assign a risk label to each GIS position. A similar study has been reported in [7] in the assessment of the Mashhad Plain basin, Iran.

Besides, the Dempster-Shafer based evidential belief function has been proposed for the assessment and spatial prediction of flood-susceptible areas [9]. A set of features is extracted from each GIS location and these features are used as inputs to the Dempster-Shafer method. Interestingly, this probabilistic -based reasoning method overtook other methods used in the literature, such as Logistic Regression and Frequency Ratio. A similar conclusion is presented in [6], where a comparison between the Dempster-Shafer of two different Decision Trees was performed- In this latter case, the features and the method varies from the former studio, but the conclusions are similar.

These studies show that perhaps ML in not the most interesting technique for this type of problems because the difficulties to obtain enough valid and representative data from all the basin, so training the model could lead to generalized models. Furthermore, the evidence that the Dempster-Shafer theory could compete with the ML methods suggests that what is needed is an Artificial Intelligence technique that can represent the knowledge from the experts, extrapolating this knowledge to the different positions in the basin.

It is worth recalling that all these approaches are focused on off-line risk assessment. Nonetheless, if we also consider the case of on-line information, not only GIS information is needed but also the sensory data and the weather forecasts could be needed in order to assess a certain scenario. With all these premises, this research proposes to use an alternative that has never been used in this context.

3 A CBR Proposal for the Flash-Flood Detection

The fundamental idea behind using the CBR is, given a current scenario, to i) retrieve the most relevant cases from past experiences and ii) merge the state of the basin from each one to conform a prediction on how the water level would be in advance. For a CBR proposal, the case representation must also be described. Furthermore, each of the four classic stages (Retrieval, Reuse, Revise, Retain) must be defined. For the purpose of this research, the Retain stage will not be implemented as it will be explained when discussing the case representation.

3.1 The Representation of a Past Experience

This study analyses the behaviour of hydrological basins, each basin including one or more locations where the prediction would be requested. A requested prediction would predict the state of the basin (in millimetres of water level) for up to 6 prediction horizons. A prediction horizon is a time step that is a function of the dynamics of the concrete basin; for instance, for some quick response basins, this time could be in minutes, while for more steady basins this period could be 1 h.

A case represents a past experience in any of the defined locations. Obviously, the case must include information from variables that represent the hidrological state of the basin. For this research, the hydrological experts from the different basins have stated the following variables (see Fig. 1):

- The average rainfall (RG) measured for the basin during the last 3 intervals.
- All the relevant water discharge level (WDL) sensors, measured for the basin during the last 6 intervals.
- The rainfall forecast (RF) for the next 3 intervals.

In these variables, an interval is a time step for which the values of the different sensors and measurements will be aggregated –the average of the values is computed–. Again, it relies on the hydrological properties of the concrete basin; for some basins this could be 15 min, for others it could be 1 h. These values, together with the prediction horizon step, the number of prediction horizons to consider, etc., are considered as parameters to the current location and basin.

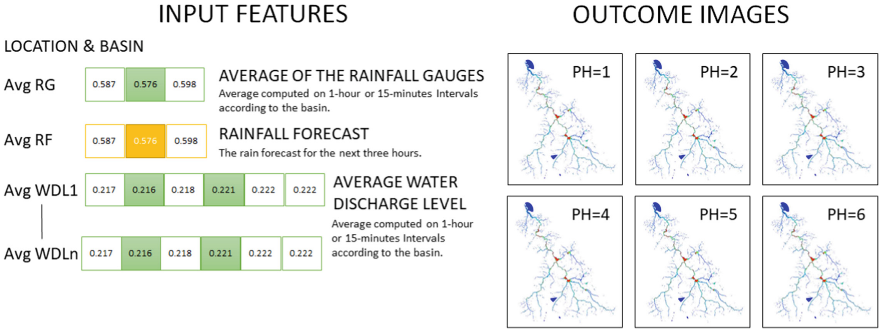

INPUT FEATURES
LOCATION & BASIN
Avg RG
0.587
0.576
0.598
AVERAGE OF THE RAINFALL GAUGES
Average computed on 1-hour or 15-minutes Intervals according to the basin.
Avg RF
0.587
0.576
0.598
RAINFALL FORECAST
The rain forecast for the next three hours.
Avg WDL1
0.217
0.216
0.218
0.221
0.222
0.222
Avg WDLn
0.217
0.216
0.218
0.221
0.222
0.222
AVERAGE WATER DISCHARGE LEVEL
Average computed on 1-hour or 15-minutes Intervals according to the basin.
OUTCOME IMAGES
PH=1
PH=2
PH=3
PH=4
PH=5
PH=6

Fig. 1. The case representation used in this research.

Moreover, the case also includes information for each of the prediction horizons considered for the corresponding location. In this research, a map of the basin representing the mm of water for each point in the basin is stored for each prediction horizon. For instance, in Fig. 1, there are 6 prediction horizons and, for each of them, a map of the basin with the mm of water is stored.

It is worth mentioning that all these maps and information is gathered from the results of the simulations of the basin using the hydrological models. For sure, there are well-defined hydrological models of the basins; however, running a simulation takes too much time and the outcomes are not expected to be available in a so tight time as a for an online request from the public safety experts. Hence, that is the reason why the whole CBR is designed: mimicking the performance of the hydrological models for the prediction of a basin behaviour.

3.2 The Three Rs CBR Process

As seen in the previous section, a case includes maps with the mm of water for each prediction horizon. This means that, in case of requiring the CBR to retain cases, these flood maps must be gathered for each prediction horizon so the CBR system could store them within the case. Nevertheless, this information is not available for the system: there is no method to recollect the mm of water for each point in a basin. Therefore, the retain phase of the CBR can not be completed and the CBR would only rely on the information extracted from simulations. We left to future work how to use ideas such concept drift to include extra information in the case base.

Hence, only three stages are to be defined: Retrieve, Reuse and Revise. For the retrieval phase, simple queries were performed retrieving the cases for the current location and basin. This might not be efficient in terms of computation; however, the scheme allows to define different distance measurements without the need of designing complex queries.

The Reuse stage is the one that performs the calculation of the distances between the retrieved cases from the case data base and the current scenario. For the purpose of this research, the square root of the weighted sum of the differences between the values for the variables describing the current scenario and a case is used as the distance measurement (see Eq. 1). To sort the cases and select the most interesting ones, single criteria sorting using the distance measurement and a predefined number of cases are the criteria to select the most relevant cases for this research. Besides, it is expected that more distances measurements can be defined, such as using the Sobol′ [8] indexes or introducing not only single criteria but multi criteria sorting -for instance, using the Pareto non-dominance concept-.

$$dist(current, case) = \frac{\sum w_i^{TH} \times (\frac{|v_i^{curr} - v_i^{case}|}{v_i^{max}})}{\sum w_i^{TH}} \tag{1}$$

Finally, the Revise stage generates the outcome of each case. To do so, the distance between each relevant case and the current scenario is used as a weight to calculate the weighted map among the maps from each relevant case (see Eq. 2). The idea is that the smaller the distance the higher the weight should be.

Therefore, each reused case is assigned with the product of the distances of the remaining cases; this value is then scaled in [0.0, 1.0]. Using these weights, and for each prediction horizon, the CBR calculates a weighted map as the agreement among the cases

$$w_i = \frac{\prod_{j \neq i} d(c_j, current\ scenario)}{\sum_j \prod_{j \neq i} d(c_j, current\ scenario)} \tag{2}$$

3.3 Exploiting the CBR Tool

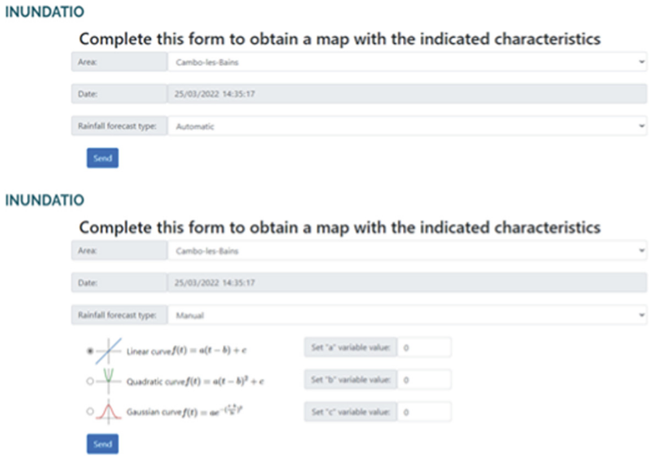
A web application has been developed for the INUNDATIO project, one of the interfaces is explicitly devoted to the basin behaviour forecast using the CBR. In this interface, the user introduces the basin and location, plus the information concerning the rain forecast. Two options are available: using the standard national forecast service or using a realistic rainfall forecast. In this latter case, the user chooses between a linear, a quadratic or Gaussian forecast by setting the functions parameters. This realistic forecast allows to evaluate the basin behaviour in case of extreme events.

Once the parameters are set, the system will call the CBR service and the sequence of maps is generated. The tool visualises the map of the mm of water a layer on top of an open-street map from the corresponding basin. An example of the interface is depicted in Fig. 2. By default, the tools request the defined weather forecast service for a rainfall prediction.

4 Experimentation and Discussion

This first prototype has been evaluated with one of the basins from the INUNDATIO project: the Venero-Claro basin (SPAIN). The Venero-Claro represents a small torrential hydrographical basin in Sierra de Gredos, Ávila. The Cabrera creek is a torrential-rain current, tributary to the Alberche river belonging to the Tajo basin. It refers to a series of small creeks in the northen side of Sierra del Valle; its vegetation is 45% Pinus Pinaster, but also includes Pinus Sylvestris and Alnus Glutinosa. The behaviour of this basin is of a very active fluvial-torrential hydrodynamics, with relatively short dynamic time periods -around 15 min to 45 min-.

A set of sensors has been placed in the basic during the execution the INUNDATIO project with the aim to develop the hydrodynamic models; these models will not be available during the INUNDATIO project, although the research to generate those models has been developed so far. As a consequence, for testing this CBR, the research team produce a realistic case base.



INUNDATIO

Complete this form to obtain a map with the indicated characteristics

Area: Cambo-les-Bains

Date: 25/03/2022 14:35:17

Rainfall forecast type: Automatic

Send

INUNDATIO

Complete this form to obtain a map with the indicated characteristics

Area: Cambo-les-Bains

Date: 25/03/2022 14:35:17

Rainfall forecast type: Manual

● Linear curve $f(t) = a(t-b)+c$ — Set "a" variable value: 0

○ Quadratic curve $f(t) = a(t-b)^2+c$ — Set "b" variable value: 0

○ Gaussian curve $f(t) = ae^{-(\frac{t-b}{c})^2}$ — Set "c" variable value: 0

Send

Fig. 2. Web interface for the CBR deployment. In the upper part, the automatic weather forecast interface; the manual rainfall forecast is shown in the bottom part.

For the development of this realistic case base three main scenarios were considered: non-saturated soil, mid-saturated soil and highly saturated soil. For each of these scenarios, the standard sensor values were estimated; these values were used to generate the scenario values for the different variables included in a case representation by random varying within certain bounds those estimated values. Furthermore, three possible forecasts were proposed: scarce rain forecast, soft rain forecast and heavy rain forecast. Again, typical rain forecast sequences for each of the possible forecasts were established; these forecast sequences were used in generating the rain forecast for each case. Finally, the outcome of each case were generated, one map for each prediction horizon. These maps were carefully chosen to represent plausible states of the basin given the basin variables and rain forecast.

To evaluate this first prototype of the CBR, different queries were performed by manually setting the rain forecast. The outcome of the tool must be such that the maps resemble the variances in mm of water in the basin. Because the tool automatically queries the on-line data bases for the basin sensory information, test with different basin initial conditions were not available.

The performance of the CBR is briefly outlines in Fig. 3. In this Figure, the CBR output map overlapped onto the basin map. The user can easily zoom in and out as well as pan over the map. This outcome comes from an scenario where the basin was suffering strong rain and then it stopped raining; the area in the upper left corner of the studied basin of Venero-Claro is completely flooded, this area is a low height plain.

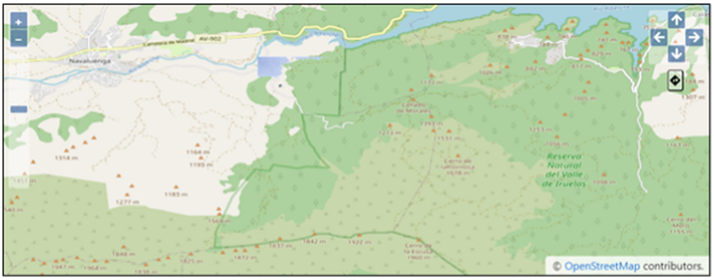


Fig. 3. CBR outcome for one prediction horizon. Notice that the limits of study abruptly ends the flooding. Besides, the common facilities for zoom and pan are available.

5 Conclusion

In this research, a Case-Based Reasoning tool has been proposed for the prediction of the behaviour of the basins; more specifically, to determine whether a fast-flood could arise or not. This proposal covers different types of basins, from those with fast response to those with a slower dynamic.

The results are still preliminary as long as the data from the different basins -that is, the relevant cases, the sources of information and forecast- are still in development. However, it shows a good capacity to predict what the experts expect from a given set of initial conditions. The evaluation has been performed with realistic data from the Venero-Claro basin in Spain; however, data from this basin and from the French basin of Nive will be available in a short time.

As future work, the implementation of different extensions, such as multi-criteria sorting using Pareto dominance or the availability of different distance measurements will be addressed.

Acknowledgements. This research has been founded by European Union's Horizon 2020 research and innovation programme (project DIH4CPS) under the Grant Agreement no 872548. Furthermore, this research has been funded by the SUDOE Interreg Program -grant INUNDATIO SOE3/P4/E0929-, by the Spanish Ministry of Economics and Industry, grant PID2020-112726RB-I00, by the Spanish Research Agency (AEI, Spain) under grant agreement RED2018-102312-T (IA-Biomed), by CDTI (Centro para el Desarrollo Tecnológico Industrial) under projects CER-20211003 and CER-20211022, by and Missions Science and Innovation project MIG-20211008 (INMERBOT). Also, by Principado de Asturias, grant SV-PA-21-AYUD/2021/50994 and by ICE (Junta de Castilla y León) under project CCTT3/20/BU/0002.

References

1. Li, W., Lin, K., Zhao, T., Lan, T., Chen, X., Du, H., Chen, H.: Risk assessment and sensitivity analysis of flash floods in ungauged basins using coupled hydrologic and hydrodynamic models. J. Hydrol. **572**, 108–120 (2019) https://doi.org/10.1016/j.jhydrol.2019.03.002. https://www.sciencedirect.com/science/article/pii/S0022169419302197
2. Lin, K., Chen, H., Xu, C.Y., Yan, P., Lan, T., Liu, Z., Dong, C.: Assessment of flash flood risk based on improved analytic hierarchy process method and integrated maximum likelihood clustering algorithm. J. Hydrol. **584**, 124696 (2020) https://doi.org/10.1016/j.jhydrol.2020.124696. https://www.sciencedirect.com/science/article/pii/S0022169420301566
3. Mishra, K., Sinha, R.: Flood risk assessment in the kosi megafan using multi-criteria decision analysis: A hydro-geomorphic approach. Geomorphology **350**, 106861 (2020). https://doi.org/10.1016/j.geomorph.2019.106861. https://www.sciencedirect.com/science/article/pii/S0169555X19303502
4. Ngo, P.T.T., Hoang, N.D., Pradhan, B., Nguyen, Q.K., Tran, X.T., Nguyen, Q.M., Nguyen, V.N., Samui, P., Tien Bui, D.: A novel hybrid swarm optimized multilayer neural network for spatial prediction of flash floods in tropical areas using sentinel-1 sar imagery and geospatial data. Sensors 18(11) (2018). DOI: https://doi.org/10.3390/s18113704,https://www.mdpi.com/1424-8220/18/11/3704
5. Pradhan, B., Lee, S.: Landslide susceptibility assessment and factor effect analysis: backpropagation artificial neural networks and their comparison with frequency ratio and bivariate logistic regression modelling. Environ. Modell. Softw. **25**(6), 747–759 (2010). https://doi.org/10.1016/j.envsoft.2009.10.016. https://www.sciencedirect.com/science/article/pii/S1364815209002886
6. Rahmati, O., Pourghasemi, H.R.: Identification of critical flood prone areas in data-scarce and ungauged regions: a comparison of three data mining models. Water Resour. Manage **31**(5), 1473–1487 (2017). https://doi.org/10.1007/s11269-017-1589-6
7. Shadmehri Toosi, A., Calbimonte, G.H., Nouri, H., Alaghmand, S.: River basin-scale flood hazard assessment using a modified multi-criteria decision analysis approach: a case study. J. Hydrol. **574**, 660–671 (2019). https://doi.org/10.1016/j.jhydrol.2019.04.072. https://www.sciencedirect.com/science/article/pii/S0022169419304123
8. Sobol, I.: Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. Math. Comput. Simul. **55**, 271–280 (2001)
9. Tehrany, M.S., Kumar, L.: The application of a Dempster–Shafer-based evidential belief function in flood susceptibility mapping and comparison with frequency ratio and logistic regression methods. Environmental Earth Sciences **77**(13), 1–24 (2018). https://doi.org/10.1007/s12665-018-7667-0
10. Tehrany, M.S., Pradhan, B., Jebur, M.N.: Spatial prediction of flood susceptible areas using rule based decision tree (dt) and a novel ensemble bivariate and multivariate statistical models in gis. J. Hydrol. **504**, 69–79 (2013). https://doi.org/10.1016/j.jhydrol.2013.09.034. https://www.sciencedirect.com/science/article/pii/S0022169413006872
11. Tehrany, M.S., Pradhan, B., Mansor, S., Ahmad, N.: Flood susceptibility assessment using gis-based support vector machine model with different kernel types. CATENA **125**, 91–101 (2015). https://doi.org/10.1016/j.catena.2014.10.017. https://www.sciencedirect.com/science/article/pii/S034181621400294X

12. Terêncio, D., Fernandes, L.S., Cortes, R., Moura, J., Pacheco, F.: Flood risk attenuation in critical zones of continental Portugal using sustainable detention basins. Sci. Total Environ. **721**, 137727 (2020). https://doi.org/10.1016/j.scitotenv.2020.137727. https://www.sciencedirect.com/science/article/pii/S0048969720312389
13. Wyżga, B., Kundzewicz, Z.W., Konieczny, R., Piniewski, M., Zawiejska, J., Radecki-Pawlik, A.: Comprehensive approach to the reduction of river flood risk: case study of the upper vistula basin. Sci. Total Environ. **631–632**, 1251–1267 (2018). https://doi.org/10.1016/j.scitotenv.2018.03.015. https://www.sciencedirect.com/science/article/pii/S0048969718307708
14. Zhang, Y., Wang, Y., Chen, Y., Liang, F., Liu, H.: Assessment of future flash flood inundations in coastal regions under climate change scenarios-a case study of hadahe river basin in northeastern china. Sci. Total Environ. **693**, 133550 (2019). https://doi.org/10.1016/j.scitotenv.2019.07.356. https://www.sciencedirect.com/science/article/pii/S0048969719334709
15. Ţîncu, R., Zêzere, J.L., Crăciun, I., Lazăr, G., Lazăr, I.: Quantitative micro-scale flood risk assessment in a section of the trotuş river, romania. Land Use Policy **95**, 103881 (2020). https://doi.org/10.1016/j.landusepol.2019.02.040. https://www.sciencedirect.com/science/article/pii/S0264837718311116

Weakly Supervised Learning of the Motion Resistance of a Locomotive Powered by Liquefied Natural Gas

Luciano Sánchez[1](✉), Pablo Luque[2], Daniel Álvarez-Mántaras[2], José Otero[3], and Nahuel Costa[3]

[1] Universidad de Oviedo, C. de la Computación e Inteligencia Artificial, Oviedo, Spain
luciano@uniovi.es

[2] Universidad de Oviedo, Ingeniería e Infraestructura de los Transportes, Oviedo, Spain
{luque,mantaras}@uniovi.es

[3] Universidad de Oviedo, Lenguajes y Sistemas Informáticos, Oviedo, Spain
{jotero,costanahuel}@uniovi.es

Abstract. A model of the running resistance of a locomotive powered by liquefied natural gas is proposed. The model uses operating data and does not require specific instrumentation. The input data consists of a succession of instantaneous speed and electrical power measurements of a diesel-electric locomotive. The slope at each point along the route is unknown and the speed is measured with a digital sensor that quantifies the signal, so acceleration estimates are also unreliable. From these data, a weakly supervised learning problem is defined that makes use of a fuzzy rule-based system to indirectly predict the effective slope, and is able to estimate the power demand of the locomotive with a margin of error close to 5%.

1 Introduction

Most trains are powered by electric or diesel engines. Electric traction is more efficient and (depending on the electric generation mix) has environmental advantages [3]. However, on non-electrified tracks, the economic and environmental impact of conversion works can offset the advantages of electric traction. For this reason, different alternative fuels to diesel, such as biodiesel, hydrogen or natural gas, are being investigated in the railway sector. In particular, the use of compressed natural gas (CNG) or liquefied natural gas (LNG) is an option that is being actively studied, as it is a widely available fuel and requires relatively low investments to adapt refueling and maintenance infrastructure [2].

The environmental benefits of converting a conventional diesel vehicle to natural gas depend on its usage pattern. In all cases, a compromise must be reached between emission reductions, vehicle range and the economic costs of operating the vehicle. In passenger vehicles, the reduction of pollutant emissions

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 606–615, 2023.
https://doi.org/10.1007/978-3-031-18050-7_59

in populated areas is the main factor [1], while in freight transport the routes are longer and the locomotives must have sufficient autonomy. For this reason, LNG is preferred in heavy vehicles (locomotives), which allows denser energy storage [2].

The assessment of the economic and environmental impact of using LNG-powered locomotives is not immediate. Spark-ignition LNG engines are less energy efficient than their diesel counterparts, and the increased consumption may offset the reduction in greenhouse gas emissions. There are numerous studies comparing carbon and pollutant gas emissions in different service patterns, but it cannot be concluded that LNG engines are superior to diesel in all circumstances. A customized study is necessary for each route and vehicle, taking into account the speed profiles and the running resistance at each point of the route [4].

The determination of the running resistance requires specific tests with instrumented vehicles, the cost of which is high. In previous works we have developed a procedure to obtain this parameter indirectly in passenger vehicles with diesel engines, with significant economic savings [6]. Continuing with this line, in this study we propose a numerical method that allows estimating the forward resistance from the data available in the control desk of a diesel-electric traction locomotive for freight transport, without the need for specific tests or additional instrumentation.

The differences between the method proposed in this contribution and previous works are due to the lack of availability of some of the operating data needed to fit the model. In this study we have data from a diesel-electric locomotive captured on a commercial route. The locomotive carries a load on the outbound trip and makes the return trip without a load. A succession of instantaneous speed and electric power measurements is available, but the slope at each point along the route is unknown and the speed is measured with a digital sensor that quantizes the signal, so the acceleration estimates are also unreliable. A weakly supervised learning problem (with imprecise supervision [9]) is proposed that consists of modeling the running resistance as a function of the quantized speed of the locomotive (aerodynamic effects) and the slope (which is also unknown).

The outline of the paper is as follows: Sect. 2 gives a formal approach to the learning problem to be solved, and introduces a set of simplifications that reduce its computational complexity. Section 3 describes the architecture of the proposed model. Section 4 provides numerical results on a commercial route. Section 5 concludes the study and proposes future lines of work.

2 Problem Statement

The variables measured are as follows:

- The power $p(t)$ delivered to the traction motors at each instant of time.
- A discretized value $s(t)$ of the instantaneous velocity $v(t)$: $v(t) \in [s(t) - \delta, s(t) + \delta]$, where δ is the resolution of the velocity sensor.

- Actuation signals and pressures in pneumatic braking circuits.
- Alternator excitation currents during traction and braking.
- Battery power consumption.
- Locomotive and towed masses.

We wish to estimate the forward resistance at each point of the path, in the outward and return directions, and determine the tolerance of this estimate. The forward resistance in the "outward" direction for the point on the path that is d kilometers away from the origin is

$$R^{\mathrm{ow}}(d) = m^{\mathrm{ow}}(R_f(v^{\mathrm{ow}}(d)) + g \tan \alpha(d)) \tag{1}$$

where $v^{\mathrm{ow}}(d)$ is the speed of the composition at point d of the path, in the "outward" direction, and R_f models the aerodynamic and friction effects at that point in the route. The variables m, g and α have their usual meaning: mass, gravity and slope. In the "inward" direction, the running resistance is similar, but the slope is inverted:

$$R^{\mathrm{iw}}(d) = m^{\mathrm{iw}}(R_f(v^{\mathrm{iw}}(d)) - g \tan \alpha(d)). \tag{2}$$

Let T be the tractive effort of the motor, F be the braking effort and a be the acceleration of the composition. At each point of the circuit the following balance of forces is satisfied:

$$T^{\mathrm{ow}}(d) - F^{\mathrm{ow}}(d) = R^{\mathrm{ow}}(d) + m^{\mathrm{ow}} a^{\mathrm{ow}}(d) \tag{3}$$

$$T^{\mathrm{iw}}(d) - F^{\mathrm{iw}}(d) = R^{\mathrm{iw}}(d) + m^{\mathrm{iw}} a^{\mathrm{iw}}(d). \tag{4}$$

The well-known Davis equation aggregates friction and aerodynamic effects into a quadratic expression,

$$R_f^{\mathrm{iw}}(v) = A_L + B_L v + C_L v^2. \tag{5}$$

In the case where the locomotive pulls loaded wagons, the forward resistance of the composition shall have the following terms [8]:

$$R_f^{\mathrm{ow}}(v) = A_L + B_L v + C_L v^2 + A_W + C_W v^2. \tag{6}$$

If the braking effort is zero, then the tractive effort at each point of the route can be estimated from the power generated by the engine (discounting the power dedicated to auxiliary consumption, such as battery charging or alternator excitation):

$$T(t) = \frac{p(t)}{v(t)} \tag{7}$$

when the braking effort is not zero we will consider that the traction effort is zero (i.e., the motor does not pull and brake at the same time). Let $d(t)$ be the point in the circuit where the locomotive is at the instant t. At the instants of time when the motor is tractioning (i.e., $T(d(t)) > 0$ and $F(d(t)) = 0$), we will have:

$$v^{\mathrm{ow}}(t) \in [s^{\mathrm{ow}}(t) - \delta, s^{\mathrm{ow}}(t) + \delta], \quad 0 \leq t \leq T^{\mathrm{ow}} \tag{8}$$

$$v^{\mathrm{iw}}(t) \in [s^{\mathrm{iw}}(t) - \delta, s^{\mathrm{iw}}(t) + \delta], \quad T^{\mathrm{ow}} \leq t \leq T^{\mathrm{ow}} + T^{\mathrm{iw}} \tag{9}$$

$$d^{\mathrm{ow}}(t) = \int_0^t v^{\mathrm{ow}}(\tau) d\tau \tag{10}$$

$$d^{\mathrm{iw}}(t) = d^{\mathrm{ow}}(T^{\mathrm{ow}}) - \int_{T^{\mathrm{ow}}}^t v^{\mathrm{iw}}(\tau) d\tau \tag{11}$$

$$\hat{p}^{\mathrm{ow}}(t) = v^{\mathrm{ow}}(t) m^{\mathrm{ow}} \left(R_f^{\mathrm{ow}}(v^{\mathrm{ow}}(t)) + g \tan \alpha(d^{\mathrm{ow}}(t)) + \left.\frac{dv^{\mathrm{ow}}}{dt}\right|_t \right), \text{ for } p^{\mathrm{ow}}(t) > 0 \tag{12}$$

$$\hat{p}^{\mathrm{iw}}(t) = v^{\mathrm{iw}}(t) m^{\mathrm{iw}} \left(R_f^{\mathrm{iw}}(v^{\mathrm{iw}}(t)) - g \tan \alpha(d^{\mathrm{iw}}(t)) + \left.\frac{dv^{\mathrm{iw}}}{dt}\right|_t \right), \text{ for } p^{\mathrm{iw}}(t) > 0 \tag{13}$$

where the input data are the circuit travel times in both directions T^{ow} and T^{iw}, the discrete velocities $s^{\mathrm{ow}}(t)$ and $s^{\mathrm{iw}}(t)$, the δ resolution of the speed sensor, the masses m^{ow} and m^{iw} of the composition in both directions and the values $\alpha(d)$, $v^{\mathrm{ow}}(t)$ and $v^{\mathrm{iw}}(t)$ defining respectively the slope at each point of the circuit and the instantaneous velocities. The output data comprise the instantaneous power $p^{\mathrm{ow}}(t)$ and $p^{\mathrm{iw}}(t)$.

Note that the mappings $R_f^{\mathrm{ow}}(v)$, $R_f^{\mathrm{iw}}(v)$, $\alpha(d)$, $v^{\mathrm{ow}}(t)$ and $v^{\mathrm{iw}}(t)$ are partially or completely unknown. The only information about $v^{\mathrm{ow}}(t)$ and $v^{\mathrm{iw}}(t)$ is given by Eq. 8 and the values of $\alpha(d)$ are missing. In the next section, a weakly supervised learning method is proposed whereby these inputs are assigned values compatible with the observations and a model of the instantaneous power is derived.

3 Proposed Method

The proposed learning problem is weakly supervised for two reasons. On the one hand, the velocity is not known but an interval containing it. On the other hand, the slopes (which are the most relevant factor in the estimation of the developed power) are unknown except for a small amount of information: the slope profile on the return trip is the same profile as the profile on the outward trip, with the opposite sign.

The learning problem is solved by assigning slopes with a non-parametric regression model whose coefficients will be determined indirectly. The slope imputed by the regression model will be one of the inputs of the locomotive power prediction model. Figure 1 summarizes the architecture of the power prediction model set forth in Eqs. 8-13. The inputs are time, vehicle mass and direction of travel, and the output is the power developed by the traction motors. The running speed is determined from Eqs. 8 and 9 ("Speed model" in the Figure), and the running resistance is defined by Eqs. 5 and 6 ("Running resistance model").

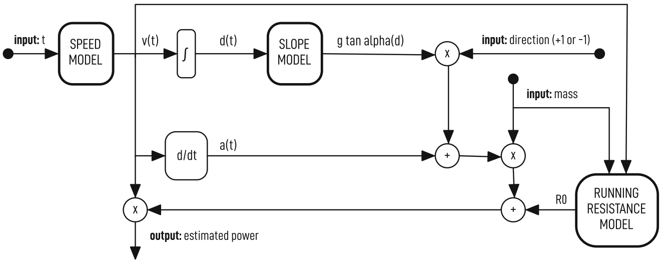


Fig. 1. Schematic diagram of the proposed system architecture

3.1 Speed Model

The velocity model is intended to smooth the velocity profile to improve acceleration estimation. The raw data consists of a profile composed of constant velocity sections with sharp jumps due to the quantization introduced by the digital sensor. The desired solution is the smoothest curve compatible with Eqs. 8 and 9; note that this is the definition of Support Vector Regression (SVR), so a ϵ-SVR fit with epsilon-insensitive hinge loss has been used as the velocity estimate [7]. A margin of tolerance of value δ is used where no penalty is given to error, and the velocity samples are the support vectors.

3.2 Running Resistance Model

The running resistance model is reduced to a quadratic expression that depends on five coefficients (Eqs. 5 and 6) that define the aerodynamic and frictional effects of the locomotive and the towed load. Tunnels, curve resistance and other dynamic factors, such as starting resistance, have not been taken into account in this phase of the study.

3.3 Slope Model

The slope model is a mapping that relates the distance to the origin of the path to the slope at that point in the path. Any nonparametric regression model, such as a lookup table, a neural network, or a fuzzy rule-based system is potentially suitable. In the next section we will experimentally compare the results of performing an estimation with different model types and show that a fuzzy rule-based system achieves the best balance between the complexity of the model and the accuracy of the estimation. Note that the choice of slope model determines the type of optimization problem to be solved to obtain the forward resistance, and has a major influence on the accuracy of the solution, as discussed below.

3.4 Numerical Optimization Problem

The numerical optimization problem consists of minimizing the difference between the power predictions on the outward and return paths (Eqs. 12 and

13) with respect to the parameters A_L, B_L, C_L, A_W, C_W and the parameters θ defining the slope model (i.e. the values defining the lookup table, the neural network weights or the coefficients associated with each rule, if applicable). Apart from the constraints associated with θ, the parameters of the running resistance model are non-negative:

$$\begin{aligned}&\text{minimize} \sum_{i\in \text{ outward}} (\hat{p}^{\text{ow}}(t_i) - p^{\text{ow}}(t_i))^2 + \sum_{i\in \text{ inward}} (\hat{p}^{\text{iw}}(t_i) - p^{\text{iw}}(t_i))^2 \\ &\text{subjected to:} \\ &A_L, B_L, C_L, A_W, C_W \geq 0\end{aligned} \tag{14}$$

4 Numerical Results

The proposed model has been validated with real data from a locomotive operating in northern Spain. The mass of the locomotive is 60 tons, and the towed load on the outbound route is 160 tons. The route has a length of 209 km. The parameters of the running resistance model were estimated using five different methods:

1. Supervised problem (no slope imputed)
2. Slope imputation by means of a neural network with two hidden layers (topology 1-6-6-1, with a total of 61 weights)
3. Slope allocation via lookup table (50 sections)
4. Slope imputation by fuzzy rule-based system (FRBS) with triangular membership (50 rules)
5. Slope imputation by a FRBS with Gaussian membership (50 rules).

The search table sections are correlative and of the same length. A constant slope is associated with each of them, so in this case the slope model depends on 50 parameters. The activation function in the hidden layers of the neural network is $\tanh(\cdot)$ and the activation of the output layer is linear. The fuzzy rules are of the form

$$\text{If } d \in \widetilde{A}_i \text{ then slope} = S_i$$

Table 1. Compared results

Slope model type	Learning type	RMS error	Motion resistance estimates				
			A_L	B_L	C_L	A_W	C_W
None	Superv	93.48	8.87	0.00	0.00	3.13	7.0×10^{-4}
DNN (1-6-6-1)	Semi-S	48.36	1.90	9.43×10^{-2}	1.1×10^{-4}	0.99	5.49×10^{-4}
Lookup table (50)	Semi-S	37.72	0.74	0.09	0.0	1.78	0.0
RB (50)/fuzzy triangular	Semi-S	17.26	5.71	1.03×10^{-6}	6.77×10^{-5}	1.44	1.58×10^{-4}
RB (50)/fuzzy gaussian	Semi-S	**15.71**	5.71	0.00	1.10×10^{-5}	1.44	5.43×10^{-5}

where $\widetilde{A}_i$ are triangular or Gaussian fuzzy memberships forming a uniform partition (see [5]) of the interval $[0, 209]$. In both cases there are 50 coefficientes, i.e. $\theta = (S_1, \ldots, S_{50})$. In all cases, the minimum of the problem defined in the Eq. 14 has been sought using the quasi-Newton descent algorithm L-BFGS-B, capable of handling box constraints.

Note first that the supervised model fit is not sufficient to accurately obtain the running resistance coefficients. The fit of this model is shown graphically at the top of Figs. 2 and 3. The upper part of both figures shows the fit of the supervised model, and the lower parts show the best model from Table 1. The coefficients of the running resistance polynomial correspond to a speed expressed

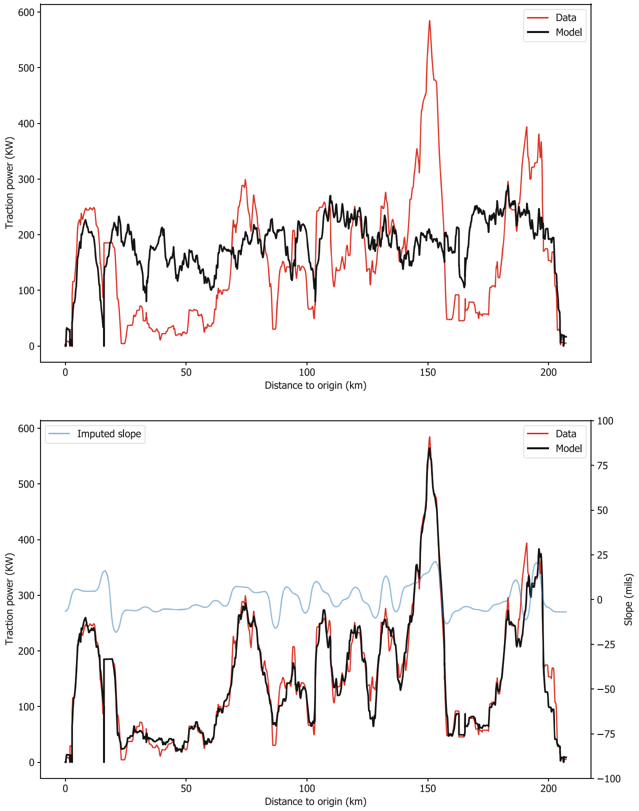


Fig. 2. Outward path. Upper part: fit of the supervised model, which does not use the "slope" variable. Lower part: fit of the weakly supervised model, with imputation of the "slope" variable by means of a FRBS with Gaussian membership. The red curves are the measured power values, and the black curves are the predictions of the best model obtained. The blue curve in the lower graph shows the imputed values for the "slope" variable.

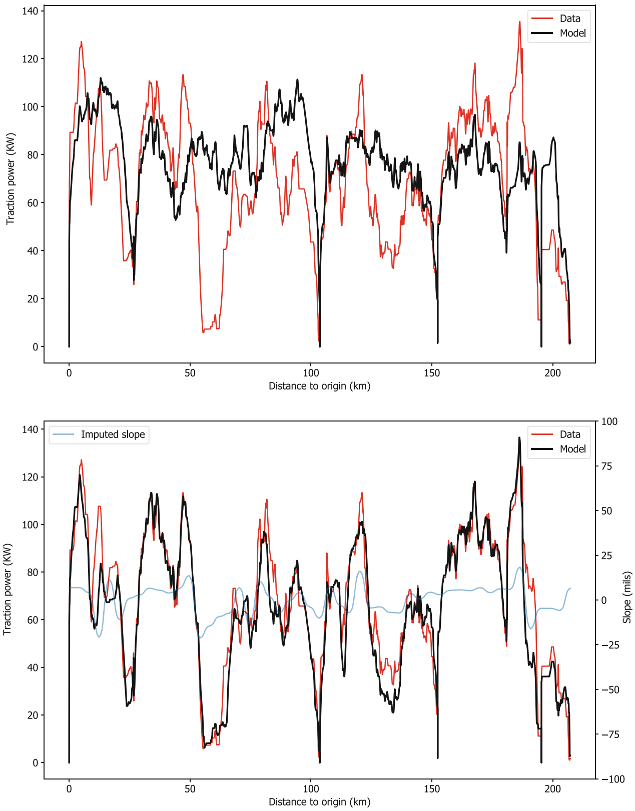


Fig. 3. Inward path. Upper part: fit of the supervised model, which does not use the "slope" variable. Lower part: fit of the weakly supervised model, with imputation of the "slope" variable by means of a FRBS with Gaussian membership. The red curves are the measured power values, and the black curves are the predictions of the best model obtained. The blue curve in the lower graph shows the imputed values for the "slope" variable (same as slope in Fig. 2, with the opposite sign.)

in km/h and a resistance measured in daN/ton. The RMS error corresponds to a power measured in KW. In terms of the approximation between the total energy developed in the path, the model error is less than 5%.

A final experimental validation of the procedure has been performed in Fig. 4. This figure shows in red the actual values of the slope of the test track, which we measured with a GPS sensor. The GPS sensor is not part of the locomotive's standard instrumentation, so these GPS measurements were not provided to the model. Note, however, that the FRBS-based slope prediction in the proposed weakly supervised learning procedure (recall Fig. 1, box "slope model") accurately matches these GPS measurements.

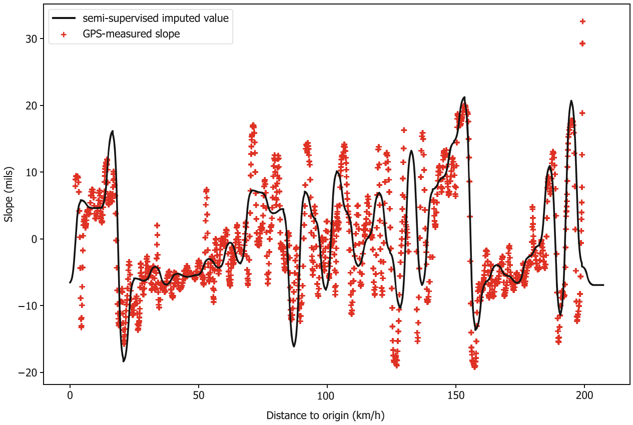


Fig. 4. Experimental validation of the procedure: the values imputed for the input variable "slope" by means of the FRBS are compared with the true slope, measured by means of a GPS sensor. It is emphasized that the red dots have not been used in the model; the black curve has been obtained by applying the proposed method to locomotive operating data, which do not include altitude.

5 Concluding Remarks and Future Work

This paper shows how a small amount of information (that the slopes on the outgoing and return paths are the same, with the opposite sign) has allowed to couple two different models and substantially reduce the approximation error by weakly supervised learning techniques. A rule-based model has been defined for the unknown input and a cascaded combination of this and other models has been optimized. The final result contains the desired value of the vehicle's forward resistance and also an estimate of the slopes in the different sections of the route. The latter estimate has been validated with an independent set of experimental measurements. The proposed forward resistance estimation is economically advantageous, as it does not require track testing and locomotive instrumentation, since the signals used are part of the locomotive control circuits.

In future work we will incorporate into this study a model of the internal combustion engine that relates the traction power to the operating point of the engine, so that predictions can be made not only of the energy consumed along the journey, but also comparisons of consumption and emissions of diesel and LNG engines on different journeys can be made.

Acknowledgements. This work has been partially funded by grants from the Spanish Ministry of Economy, Industry and Competitiveness (ref. PID2020-112726-RB-I00) and the Principality of Asturias (ref. SV-PA-21-AYUD/2021/50994).

References

1. Corfa, E., Maury, F., Segers, P., Fresneau, A., Albergel, A.: Short-range evaluation of air pollution near bus and railway stations. Sci. Total Environ. **334**, 223–230 (2004)
2. Dincer, I., Zamfirescu, C.: A review of novel energy options for clean rail applications. J. Nat. Gas Sci. Eng. **28**, 461–478 (2016)
3. Hoffrichter, A., Miller, A.R., Hillmansen, S., Roberts, C.: Well-to-wheel analysis for electric, diesel and hydrogen traction for railways. Transp. Res. Part D: Transp. Environ. **17**(1), 28–34 (2012)
4. Langshaw, L., Ainalis, D., Acha, S., Shah, N., Stettler, M.E.: Environmental and economic analysis of liquefied natural gas (lng) for heavy goods vehicles in the uk: a well-to-wheel and total cost of ownership evaluation. Energy Policy **137**, 111161 (2020)
5. Ruspini, E.H.: Numerical methods for fuzzy clustering. Inf. Sci. **2**(3), 319–350 (1970)
6. Sanchez, L., Luque, P., Alvarez, D.: Assessment of the running resistance of a diesel passenger train using evolutionary bilevel algorithms and operational data. Eng. Appl. Artif. Intell. **105**, 104405 (2021)
7. Suykens, J.A., Vandewalle, J.: Least squares support vector machine classifiers. Neural Process. Lett. **9**(3), 293–300 (1999)
8. Szanto, F.: Rolling resistance revisited. In: CORE 2016, Maintaining the Momentum, Conference on Railway Excellence, Melbourne, Victoria, 16-18 May 2016 (2016)
9. Zhou, Z.H.: A brief introduction to weakly supervised learning. Natl. Sci. Rev. **5**(1), 44–53 (2018)

Node Location Optimization for Localizing UAVs in Urban Scenarios

Paula Verde(✉), Rubén Ferrero-Guillén, José-Manuel Alija-Pérez, Alberto Martínez-Gutiérrez, Javier Díez-González, and Hilde Perez

Department of Mechanical, Computer and Aerospace Engineering,
Universidad de León, 24071 León, Spain
{paula.verde,jmalip,amartg,javier.diez,hilde.perez}@unileon.es,
rferrg00@estudiantes.unileon.es

Abstract. Unmanned-Aerial-Vehicles (UAV) widespread use has grown significantly in recent years. Their insertion in the civil sector allows their implementation in the agricultural and industrial sectors, as well as their use for surveillance or delivery applications. However, the efficient development of these applications depends on the drone's ability to position itself autonomously. Although it is common to find drones with satellite positioning systems (GNSS), these systems are insufficient for autonomous navigation in urban or indoor environments. In these scenarios, the implementation of local positioning systems (LPS) is widely extended due to their adaptability capabilities. Through the optimal distribution of the sensors that constitute this system, they can adapt to almost any environment while also improving its performance. However, the complexity of this problem has been characterized as NP-Hard, which complicates its resolution. In this paper, a genetic algorithm is developed to optimize LPS in different environments. This algorithm, pioneer in the design of LPS for UAV localization, is tested on a generated urban environment. The results obtained denote the effectiveness of the methodology by obtaining location uncertainties significantly lower than GNSS.

Keywords: Unmanned-Aerial-Vehicles · Local positioning systems · Genetic Algorithms · Cramèr-Rao Bounds

1 Introduction

Unmanned-Aerial-Vehicles (UAV), usually referred to as drones, constitute a modality of vehicles whose use have grown significantly over the last years. While the initial development of these aircraft was mostly pursued for military operations, their current widespread use is characterized by their insertion into the civil sector [1].

The wide range of applications of these vehicles includes their use in surveillance [2], structural inspection [3], delivery services [4], industrial transport [5]

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 616–625, 2023.
https://doi.org/10.1007/978-3-031-18050-7_60

and agricultural services [6]. The introduction of UAV into these already existing applications provides multiple benefits, such as a higher mobility than land vehicles and the autonomous realization of arduous and repetitive tasks.

However, the efficient development of these applications depends on the drone navigation capabilities. Most drones are operated through a visual navigation system, where the user positions the drone based on its visual perception and based on the recorded drone images.

Although this methodology may be sufficient for certain applications, such as photographic and recreational uses, this localization methodology is inadequate for low visibility flight conditions and for autonomous applications, such as drone delivery [7].

Nevertheless, some drones complement visual navigation with satellite positioning systems (GNSS), however, these systems are insufficient for high precision applications [8]. Furthermore, their localization performance may be compromised in scenarios where the reception of the GNSS signal is compromised, such is the case of indoor and urban environments [9], thus affecting the positioning of UAV for most of their uses.

In this context, Local Positioning Systems (LPS) have been widely proposed in the literature for constituting a solid positioning methodology for drones in harsh environments [10]. LPS are localization systems based on the deployment of Wireless Sensor Networks (WSN) over a given region.

LPS determine the location of a target through the measurement of different physical properties (e.g., power, angle, frequency) [11]. However, time-based LPS represent the most extended localization systems due to their trade-off among accuracy, stability, hardware implementation ease and costs [12].

Furthermore, multiple time-based LPS architectures are commonly encountered throughout the literature. Time Difference of Arrival (TDOA) [13] and Time of Arrival (TOA) architectures constitutes the most expanded time-based LPS [14].

While the TOA architecture entails a more versatile deployment of sensors for the positioning performance, the measurement of the total time of flight of the positioning signal demands a complete synchronization among the network nodes and the localization targets. The enforcement of such restrictions over civil drones would elevate the drone acquisition costs and the overall viability of the system [15].

On the other hand, TDOA architectures perform the localization through the measurement of the time difference of arrival of the positioning signals, which do not require the target clock synchronization with the LPS architecture [16], thus promoting the use of TDOA LPS for UAV localization [17].

Nevertheless, for any architecture, the coverage, localization uncertainties and the overall performance of LPS depends on the actual position of all the nodes that constitute them. The obtention of the optimal Cartesian coordinates of each node entails the addressing of the Node Location Problem (NLP), a combinatorial optimization problem categorized as NP-Hard [18]. This optimization represents a vital step for the development of a LPS for localizing UAV, yet currently unresolved in the literature for these devices.

In order to achieve the optimal solution to the NLP, different metaheuristic techniques have been implemented throughout the literature for this problem, including Genetic Algorithms (GA) [19], grey wolf optimization [20], butterfly optimization [21], simulated annealing [22] and differential evolution among others [23]. However, GA stand as one of the most utilised methodology due to the overall robustness of this technique and due to the attained balance between diversification and intensification in the optimization [19].

Furthermore, LPS also entail a different optimization problem once the sensors have been fixed in place, the Sensor Selection Problem (SSP). This problem entails the definition of the optimal subset of sensors for localizing each potential location under coverage. The SSP has also been categorized as NP-Hard and represents a cardinal step for improving the performance of LPS [24].

Therefore, in this paper, a NLP optimization methodology is proposed for the first time in the authors' best knowledge for devising the optimal sensor distribution for UAV in urban scenarios. The proposed optimization also addresses the SSP within the NLP in order to optimize the overall performance of the positioning system, since the resolution of the SSP for a fixed node distribution is restricted by that particular node deployment.

In order to attain both problems simultaneously, a GA optimization is proposed for obtaining the optimal solution for both problems using the Cramèr-Rao Bounds (CRB) as an unbiased estimator of the system performance [25]. The proposed methodology is then evaluated in a realistic urban environment for determining a robust positioning system for UAV.

The remainder of the paper is organized as follows: Sect. 2 describes the mathematical model regarding the NLP and the SSP, detailing the proposed approach for combining both optimizations within the same GA optimization; Sect. 3 describes the proposed GA for devising an optimal node distribution for UAV within urban environments, while Sect. 4 and 5 presents the attained results and conclusions of this paper.

2 Mathematical Model of the NLP

The performance of any LPS is heavily dependent on the distribution of the sensors that constitute the WSN over the studied region. The NLP aims to find the subset of optimal Cartesian coordinates (S_l) for each node of the localization system that maximizes the coverage region while minimizing the localization uncertainties, where (S_l) can be characterized as:

$$\langle S_l \rangle = (P_1, P_2, \dots, P_n); \quad P_i = (x_i, y_i, z_i); \quad i \in \{1, 2, \dots, n\} \tag{1}$$

where P_i represents the Cartesian coordinates of sensor i and n is the total number of sensors deployed. The subset containing the combination of optimal positions (S_l) is contained within the set of all possible sensor distributions (S).

The number of possible solutions can be deduced from the combinatorial nature of the NLP, resulting in the following expression:

$$C = \prod_{i=1}^{n} (n_{NLE} - i) \tag{2}$$

where C is the number of possible solutions (i.e., possible sensor distributions) and n_{NLE} the total number of NLE (Node Location Environment) considered during the optimization, which represents the number of possible sensor locations. Furthermore, the NLP can be described through the following mathematical model:

$$\begin{gathered}
\text{Maximize } Z = ff_{CRB} \\
\text{Subject to:} \\
x_{lim_1} \leq x_i \leq x_{lim_2};\ \forall\ x_i \in S;\ S \not\subset U \\
y_{lim_1} \leq y_i \leq y_{lim_2};\ \forall\ y_i \in S;\ S \not\subset U \\
z_{lim_1} \leq z_i \leq z_{lim_2};\ \forall\ z_i \in S;\ S \not\subset U \\
Cov_{TLE_k} \geq n_{min\ TDOA};\ \forall\ k \in n_{TLE} \\
Cov_{TLE_k} = \sum_{i=1}^{n} Cov_{TLE_{k_i}} \\
Cov_{TLE_{k_i}} = \begin{cases} 1 \text{ if } SNR_{TLE_{ki}} \geq SNR_{lim} \\ 0 \qquad\quad \text{otherwise} \end{cases}
\end{gathered} \tag{3}$$

where ff_{CRB} is the CRB fitness function; $x_{lim_1}, y_{lim_1}, z_{lim_1}$ and $x_{lim_2}, y_{lim_2}, z_{lim_2}$ represent the lower and upper limits of the scenario discretization; U is the subset containing all invalid sensor distributions; Cov_{TLE_k} is the number of sensors under coverage for the scenario point k; $n_{min\ TDOA}$ is the minimum number of sensors for localizing a scenario point, requiring 5 nodes for the TDOA architecture [26]; n_{TLE} is the number of discretized points of the TLE (Target Location Environment), which represents the number of discretized points where a UAV may be located; Cov_{TLE_k} represents the coverage for sensor i over point k of the TLE; $SNR_{TLE_{ki}}$ is the signal-to-noise ratio of the positioning signal emitted from sensor i and received from location k and SNR_{lim} is the limit value for which the positioning signal can be deciphered.

The main optimization objective is to maximize a fitness function based on the CRB, a maximum likelihood estimator computed from the Fisher Information Matrix (FIM) that determines the minimum variance of an unbiased estimator [25].

In the localization field, the CRB is widely used as a performance index, since it represents the minimum achievable localization error and results applicable for heteroscedastic noise propagation conditions, a common model in LPS [27]. The CRB can be derived for any sensor architecture by characterizing the covariance matrix ($\boldsymbol{R}$), following the FIM expression [28]:

$$FIM_{m,n} = \left(\frac{\partial h(TS)}{\partial TS_m}\right) R^{-1}(TS) \left(\frac{\partial h(TS)}{\partial TS_n}\right) \\ + \frac{1}{2} tr \left\{ \boldsymbol{R}^{-1}(TS) \left(\frac{\partial R(TS)}{\partial TS_m}\right) \boldsymbol{R}^{-1}(TS) \left(\frac{\partial \boldsymbol{R}(TS)}{\partial TS_n}\right) \right\} \quad (4)$$

where $FIM_{m,n}$ represents the (m, n) element of the FIM and $h(TS)$ is the vector that contains the signal propagation trajectory from the Target Sensor (TS) to each pair of the architecture sensors [26].

The localization uncertainty for each analysed point of the TLE can then be obtained through the Root Mean Square Error (RMSE) of the trace of the inverse of the FIM. From this parameter, the following fitness function for evaluating the performance of a node distribution is proposed:

$$ff_{CRB} = \frac{1}{n_{TLE}} \sum_{k=1}^{n_{TLE}} 1 - \frac{RMSE_k}{RMSE_{ref}}; \quad RMSE_k = \sqrt{trace\left(\boldsymbol{FIM^{-1}}\right)} \quad (5)$$

where $RMSE_{ref}$ is the reference RMSE that represents the maximum achievable RMSE value in the analysed scenario, thus the domain of ff_{CRB} is contained in the interval $[0, 1]$.

Furthermore, the RMSE for each point k of the TLE is obtained from the combination of sensors that results in the least RMSE error for that point. Through this combined consideration, knowledge of the SSP is introduced during the NLP optimization, guiding the optimization into a solution that satisfies both problems unbiased by a fixed node distribution [29].

However, the NLP has been already categorized as NP-Hard [18], consequently, varying heuristic methodologies have been proposed for obtaining a near optimal solution within a polynomial time. Amongst these techniques, Genetic Algorithms represent one of the most expanded methodologies due to their overall robustness and performance, providing a trade-off between exploration and intensification in the performed optimization [19].

3 Genetic Algorithm Optimization

GA address optimization problems through different methodologies founded in the theory of evolution, representing one of the most expanded branches within Evolutionary Algorithms (EA). These algorithms achieve the optimal solution of a problem by enforcing the adaptation of an encoded population of individuals, where each individual represents a particular solution to the problem [30].

The followed codification for the attained problem is depicted in Fig. 1, where each individual represents a particular sensor distribution. This distribution is characterized as the combination of a series of sensor coordinates, where each node location is composed of the three Cartesian coordinates, coded in binary.

However, due to the irregularity of the proposed scenario, containing surface deviation and obstacles, a scale decodification from binary to a decimal value is proposed. In the proposed decodification, each bit within each Cartesian coordinate may represent a different value depending on the values of the other coordinates, as detailed in [19].

Finally, individuals are evaluated through the fitness function described in Eq. 5, which evaluates the performance of each sensor distribution based on the Crámer-Rao of the sensor architecture.

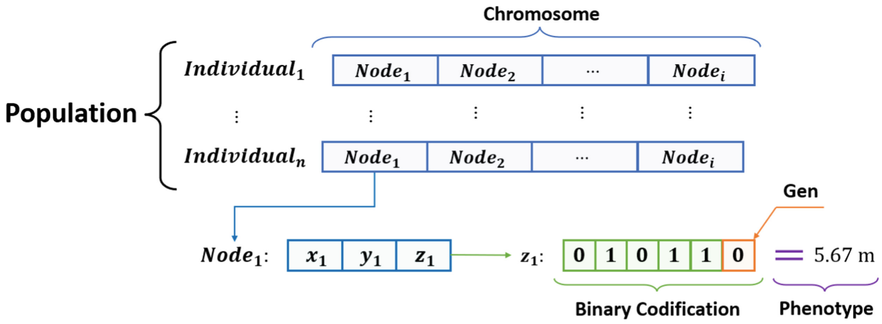


Fig. 1. Followed codification for the proposed NLP optimization.

4 Results

In order to validate the devised methodology, a simulation on an urban scenario is proposed. The designed urban scenario entails an airway where drones may fly, surrounded by buildings. This movement limitation for UAV follows other literature for communing human and UAV transit [7], as well as the Spanish legislation regarding urban transit for UAV.

The drone airway represents the TLE, depicted in Fig. 2, which is discretized into different points for further analysis. On the other hand, the NLE, where the architecture nodes can be deployed, has also been delimited, registering a node height between 5 and 6 m over the base terrain and buildings.

Through this scenario, the proposed methodology is analysed. All simulations were performed within the M programming software, through an Intel® i7 2.4 GHz of CPU and 16 GB of RAM. Table 1 indicates the selected parameters for the represented results.

Based on these parameters, multiple simulations with varying number of sensors were performed, attaining the results represented in Table 2.

Results show that both GA optimizations achieve a robust LPS for localizing drones in the proposed urban scenario. Furthermore, the inclusion of more architecture sensors into the node distribution achieves an overall lower localization uncertainty, and a more uniform and robust positioning system.

Moreover, the achieved localization uncertainty represents the first iteration of the localization algorithm. Subsequent localizations could result in a more robust positioning.

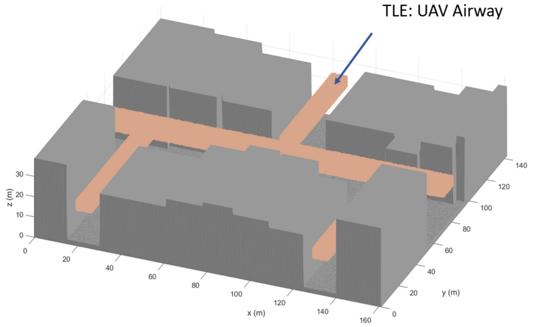


Fig. 2. Devised scenario for the NLP optimization.

Table 1. List of parameters for the represented simulations. The hyperparameters values were adjusted experimentally following the methodology devised in [31].

(a) TDOA Parameters.

Parameter	Value
Bandwidth	100 MHz
Receptor sensibility	−90 dBm
Frequency of emission	5465 MHz
Transmission power	1 W
Mean noise power	−94 dBm
Time from synchronization	1 μs
Clock frequency	1 GHz
Frequency-drift	U{-10,10} ppm
Initial-time offset	U{15,30} ns
LOS Path loss exponent	2.1
NLOS Path loss exponent	4.1

(b) GA Hyperparameters, following [31]

Hyperparameter	Value
Population	80
Selection Operator	Tournament 2
Crossover Operator	Multi-point
Mutation Operator	2%
Elitism Operator	7%
Stop Criteria	100 Generations or Convergence
Airway Height	$z_{UAV} \in [16, 20]$ meters
TLE Points	28701
NLE Points	549153
Number of combinations	$2.494 \cdot 10^{57}$

Table 2. Results of the achieved node distribution by the proposed methodology.

GA Optimizations	μ_{RMSE}	σ_{RMSE}	RMSE $< X$		
			$X = 3$	$X = 4$	$X = 5$
8 Node Distribution	3.31 m	0.51 m	26.6%	92.5 %	97.5%
10 Node Distribution	2.80 m	0.33 m	69.0 %	99.6 %	100 %

Finally, Fig. 3 shows the RMSE associated with each TLE point for the 10 nodes distribution. Most of the scenario points present a RMSE lower than 3 m, thus improving the uncertainties achieved by GNSS localization systems for UAVs, which attains mean localization uncertainties ranging from 4 to 8 m [32]. Furthermore, the attained distributions also improves the performance of unoptimized LPS, increasing the uniformity and reducing the localization uncertainty [33].

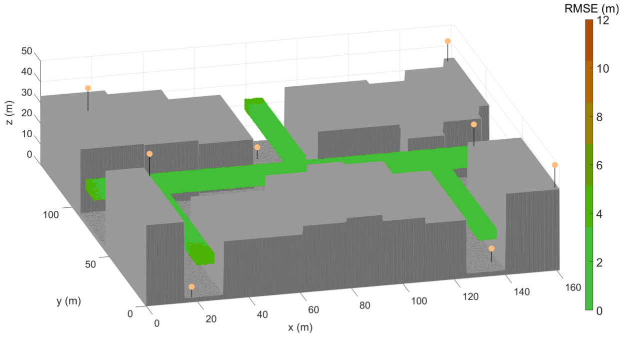


Fig. 3. RMSE distribution for the attained 10 sensors distribution.

5 Conclusions

The development of a robust localization system represents a cardinal step for the materialization of multiple autonomous tasks for drones, such as delivery services, especially in indoor and urban scenarios, where the reception of the GNSS signals is compromised.

LPS have been widely proposed throughout the literature for localizing drones in these scenarios, however, the adequate implementations of these systems requires the determination of the optimal positions of each of the nodes that constitutes the WSN. However, this problem has been defined as NP-Hard, and it is yet to be resolved for localizing UAV.

Therefore, in this paper, a GA optimization is proposed for obtaining a near optimal solution for the NLP while also considering the SSP into the optimization, thus improving the attained results. Both the architecture and the GA proposal have been specifically devised for obtaining a valid and robust LPS for localizing drones in urban scenarios.

Results confirm the viability of the proposed methodology, obtaining a uniform and a robust localization architecture, improving the localization accuracy from GNSS systems. Consequently, the proposed methodology constitutes a viable solution for the development of autonomous navigation in UAV, thus fulfilling the main objectives of this paper.

Acknowledgements. This work was supported by the Spanish Research Agency (AEI) under grant number PID2019-108277GB-C21/AEI/10.13039/501100011033.

References

1. Gupta, S.G., Ghonge, D., Jawandhiya, P.M., et al.: Review of unmanned aircraft system (UAS). Int. J. Adv. Res. Comput. Eng. Technol. **2** (2013)
2. Semsch, E., Jakob, M., Pavlicek, D., Pechoucek, M.: Autonomous UAV surveillance in complex urban environments. In: 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, vol. 2, pp. 82–85. IEEE (2009)

3. Nikolic, J., Burri, M., Rehder, J., Leutenegger, S., Huerzeler, C., Siegwart, R.: A UAV system for inspection of industrial facilities. In: 2013 IEEE Aerospace Conference, pp. 1–8. IEEE (2013)
4. Chiang, W.-C., Li, Y., Shang, J., Urban, T.L.: Impact of drone delivery on sustainability and cost: realizing the UAV potential through vehicle routing optimization. Appl. Energy **242**, 1164–1175 (2019)
5. Fernández-Caramés, T.M., Blanco-Novoa, O., Froiz-Míguez, I., Fraga-Lamas, P.: Towards an autonomous industry 4.0 warehouse: a UAV and blockchain-based system for inventory and traceability applications in big data-driven supply chain management. Sensors **19**(10), 2394 (2019)
6. Kim, J., Kim, S., Ju, C., Son, H.I.: Unmanned aerial vehicles in agriculture: a review of perspective of platform, control, and applications. IEEE Access **7**, 105100–105115 (2019)
7. Brunner, G., Szebedy, B., Tanner, S., Wattenhofer, R.: The urban last mile problem: autonomous drone delivery to your balcony. In: 2019 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 1005–1012. IEEE (2019)
8. Pavlenko, T., Schütz, M., Vossiek, M., Walter, T., Montenegro, S.: Wireless local positioning system for controlled UAV landing in GNSS-denied environment. In: 2019 IEEE 5th International Workshop on Metrology for AeroSpace (MetroAeroSpace), pp. 171–175. IEEE (2019)
9. Bijjahalli, S., Gardi, A., Sabatini, R.: GNSS performance modelling for positioning and navigation in urban environments. In: 2018 5th IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace), pp. 521–526. IEEE (2018)
10. Arifin, M., Nazaruddin, Y., Tamba, T., Santosa, R., Widyotriatmo, A.: Experimental modeling of a quadrotor UAV using an indoor local positioning system. In: 2018 5th International Conference on Electric Vehicular Technology (ICEVT), pp. 25–30. IEEE (2018)
11. Amar, A., Weiss, A.J.: Localization of narrowband radio emitters based on doppler frequency shifts. IEEE Trans. Signal Process. **56**(11), 5500–5508 (2008)
12. Wang, Y., Ma, X., Leus, G.: Robust time-based localization for asynchronous networks. IEEE Trans. Signal Process. **59**(9), 4397–4410 (2011)
13. Zhao, W., Panerati, J., Schoellig, A.P.: Learning-based bias correction for time difference of arrival ultra-wideband localization of resource-constrained mobile robots. IEEE Robot. Automat. Lett. **6**(2), 3639–3646 (2021)
14. Deng, Z., Tang, S., Deng, X., Yin, L., Liu, J.: A novel location source optimization algorithm for low anchor node density wireless sensor networks. Sensors **21**(5), 1890 (2021)
15. Khalaf-Allah, M.: Novel solutions to the three-anchor TOA-based three-dimensional positioning problem. Sensors **21**(21), 7325 (2021)
16. Álvarez, R., Díez-González, J., Alonso, E., Fernández-Robles, L., Castejón-Limas, M., Perez, H.: Accuracy analysis in sensor networks for asynchronous positioning methods. Sensors **19**(13), 3024 (2019)
17. Hu, J., et al.: A brief review on the positioning technologies for unmanned aerial vehicles. In: 2017 IEEE International Conference on Unmanned Systems (ICUS), pp. 527–532. IEEE (2017)
18. Nguyen, N.-T., Liu, B.-H.: The mobile sensor deployment problem and the target coverage problem in mobile wireless sensor networks are np-hard. IEEE Syst. J. **13**(2), 1312–1315 (2018)
19. Díez-González, J., et al.: Genetic algorithm approach to the 3d node localization in TDOA systems. Sensors **19**(18), 3880 (2019)

20. Rajakumar, R., Amudhavel, J., Dhavachelvan, P., Vengattaraman, T.: GWO-LPWSN: grey wolf optimization algorithm for node localization problem in wireless sensor networks. J. Comput. Netw. Commun. **2017** (2017)
21. Maheshwari, P., Sharma, A.K., Verma, K.: Energy efficient cluster based routing protocol for WSN using butterfly optimization algorithm and ant colony optimization. Ad Hoc Netw. **110**, 102317 (2021)
22. Kannan, A.A., Mao, G., Vucetic, B.: Simulated annealing based wireless sensor network localization. J. Comput. **1**(2), 15–22 (2006)
23. Annepu, V., Rajesh, A.: Implementation of self adaptive mutation factor and cross-over probability based differential evolution algorithm for node localization in wireless sensor networks. Evolution. Intell. **12**(3), 469–478 (2019)
24. Bajovic, D., Sinopoli, B., Xavier, J.: Sensor selection for event detection in wireless sensor networks. IEEE Trans. Signal Process. **59**(10), 4938–4953 (2011)
25. Vankayalapati, N., Kay, S., Ding, Q.: TDOA based direct positioning maximum likelihood estimator and the Cramer-Rao bound. IEEE Trans. Aerosp. Electron. Syst. **50**(3), 1616–1635 (2014)
26. Díez-González, J., Verde, P., Ferrero-Guillén, R., Álvarez, R., Pérez, H.: Hybrid memetic algorithm for the node location problem in local positioning systems. Sensors **20**(19), 5475 (2020)
27. Yang, B., Scheuing, J.: Cramer-Rao bound and optimum sensor array for source localization from time differences of arrival. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2005), vol. 4, pp. iv–961. IEEE (2005)
28. Kaune, R., Hörst, J., Koch, W.: Accuracy analysis for TDOA localization in sensor networks. In: 14th International Conference on Information Fusion, pp. 1–8. IEEE (2011)
29. Díez-González, J., et al.: Optimal node distribution in wireless sensor networks considering sensor selection. In: Sanjurjo González, H., et al. (eds.) SOCO 2021. AISC, vol. 1401, pp. 512–522. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-87869-6_49
30. Mirjalili, S.: Genetic algorithm. In: Evolutionary Algorithms and Neural Networks. SCI, vol. 780, pp. 43–55. Springer, Cham (2019). https://doi.org/10.1007/978-3-319-93025-1_4
31. Ferrero-Guillén, R., Díez-González, J., Álvarez, R., Pérez, H.: Analysis of the genetic algorithm operators for the node location problem in local positioning systems. In: de la Cal, E.A., Villar Flecha, J.R., Quintián, H., Corchado, E. (eds.) HAIS 2020. LNCS (LNAI), vol. 12344, pp. 273–283. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-61705-9_23
32. Wang, L., Groves, P.D., Ziebart, M.K.: GNSS shadow matching: improving urban positioning accuracy using a 3D city model with optimized visibility scoring scheme. NAVIGATION, J. Inst. Navigat. **60**(3), 195–207 (2013)
33. Bouloukou, M., Masiero, A., Vettore, A., Gikas, V.: UAV UWB positioning close to building facades: a case study. Int. Archiv. Photogram. Remote Sens. Spatial Inf. Sci. **43**, 97–102 (2021)

Applying Deep Q-learning for Multi-agent Cooperative-Competitive Environments

Anikó Kopacz(✉), Lehel Csató, and Camelia Chira

Faculty of Mathematics and Computer Science, Babeş-Bolyai University,
Cluj-Napoca, Romania
{aniko.kopacz,lehel.csato,camelia.chira}@ubbcluj.ro

Abstract. Cooperative-competitive social group dynamics may be modelled with multi-agent environments with a large number of agents from a few distinct agent-types. Even the simplest games modelling social interactions are suitable to analyze emerging group dynamics. In many cases, the underlying computational problem is NP-complex, thus various machine learning techniques are implemented to accelerate the optimization process. Multi-agent reinforcement learning provides an effective framework to train autonomous agents with an adaptive nature. We analyze the performance of centralized and decentralized training featuring Deep Q-Networks on cooperative-competitive environments introduced in the MAgent library. Our experiments demonstrate that sensible policies may be constructed utilizing centralized and decentralized reinforcement learning methods by observing the mean rewards accumulated during training episodes.

1 Introduction

Most multi-agent systems [3] can be modelled as networks, where the agents are going to be considered nodes of the networks and the various relationships between the agents are represented by edges within the networks [10]. Temporal networks are graphs having a fixed set of nodes and dynamically changing edges, that are available only for a given period of time [19]. The dynamic nature of temporal networks allows the network topology to change over a given period. Temporal networks have a vast range of applications due to the dynamic characteristics of real-world problems, such as traffic management problems [14], epidemiology [23], modeling and analyzing interactions within social groups [11]. The efficiency of planning tasks concerning temporal networks is NP-hard [7]. In order to reduce the required computational resources for planning tasks featuring temporal networks is to apply machine learning methods, for instance reinforcement learning algorithms.

Multi-agent reinforcement learning has several areas of application to real-world problems, for example: fake news detection in social networks [1], traffic simulation [5], multi-agent navigation problem [21] and various cooperative game play scenarios [12,29]. Several data analysis approaches could be extended to be

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 626–634, 2023.
https://doi.org/10.1007/978-3-031-18050-7_61

able to operate on multi-agent systems, however computational trade-offs shall be considered. Introducing several actors to a non-trivial system can significantly increase the demand for computational resources, rendering some methods infeasible in practice. Thus, various heuristics might get favored over exact solutions in the case of NP-hard problems regarding multi-agent systems, e.g. [8,28,31].

Many real-life inspired problems require the analysis of social interactions and group dynamics on a global level. In cases when the available input data is limited or significantly hard to process regarding the examined social interactions, applying descriptive computational simulations is beneficial. In this paper, experiments were conducted on cooperative-competitive multi-agent scenarios applying deep reinforcement learning techniques. We analyze the agent policies trained with sequences of agent-environment interactions by observing the mean episode reward accumulated by selected agent groups.

This paper is organized as follows. First, Sect. 2 presents cooperative-competitive multi-agent games. Then, Sect. 3 presents the centralized and decentralized multi-agent reinforcement learning approaches used later. Section 4 describes our experiments on cooperative-competitive multi-agent environments. Lastly, our conclusions and suggestions for further improvements are formulated in Sect. 5.

2 Cooperative-Competitive Multi-agent Environments

Multi-agent planning systems have multiple real-life applications, such as fake news detection [1], tracking evolutionary phenomena [2], collision avoidance in multi-robot systems [4] and analyzing social networks [6], to name a few. Multi-agent environments, where a group of agents has a shared goal to achieve, are referred to as cooperative settings. Cooperative-competitive environments are formed by multiple cooperating groups having objectives opposing other agent-groups. Cooperative-competitive environment settings are observable in cases of social networks and evolutionary studies. Multi-agent reinforcement learning is suitable for simulating and deconstructing social interactions [13].

In [13] the authors analyzed the performance of multi-agent reinforcement learning methods on sequential social dilemmas. Machine learning methods that perform efficiently on populations modelling complex relationships and diverse interactions schemes, can be further refined to be suitable for analyzing dynamics emerging in social networks. In this article, we study the impact of agent policies on achieving shared agent objectives and the evolution of biological populations. The main advantages of studying population-based cooperative-competitive games include the understanding of group dynamics, the identification of underlying group characteristics, the prediction of social behavior for conflict de-escalation, allocating sufficient resources, and favoring collaborative intelligence in corporate settings.

In practice, the social multi-agent systems have a complex structure and the models incorporate multiple independent variables. To disseminate the core problem and concentrate solely on the social interaction component, the analysis of simpler cooperative-competitive environments is beneficial. The methods efficient for cooperative-competitive games are suitable to adopt in real life social network analysis and prediction of group dynamics.

Recent contributions focus on implementing various cooperative-competitive environments in order to provide benchmarks for multi-agent reinforcement learning methods, e.g., [16,27,29]. The PettingZoo library [27] introduced a unified API for several multi-agent reinforcement learning scenarios with the goal to advance evaluation of multi-agent reinforcement learning approaches on various problems. The robustness of state-of-the-art machine learning methods consists of the possibility to adapt for different real-life inspired problems.

The PettingZoo library integrates various multi-agent games that are suitable for modelling more complex multi-agent relationships of real-life problems, such as discrete and continuous control tasks [9], Multi Particle Environments [18], MAgent platform [34]. The MAgent library was first introduced in [34] designed to implement large multi-agent environments. The incorporated problems reflect techniques of cooperation and competition with a focus on predator-prey type of dynamics.

In order to address large-scale training in the case of multi-agent reinforcement learning, value-function decomposition methods [22,24] could be utilized during the agent action selection process. Another approach to decrease the complexity of the multi-agent environment is mean field multi-agent reinforcement learning [32]. The performance of the mean field reinforcement learning is evaluated on multi-agent environments featuring similar agents. For example, the game *Battle* incorporated in the MAgent library is utilized as a benchmark to compare the efficiency of mean field theory combined with multi-agent reinforcement learning to state-of-the-art multi agent reinforcement learning approaches in [33].

3 Single and Multi-agent Reinforcement Learning

Reinforcement learning utilizes previous experiences of one or more agents interacting with their environment to acquire desired skill sets or knowledge to solve predefined tasks [25]. Multi-agent reinforcement learning extends the single agent reinforcement learning setting and models problems, where multiple separate autonomous entities operate in the same environment in a parallel manner [20]. While several single-agent reinforcement learning approaches could be applied to the multi-agent setting, the complexity of the solution space grows exponentially with the number of agents. The efficiency of multi-agent reinforcement learning approaches may be increased with several techniques, such as the sharing experiences with other agents during the training process [26].

Multiple reinforcement learning settings can be distinguished based on the task specific agent goals. If the agents' goal is to achieve cooperation and to optimize a global reward function through their actions, then the analyzed reinforcement learning task is cooperative. On the other hand, competitive multi-agent environments multiple agents coexist in the same environment, but compete for the same resources or have opposite goals. If the agents have assigned groups in a way that agents within a group have the same goals, but the different formations have opposite goals, we characterize the environment as a cooperative-competitive multi-agent environment. In the latter case, the agents need to learn to collaborate with each other, but compete with their opponents or collectively aim to solve a particular task in shorter time than their opponents.

Approaches that manage tasks associated with multi-agent scenarios can be split into two categories regarding the way they manage the available information: centralized and decentralized policy execution [30]. Centralized approaches tend to integrate a global view of the environment, where every information gathered by the agents is available as an input for the learning algorithm and consequently the selection of agent actions is optimized jointly. On the contrary, in the decentralized setting agents operate in a parallel manner. Decentralized agents do not access information gathered by other agents and try to optimize their actions separately based on the available reward signal. Considering multi-agent environments, the available reward signal can be global, so a single reward value incorporates the goodness of a decision. On the other hand, the reward can be defined in a distributed manner, so a separate reward signal is available for every agent. Even if a local reward value is available for every agent, a global reward value can be calculated by aggregating the distinct reward signals.

One of the main advantages of decentralized execution over a centralized one is scalability, since the number of necessary learning samples for centralized approaches grow exponentially with the number of agents. Decentralized methods provide a more robust framework in regards to the increase in the number of agents, because of the fact that agents choose their actions separately given a particular state of the environment. At the same time, if the agents have only a global reward signal, in the case of using decentralized approaches, the problem of decomposing the global reward signal arises. The available global reward signal is preferred to be decomposed in order to estimate each agent actions' attribution to the current state of the environment.

The action-value function denoted by Q for a state-action pair measures the goodness of choosing that action given over any other available action given the current state of the agent. As a result, we consider the optimal action at a given state to be the one that has the maximal associated Q-value. Q-learning estimates the action-value function of the agent. [17] introduced Deep Q-Networks (DQN) in [17], a modified Q-learning method combined with a deep learning architecture to determine the optimal policy of a single agent.

Monotonic value function factorization for deep reinforcement learning (QMIX) [22] utilizes the global reward signal for training. QMIX factorizes the joint action-value function (Q_{total}) into separate components corresponding to each agent. For every a agent a separate utility function is constructed denoted by Q_a and the agent acts greedily maximizing their own utility function. Given the fact that Q_a network aims to approximate the overall goodness of state-action pairs, the agent maximizing the utility function is choosing the best possible outcome given a particular state.

QMIX applies a compound structure of multiple neural network architectures during training and evaluation: (i) agent neural networks, that aim to learn to approximate the Q_a function and (ii) a mixing network that synthesizes the observation-action histories and action pairs available for every agent from the environment. For every agent, a Deep Q-Network approach is implemented using deep recurrent network to learn from a larger sequence of observation and action history.

4 Methods and Experiments

In this paper, we aim to investigate an approach based on Independent Learning [26] applied with DQN [17] and the QMIX approach [22] in context of cooperative-competitive environments. The training and evaluation of reinforcement learning algorithms builds upon the RLlib Python library [15], which implements several reinforcement learning methods.

DQN trains a single neural network for selecting a single agent's optimal action for any given environmental state. A straightforward approach for extending to the multi-agent scenario using the same neural network for each agent's action selection [26]. The obtained system is trained in a decentralized manner since the outcome of an agent action is considered independent from other agent actions taken at the same time step. QMIX ensembles multiple Q-Networks to learn the joint optimal action from the episodes. The reward signal is decomposed and the impact of other agents is factored into the decision. While QMIX requires higher computational resources for training compared to DQN, QMIX can incorporate complex relationship between independent agent actions into the decision-making process, thus outperforming decentralized trainers.

4.1 Proposed Approach

In this article, we compare two deep reinforcement learning methods. On one hand, we experimented with Independent Learning [26] combined with DQN [17] for creating a baseline for evaluation. Due to applying Independent Learning, the multi-agent scenario is treated as a single agent reinforcement learning problem, rendering other agent actions and objectives as part of the environment. The learning process is decentralized; however, agents share experiences with each other. On the other hand, a QMIX trainer is evaluated for learning to optimize agent actions. The centralized trainer receives the sum of the agent rewards grouped by the types of the agents.

We trained two separate policies for selecting actions: a separate policy is trained to select actions for predator- and prey-type of agents, respectively. The two separate policies are initialized with the same parameters, however during the training phase the agents operate following their predefined policy. Thus, agents with similar goals share experiences, but they do not weigh in the optimization of the rewards attributed to adversarial agents.

4.2 Experimental Setup

The experiments were conducted on training on the *Adversarial pursuit* environment from MAgent provided by the PettingZoo library [27]. MAgent environments implement large scale multi-agent interactions that have cooperative-competitive nature [34]. The *Adversarial pursuit* setting consists of multiple agents that belong to one of two distinct types with opposite goals: *predators* and *preys* (see Fig. 1).

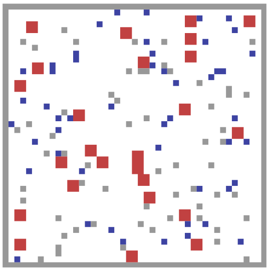


Fig. 1. Example of the *adversarial_pursuit_v3* environment, red squares representing predators and blue ones denoting preys.

Predators are rewarded whenever they tag any prey, thus the predators' main goal is to locate the nearby prey and then trap them. On the other hand, prey-agents are encouraged to avoid getting tagged and receive a negative reward when any of the predators managed to tag them. The agents operate in a grid-world environment being represented by particles themselves. Predators occupy more space on the map and are slower than their prey. Consequently, a beneficial technique for the predators is cooperating with other predators and trapping prey to catch them.

During the training of the agents, the maximum length of episodes is set to 512, after reaching the maximum number of time steps, the episode restarts from an initial state. The discount factor used in calculating the discounted reward for the group policies is set to 0.85.

4.3 Discussion

Figure 2 shows the mean reward calculated for each episode of the training, the accumulated rewards are summed over all agents having the same type and the rewards received by the two different agent groups are reported separately. The policies trained with the DQN method outperform the QMIX value factorization method in for the predator-type agents, as well for the preys.

Centralized learning in multi-agent reinforcement learning is a complex problem. On one hand, independent learning does not consider the dependence of agent actions on other agents' policies. On the other hand, the centralization

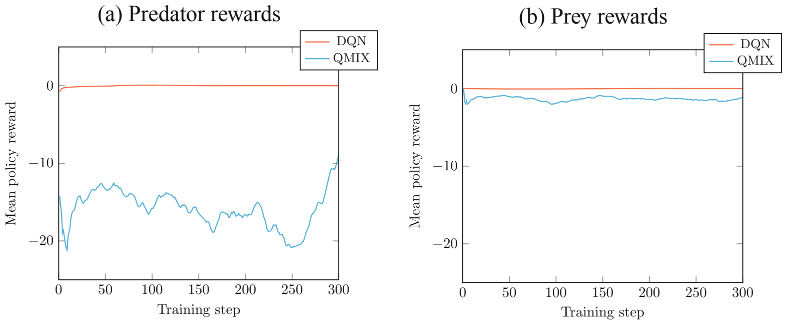


Fig. 2. Mean episode reward during training a DQN and QMIX learner shared between agents evaluated in the *adversarial_pursuit_v3* environment.

of the learning process comes with increases the demand for training roll-outs in order to sufficiently explore the solution space. Aggregating similar agents into a group may be beneficial concerning the maximization of the accumulated reward, however, the construction of such groups is non-trivial. Centralization of the optimization problem is difficult in multi-agent cooperative-competitive environment, where the number of agents is high and the agents may belong to multiple distinct types of groups.

Even in cases where the environment features only one type of agent, the problem of distribution of agents into groups arises. To balance the individual agent objectives and the objectives of a selected group, we constructed groups from agents that have similar goals. Thus, the mean reward for the group reflects the success of the individuals for completing their predefined goal.

5 Conclusions and Future Work

Multi-agent reinforcement learning provides a framework for autonomous agents to extract meaningful information from previous experiences of agent-environment interaction. Various predator-prey like dynamics could be modelled by adequate reinforcement learning setups creating instances of cooperative-competitive environments. In the case of large-scale environments featuring multiple types of agents with opposite end-goals, the decomposition of the joint reward signal may be beneficial in order to better align with the competitive nature of the environment. In this paper, we applied deep reinforcement learning methods for cooperative-competitive test environments. We compared independent Deep Q-learning with Monotonic value factorization with deep learning and concluded that independent learning yields a higher episode mean reward. Further improvements include testing multi-agent reinforcement learning algorithms on various social networks and analyzing social dynamics in complex systems.

References

1. Aymanns, C., Foerster, J., Georg, C.-P.: Fake news in social networks. SSRN Electr. J. Paper No. 2018/4 (2017)
2. Bloembergen, D., Tuyls, K., Hennes, D., Kaisers, M.: Evolutionary dynamics of multi-agent learning: a survey. J. Artif. Intell. Res. **53**, 659–697 (2015)
3. Brooks, R.A.: Intelligence without representation. Artif. Intell. **47**(1), 139–159 (1991)
4. Cai, C., Yang, C., Zhu, Q., Liang, Y.: Collision avoidance in multi-robot systems. In: 2007 International Conference on Mechatronics and Automation, pp. 2795–2800 (2007)
5. Cai, P., Lee, Y., Luo, Y., Hsu, D.: Summit: A simulator for urban driving in massive mixed traffic. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 4023–4029. IEEE (2020)
6. Carley, K., Martin, M., Hirshman, B.: The etiology of social change. Top. Cogn. Sci. **1**, 621–650 (2009)
7. Dechter, R., Meiri, I., Pearl, J.: Temporal constraint networks. Artif. Intell. **49**(1), 61–95 (1991)
8. dos Santos, D.S., Bazzan, A.L.: Distributed clustering for group formation and task allocation in multiagent systems: a swarm intelligence approach. Appl. Soft Comput. **12**(8), 2123–2131 (2012)
9. Gupta, J.K., Egorov, M., Kochenderfer, M.: Cooperative multi-agent control using deep reinforcement learning. In: Sukthankar, G., Rodriguez-Aguilar, J.A. (eds.) AAMAS 2017. LNCS (LNAI), vol. 10642, pp. 66–83. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-71682-4_5
10. Herrera, M., Pérez-Hernández, M., Kumar Parlikad, A., Izquierdo, J.: Multi-agent systems and complex networks: review and applications in systems engineering. Processes **8**(3), 312 (2020)
11. Kossinets, G., Watts, D.J.: Empirical analysis of an evolving social network. Science **311**, 88–90 (2006)
12. Leibo, J. Z., et al.: Scalable evaluation of multi-agent reinforcement learning with melting pot. In: International Conference on Machine Learning, pp. 6187–6199. PMLR (2021)
13. Leibo, J.Z., Zambaldi, V., Lanctot, M., Marecki, J., Graepel, T.: Multi-agent reinforcement learning in sequential social dilemmas. In: Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS '17, pp. 464–473 (2017)
14. Li, J.-Q., Mirchandani, P.B., Borenstein, D.: The vehicle rescheduling problem: model and algorithms. Networks **50**(3), 211–229 (2007)
15. Liang, E., et al.: Ray RLlib: A composable and scalable reinforcement learning library. arXiv preprint arXiv:abs/1712.09381 (2017)
16. Liu, S., Lever, G., Merel, J., Tunyasuvunakool, S., Heess, N., Graepel, T.: Emergent coordination through competition. arXiv preprint arXiv:abs/1902.07151 (2019)
17. Mnih, V., et al.: Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602 (2013)
18. Mordatch, I., Abbeel, P.: Emergence of grounded compositional language in multi-agent populations. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
19. Nicosia, V., Tang, J., Musolesi, M., Russo, G., Mascolo, C., Latora, V.: Components in time-varying graphs. Chaos: Interdisc. J. Nonlinear. Sci. **22**(2), 023101 (2012)

20. Nowé, A., Vrancx, P., Hauwere, Y.-M.D.: Game theory and multi-agent reinforcement learning. In: Wiering, M., van Otterlo, M. (eds) Reinforcement Learning. Adaptation, Learning, and Optimization, pp. 441–470. Springer, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-27645-3_14
21. Papoudakis, G., Christianos, F., Schäfer, L., Albrecht, S.V.: Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks. In: Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS) (2021)
22. Rashid, T., Samvelyan, M., Witt, C.S.D., Farquhar, G., Foerster, J.N., Whiteson, S.: Monotonic value function factorisation for deep multi-agent reinforcement learning. J. Mach. Learn. Res. **21**, 178:1–178:51 (2020)
23. Stehlé, J.: Simulation of an SEIR infectious disease model on the dynamic contact network of conference attendees. BMC Med. **9**(1), 1–15 (2011)
24. Sunehag, P., et al.: Value-decomposition networks for cooperative multi-agent learning based on team reward. In: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '18, pp. 2085–2087 (2018)
25. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction, 2nd edn. The MIT Press (2018)
26. Tan, M.: Multi-agent reinforcement learning: Independent vs. cooperative agents. In: Proceedings of the Tenth International Conference on Machine Learning, pp. 330–337. Morgan Kaufmann (1993)
27. Terry, J.K., et al.: PettingZoo: Gym for multi-agent reinforcement learning. arXiv preprint arXiv:2009.14471 (2020)
28. Törnquist, J., Persson, J., et al.: Train traffic deviation handling using tabu search and simulated annealing. In: Proceedings of the 38th Hawaii International Conference on System Sciences, pp. 1–10 (2005)
29. Vinyals, O., et al.: Starcraft II: A new challenge for reinforcement learning. arXiv preprint arXiv:abs/1708.04782 (2017)
30. Vrancx, P., Verbeeck, K., Nowe, A.: Decentralized learning in Markov games. Part B, Cybern.: a publication of the IEEE Syst., Man, Cybern. Soc. **38**, 976–981 (2008)
31. Wang, L., Wang, Z., Hu, S., Liu, L.: Ant colony optimization for task allocation in multi-agent systems. China Commun. **10**(3), 125–132 (2013)
32. Yang, Y., Luo, R., Li, M., Zhou, M., Zhang, W., Wang, J.: Mean field multi-agent reinforcement learning. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning, vol. 80 of Proceedings of Machine Learning Research, pp. 5571–5580. PMLR (2018)
33. Zhang, T., Ye, Q., Bian, J., Xie, G., Liu, T.-Y.: MFVFD: a multi-agent Q-learning approach to cooperative and non-cooperative tasks. In: Zhou, Z.-H. (ed.) Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, pp. 500–506 (2021)
34. Zheng, L., et al.: MAgent: A many-agent reinforcement learning platform for artificial collective intelligence. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)

A Comparison of Two Speech Emotion Recognition Algorithms: Pepper Humanoid Versus Bag of Models

Enrique de la Cal[1](✉), Javier Sedano[2], Alberto Gallucci[1], and Paloma Valderde[2]

[1] Computer Science Department, University of Oviedo, Oviedo, Spain
delacal@uniovi.es
[2] ITCL, Burgos, Spain
{javier.sedano,paloma.valverde}@itcl.es

Abstract. Some of the most exciting applications of Speech Emotion Recognition (SER) focus on gathering emotions in daily life contexts, such as social robotics, voice assistants, entertainment industries, and health support systems. Among the most popular social humanoids launched in the last years, Softbank Pepper® can be remarked. This humanoid sports an exciting multi-modal emotional module, including face gesture recognition and Speech Emotion Recognition. On the other hand, a competitive SER algorithm for embedded systems [2] based on a bag of models (BoM) method was presented in previous works. As Pepper is an exciting and extensible platform, current work represents the first step to a series of future social robotics projects. Specifically, this paper systematically compared Pepper's SER module (SER-Pepper) against a new release of our SER algorithm based on a BoM of XTraTress and CatBoost (SER-BoM). A complete workbench to achieve a fair comparison has been deployed, including other issues: selecting two well-known SER datasets, SAVEE and RAVNESS, and a standardised playing and recording environment for the files of the former datasets. The SER-BoM algorithm has shown better results in all the validation contexts.

1 Introduction and Motivation

Speech Emotion Recognition (SER) is a subtopic of the research field of Affective Computing that includes the techniques devoted to identifying and understanding human emotions, with applications like social robotics, voice assistants, entertainment industries, and health support systems.

Among the most popular social humanoids launched in the last years, it can be remarked the Softbank Pepper® model. This humanoid sports an exciting multi-modal emotional module, including face gesture recognition and Speech Emotion Recognition. On the other hand, in previous works, a competitive SER algorithm for embedded systems [2] based on a bag of models (BoM) method was

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 635–644, 2023.
https://doi.org/10.1007/978-3-031-18050-7_62

presented. Specifically, this paper has carried out a systematic comparison of the Pepper's SER module (SER-Pepper) against a new release of our SER algorithm based on a BoM of XTraTress and CatBoost (SER-BoM). A complete workbench to achieve a fair comparison has been deployed, including other issues: selecting two well-known SER datasets, SAVEE and RAVNESS, and a standardised playing and recording environment for the files of the former datasets.

The sections of this work are arranged this way. Section 2 includes a detailed description of the SER systems to be compared, including the Pepper SER subsystem, as well as a new proposal of our SER algorithm for embedded systems. The following section will define a fair workbench to compare both algorithms and the numerical results section. Finally, conclusions and future work are depicted.

2 The SER Subsystems

2.1 Description of the Pepper and Its SER Subsystem

Pepper is a social humanoid robot measuring 120 cm in height developed by Softbank Robotics and launched in June 2014 [12]. This robot is equipped with a large panel of sensors and actuators like six-axes accelerometers, cameras and 3D sensors, tactile sensors, laser sensing, and microphones (see Fig. 1). However, from the sensory point of view, this work is focused on the microphones set (see Table 1).

Several software pieces were developed to facilitate the necessary perception abilities and ensure a smooth HRI, including the capacity to recognise and respond to human emotions through a multimodal set of emotional skills: face gesture recognition and Speech Emotion Recognition. ALVoiceEmotionAnalysis identifies the emotion expressed by the speaker's voice, independently of what is being said.

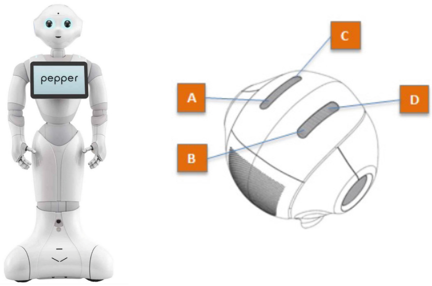


Fig. 1. Softbank Pepper and its microphones set

Table 1. Pepper microphones specifications

Parameter	Detail
Microphones location	x4 on the head
Sensitivity	300 mV/Pa ± 3 dB at 1 kHz
Frequency range	300 Hz to 12 kHz (−10 dB relative to 1 kHz)

2.1.1 The SER-Pepper Module

ALVoiceEmotionAnalysis module [3] relies on the ST Emotion Voice Analysis (EVA) technology by AGI. EVA analyses the fundamental frequency of a human voice taken from continuous natural speech. This system detects robust fundamental frequencies and intonations by parameterising them into pitch, power, and power deviation. Based on these parameters, data were classified via decision-tree logic into anger, joy, sorrow, and calmness. In addition, the degree of excitement was also extracted [11].

Third parties evaluated the system by matching the system performance to human subjective classification for each element. Results obtained by third parties sported an overall matching rate of 70%, and the matching rate was 86% compared to the subjects' assessment of their own voices.

The module included also depends on ALSpeechRecognition for speech detection, so ALSpeechRecognition (or ALDialog) needs to be started correctly for it to work.

At the end of the utterance, the engine processes the emotion levels and raises the "ALVoiceEmotionAnalysis/EmotionRecognized" event, which outputs four values:

- Matched emotion index: it is the index of the dominant emotion selected by the engine in the emotion levels vector [0 = unknown, 1 = calm, 2 = anger, 3 = joy, 4 = sorrow].
- Matched emotion level: it is the level of the dominant emotion.
- Emotion levels: it is the vector of the emotions' scores. Scores are integers between 0 and 10.
- Excitement level: it is a separate value that measures the amount of excitement in the voice. High excitement is often linked with joy or anger.

For example, [[3, 5], [0, 2, 3, 5, 0], 1] represent the matched emotion is joy with a level of 5. We can find this value in the emotion levels vector as well. Excitement is quite low (1), meaning the voice is quiet monotone.

2.2 Descripcion of the SER-BOM Algorithm

The SER algorithm designed by the authors was presented in [2], and it is based on a Bag of Models paradigm [13] with two base models (SER-BoM): XtraTrees [5] and Deep Learning pre-trained network from Resnet18 [9]. The winner technique has been selected as the voting algorithm. In order to reduce the computing weight as well as the complexity of the SER-Pepper algorithm, we have replaced the DeepLearning model with CatBoost [4] algorithm. Algorithm 1 details the stages of SER-BoM for one generic speech dataset D composed of NF files.

2.2.1 Preprocessing and Feature Computing

The sampling rate of all the dataset D files has been set to 16 kHz, and the stereo channels have been unified in mono. Moreover, a trim of the head and tail of the file has been carried out considering a valid signal 10 dB over-passing the background noise (25 dB + 10 dB = 35 dB).

Just segmental transformations have been considered, avoiding spectrographic ones. Thus, two groups of segmental features were selected: a group of prosodic (P) ones to capture the rhythm and a group of spectral (S) to capture the frequency: Root-mean-square (P) [1], Mel-Frequency Cepstral Coefficients (S), Chroma_stft (S), Spectral_centroid (S), Spectral_bandwidth (S), Spectral_rolloff (S): Compute roll-off frequency (S) and Zero_crossing_rate (S).

The original SER-BoM was designed to be deployed in a Rasberry PI (RPI) as part of the Alexa-Emotions device. Therefore, the base models must be trained with any validation technique such as cross-validation or LOO.

Algorithm 1. SER-BoM(D: Dataset, NF: Number of Files in D, WS: Window Size, WO: Window Overlapping, Fet: Features, FP: Features Parameters, NE: Number of Emotions, NP: Number of Participants, SR: SamplingRate)

```
Step1: Preprocessing & feature computing
W ← []
for f in 1:NF do
    D[f]← SetStandardSamplingRate(D[f], SR)
    D[f]← ConvertToMono(D[f])
    D[f]← Normalization(D[f])
    W← W + WindowFraming(D[f], WS, WO)
end for
for w in 1:Size(W) do
    for ft in 1:Size(Features) do
        FEATURES[w, ft] ← ComputeFeature(W, ft, FP[ft])
    end for
end for
Step2: Base Models Training
BaseModel1←Training_XTraTrees(Fet)
BaseModel2←Training_CatBoost(Fet)
Step3: Model running
EMOTION ← BaggingModelDeployment(BaseModel1, BaseModel2, Fet)
```

3 Materials and Methods

3.1 SER Datasets

In order to validate the comparison presented in this work two data publicy datasets have been selected: SAVEE [6–8] and RAVDESS [10] (see Table 2).

Table 2. Summary of the speech emotion dataset details. In bold the selected emotions.

Dataset	Participants	Language	Type	Emotions	#Utterances
SAVEE	4 actors (male gender)	English	Simulated	**anger**, disgust, fear, **happiness**, **sadness**, surprise and neutral (7)	480
RAVDESS	12 actresses and 12 actors (mixed gender)	English	Elicited	Neutral, **calm**, **anger**, **joy**, **sadness**, fearful, surprise, disgust and boredom (9)	1440

Table 3 includes in boldface the matching of the emotions considered in the studied dataset versus the emotions labels that the SER-Pepper module outputs.

Table 3. SER Datasets vs SER-Pepper emotions labels matching

Emotion	Unknown	Neutral	Calm	Anger	Joy	Sadness	Disgust	Fear	Surprise	Boredom
SER-Pepper	Yes	No	**Yes**	**Yes**	**Yes**	**Yes**	No	No	No	No
SAVEE	–	Yes	No	**Yes**	**Yes**	**Yes**	Yes	Yes	Yes	No
RAVNESS	–	Yes	**Yes**	**Yes**	**Yes**	**Yes**	Yes	Yes	Yes	Yes

3.2 Design of the Testing Workbench

Table 4 shows the workbench selected for the comparison carried out in this work. Because of logistic issues, the tests for both algorithms were carried out in different locations and with different auxiliary devices (speakers, microphones and player computer). In both cases, the speakers were located at 30 cm from the microphones (see Fig. 2) in a room space with background noise (BN) of a maximum of 20 dB in both cases. Nevertheless, in the case of the Pepper layout, we have stated that after switching on the robot, the fanning system raises the background noise to 25.5 dB.

3.2.1 Minimum dB Needed

Other important parameter of the testing workbench is the minimum of dB to detect voice activity: i) in the case of the SER-Pepper system, it needed a positive difference of +3 dB respect to the BN, and ii) in the case of the SER-BoM, the sound-capture function needs a minimum of 5 dB respect to the BN.

Table 4. Testing workbench issues

SER Algo.	Speakers	Microphone	Distance	Back. Noise	Final volumes
SER-Pepper	Polk Audio Silicon Graphics	Pepper Micros, 300 Hz–12 KHz	30 cm	18.8 dB/25.5 dB	40 dB, 55 dB, 70 dB
SER-BoM	Acer N17C4 Laptop internal speakers	Jetty Blue, (20 Hz–20 kHz)	30 cm	20 dB	40 dB, 55 dB, 70 dB

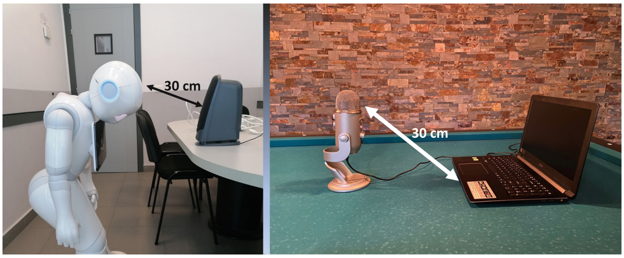


Fig. 2. Left) SER-Pepper recording layout, Right) SER-BoM recording layout

3.2.2 Volume Calibration

In order to check the correlation between the sound volume and the accuracy of the SER algorithms, a preliminary test was carried out by playing one file per actor, emotion and dataset (15files for SAVEE and 150 for RAVNESS) using volumes from −40dBFS to −10dBFS with a step of +5dBFS. Very preliminary tests showed that SER-Pepper is more sensitive to volume playing; these tests were carried out just with the SER-Pepper system. The volume −10 dB was removed from the results since it saturated the playing. Figure 3 shows that volumes −40dBFS, −25dBFS and −15dBFS set the best SER-Pepper performance at least for one of both datasets.

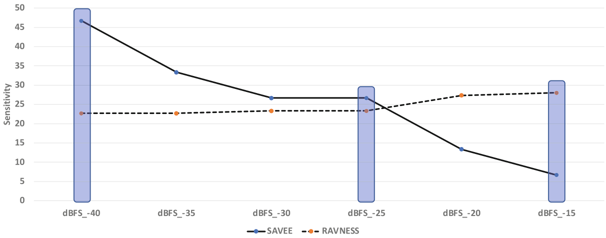


Fig. 3. Preliminary analysis of optimal volume for SER-Pepper algorithm

3.2.3 Datasets Playing and SER Output Processing

Attending the parameters defined in the previous section, the SER algorithms were tested in the following way:

- SER-Pepper: both datasets files were played[1] using the layout in Fig. 2 for 10 times at the selected 3 volumes, and the SER-Pepper output was recorded. The SAVEE dataset was played completely, but because of time restrictions, the RAVNESS dataset was played just for the first 10 actors out of 24. The predicted emotion was calculated taking the ten recorded outputs of SER-Pepper ($Output[0..9]$) for each played $file$:
  1. If (There is simple majority of wins) PredictedEmotion = WinnerEmotion
  2. else PredictedEmotion = $emo\_i/Max_{i=emo1}^{emon}(MeanLevelWinner(i, Output))$

  where $MeanLevelWinner(i, Output)$ calculated of mean level (Output.Level) of the wins of emotion i.
- SER-BoM: Each file was played and recorded just once at the selected 3 volumes using the layout described. After that the SER-BoM algorithm was launched in different configurations that will be described in Sect. 4.

4 Numerical Results

4.1 Results for SER-Pepper

SER-Pepper module was launched according to the conditions described in the workbench section obtaining the results of Table 5. It can be stated that its performance is really poor on most emotions, but in Sadness emotion for SAVEE dataset. The explanation of NA in columns Unknown and Calm is that: i) since SER-Pepper can outputs Unknown emotion, most of times don't do it, and ii) SAVEE dataset does't include Calm emotion. On the other side, the confusion matrices for both datasets (See Fig. 4) show that, although the Sorrow/Sadness emotion gets the better results in the SAVEE dataset, this emotion also represents an important False Positive number for other emotions like Joy and Anger in both datasets and Calm in the case of RAVNESS.

4.2 Results for SER-BoM

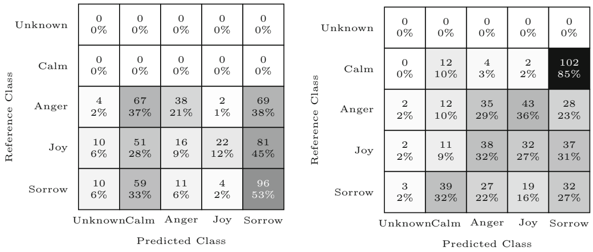
In order to get a fair comparison of both SER algorithms, the selected datasets were recorded using the layout described in Sect. 4. In addition, the recorded datasets were considered in two modalities, normalised and not normalised, so that we will launch SER-BoM[2] with three releases of each dataset: the Original (Not Normalised) and the Recorded (as it) and Normalised (Nor). Finally, two validation strategies were used: 10kfoldCV sample-centred and user-centred LOO, to get a scenario of experimentation similar to SER-Pepper.

[1] Pepper NAO operating system doesn't allow launching the SER-Pepper module fed with files stored inside Pepper.

[2] The hyperparameters for CatBoost and ExtraTree were {iterations = 10000, learning_rate = 0.04} and {n_estimators = 20000} respectively.

Table 5. Summary of performance of SER-Pepper module for SAVEE and RAVNESS datasets

	Unknown	Calm	Anger	Happy	Sad
SAVEE Sens.	NA	NA	0.21111	0.12222	**0.5333**
SAVEE Spec.	0.95556	0.6722	0.92500	0.98333	**0.5833**
RAVNESS Sens.	NA	0.1000	0.29167	0.26667	0.26667
RAVNESS Spec.	0.98542	0.8278	0.80833	0.82222	0.53611


Reference Class
Predicted Class

	Unknown	Calm	Anger	Joy	Sorrow
Unknown	0 0%	0 0%	0 0%	0 0%	0 0%
Calm	0 0%	0 0%	0 0%	0 0%	0 0%
Anger	4 2%	67 37%	38 21%	2 1%	69 38%
Joy	10 6%	51 28%	16 9%	22 12%	81 45%
Sorrow	10 6%	59 33%	11 6%	4 2%	96 53%

	Unknown	Calm	Anger	Joy	Sorrow
Unknown	0 0%	0 0%	0 0%	0 0%	0 0%
Calm	0 0%	12 10%	4 3%	2 2%	102 85%
Anger	2 2%	12 10%	35 29%	43 36%	28 23%
Joy	2 2%	11 9%	38 32%	32 27%	37 31%
Sorrow	3 2%	39 32%	27 22%	19 16%	32 27%

Fig. 4. Confusion matrix for SER-Pepper run on: left) SAVEE dataset, right) RAVNESS dataset

The results of SER-BoM using the 10kfold validation were not too bad for most variants and datasets. It can be observed that the normalised variants outperform the remaining results for each dataset (Fig. 5).

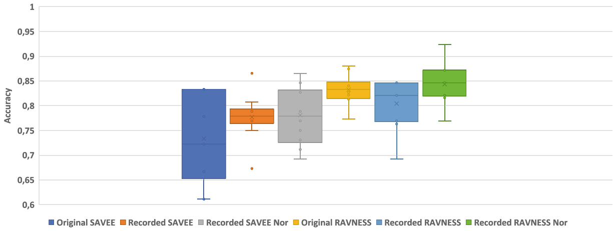


Fig. 5. SER-BoM accuracy after 10kfoldCV with SAVEE and RAVNESS datasets

The LOO results (see Fig. 6) state that the performance of SER-BoM is poor for most variants but with Original RAVNESS. Again the normalised SAVEE variant outperforms the other SAVEE variants lightly, but not the corresponding RAVNESS.

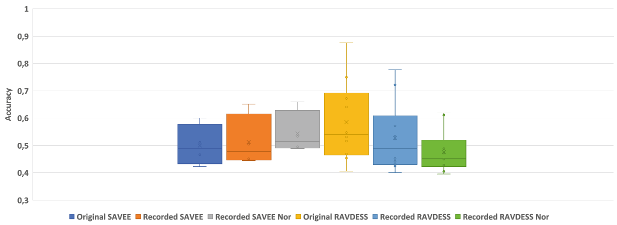


Fig. 6. SER-BoM accuracy after LOO with SAVEE and RAVNESS datasets

4.3 Comparison

Finally (see Fig. 7), it can be observed that our SER-BoM algorithm outperforms SER-Pepper clearly in all the variants, and obviously, the best sample-oriented normalised results (10CV*Nor) outperform the remaining the results.

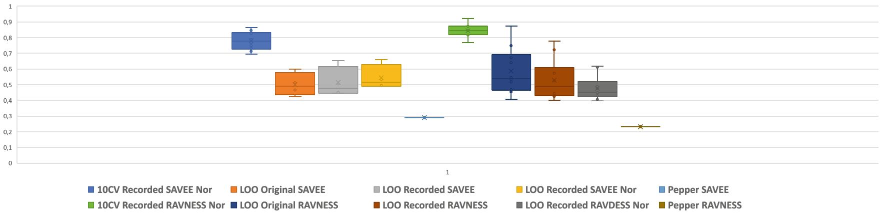


Fig. 7. SER-BoM vs SER-Pepper accuracy with SAVEE and RAVNESS datasets

5 Conclusion and Future Work

Since the performance of the SER-Pepper module sports really poor results for all the emotions in two well-known SER datasets, it seems that this module has enormous room for improvement. Furthermore, although the SER-BoM got acceptable results for the sample-centred validation strategy (10kfoldCV), the user-centred LOO validation's real scenario tests tell us that at least the SER-BoM should be improved by adding features, models as well as noise and artifacts removal module.

The further proposal is upgrading Pepper's emotional skills with a new SER custom algorithm like SER-BoMv2.0. However, this task requires an upgrade of the current Pepper's Operating System (from NAO® to Android). One of the challenging future projects is delivering Pepper® in a Senior-House gathering systematically the emotional state of older adults for mental health profiling. This project would comprise SER and conversational tasks and personal identification based on face recognition or biometric identification like Voice-Recognition or NLP.

Acknowledgement. This research has been funded by the Spanish Ministry of Economics and Industry, grant PID2020-112726RB-I00, by the Spanish Research Agency (AEI, Spain) under grant agreement RED2018-102312-T (IA-Biomed), and by the Ministry of Science and Innovation under CERVERA Excellence Network project CER-20211003 (IBERUS) and Missions Science and Innovation project MIG-20211008 (INMERBOT). Also, by Principado de Asturias, grant SV-PA-21-AYUD/2021/50994, and by European Union's Horizon 2020 research and innovation programme (project DIH4CPS) under the Grant Agreement no 872548. And by CDTI (Centro para el Desarrollo Tecnológico Industrial) under projects CER-20211003 and CER-20211022 and by ICE (Junta de Castilla y León) under project CCTT3/20/BU/0002.

References

1. Ahsan, M., Kumari, M.: Physical features based speech emotion recognition using predictive classification. Int. J. Comput. Sci. Inf. Technol. **8**(2), 63–74 (2016)
2. de la Cal, E., Gallucci, A., Villar, J.R., Yoshida, K., Koeppen, M.: A first prototype of an emotional smart speaker. In: Sanjurjo González, H., Pastor López, I., García Bringas, P., Quintián, H., Corchado, E. (eds.) SOCO 2021. AISC, vol. 1401, pp. 304–313. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-87869-6_29
3. Documentation, S.R.: Pepper SER algorithm - ALVoiceEmotionAnalysis (2012). http://doc.aldebaran.com/2-5/naoqi/audio/alvoiceemotionanalysis.html#alvoiceemotionanalysis
4. Dorogush, A.V., Ershov, V., Gulin, A.: CatBoost: gradient boosting with categorical features support, pp. 1–7 (2018)
5. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Mach. Learn. **63**(1), 3–42 (2006)
6. Haq, S., Jackson, P., Edge, J.: Audio-visual feature selection and reduction for emotion classification. In: Expert Systems with Applications, vol. 39, pp. 7420–7431 (2008)
7. Haq, S., Jackson, P.J.B.: Speaker-dependent audio-visual emotion recognition. In: Proceedings of the International Conference on Auditory-Visual Speech Processing (AVSP 2008), Norwich, UK (2009)
8. Haq, S., Jackson, P.J.B.: Machine Audition: Principles, Algorithms and Systems. Chap. Multimodal, pp. 398–423. IGI Global, Hershey (2010)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR abs/1512.0 (2015)
10. Livingstone, S.R., Russo, F.A.: The Ryerson audio-visual database of emotional speech and song (RAVDESS): a dynamic, multimodal set of facial and vocal expressions in north American english. PLoS ONE **13**(5), e0196391 (2018). https://doi.org/10.1371/journal.pone.0196391
11. Mitsuyoshi, S., Ren, F., Tanaka, Y., Kuroiwa, S.: Non-verbal voice emotion analysis system **2**(4), 4198 (2006)
12. Pandey, A.K., Gelin, R.: A mass-produced sociable humanoid robot: pepper: the first machine of its kind. IEEE Robot. Autom. Mag. **25**(3), 40–48 (2018)
13. Van Erp, M., Vuurpijl, L., Schomaker, L.: An overview and comparison of voting methods for pattern recognition. In: Proceedings - International Workshop on Frontiers in Handwriting Recognition, IWFHR, pp. 195–200 (2002)

Fine-Tuning of Optimisation Parameters in a Firefly Algorithm in Inventory Management

Dragan Simić[1](✉), José Luis Calvo-Rolle[2], José R. Villar[3], Vladimir Ilin[1], Svetislav D. Simić[1], and Svetlana Simić[4]

[1] Faculty of Technical Sciences, University of Novi Sad, Trg Dositeja Obradovića 6,
21000 Novi Sad, Serbia
dsimic@eunet.rs, {dsimic,v.ilin,simicsvetislav}@uns.ac.rs

[2] Department of Industrial Engineering, University of A Coruña, Avda. 19 de febrero s/n,
15405 Ferrol, A Coruña, Spain
jose.rolle@udc.es

[3] University of Oviedo, Campus de Llamaquique, 33005 Oviedo, Spain
villarjose@uniovi.es

[4] Faculty of Medicine, University of Novi Sad, Hajduk Veljkova 1–9, 21000 Novi Sad, Serbia
svetlana.simic@mf.uns.ac.rs

Abstract. The interest in the problems related to optimal inventory management at a scientific level goes back to the start of the 20th century. The inventory is a necessary feature to control and to forecast the level of future demand. Inventory control techniques are very important components and most organizations can substantially reduce their costs. This paper presents the firefly algorithm (FFA) for modelling the inventory control in a production system. The aim of this research is fine-tuning of parameters in FFA in inventory management in order to minimize production cost according to the price of items and inventory keeping cost. The experimental results demonstrate that it is possible to select values of FFA parameters that significantly reduce production cost.

Keywords: Inventory control · Firefly optimisation algorithm · Production system · Production cost

1 Introduction

The history of the supply chain engineering management is routed in scientific management pioneered during the Second Industrial Revolution. The foundation of this labour is routed in work measurement and different methods have been developed to organise production processes better. One of the expansions from these methods is inventory control [1]. Inventory control is the science-based art of controlling the amount of inventory (or stock) in various forms, within an organization, to meet the economic demand placed upon that business. Inventory amount and size decisions are traditionally driven by the costs of maintaining inventories and costs of being out of stock. Operations managers

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 645–654, 2023.
https://doi.org/10.1007/978-3-031-18050-7_63

seek to maintain inventory levels that minimize the total cost of both [1]. The interest in the problems of optimal stock management at a scientific level goes back to the start of the 20th century [2].

The importance of inventory control in business increased dramatically with the increasing interest rates of the 1970s. It was the rule of the hour to release surplus operating capital tied up in excessive inventories and to use the resulting liquidity to finance new investments. In order to control the amount of inventory, it is necessary to forecast the level of future demand, where such demand can be regarded as essentially either independent or dependent [3].

This paper presents the*firefly algorithm* (FFA) for modelling the inventory control in production systems. The aim of this research is fine-tuning of parameters in FFA in inventory management in order to minimize production cost according to the price of items and inventory keeping cost. Finally, the experimental results exhibit that it is possible to select values of FFA optimisation parameters that significantly reduce production cost. This research directly continues and expands the authors' previous research on inventory management [4]. Also, this paper, in general, continues the authors' previous research in supply chain management, and inventory management presented in [5–9].

The rest of the paper is organized in the following way: Sect. 2 overviews the related work. Section 3 presents modelling the inventory management, and the firefly optimisation algorithm implemented in it. This section also describes the used dataset. Experimental results and discussion are presented in Sect. 4 and finally, Sect. 5 provides concluding remarks.

2 Related Work

A fuzzy logic-based inventory control model is proposed in [10]. It maintains the inventory at a desired level despite fluctuations in the demand, taking into account the dynamics of the production system. The concept of fuzzy set theory has been applied to inventory control models considering the fuzziness of inputs and the dynamics of the systems.

An intelligent system implemented in a web-based environment and utilized to integrate multiple stores, intelligently determining the different reorder points in the systems, is proposed in [11]. The paper presents the optimisation of the performance of inventory management integrating multiple systems and providing an efficient coordination and monitoring by moving away from a single store into a distribution system and relating real time status of supplies at different stores.

The paper [12] presents the effective inquiry of artificial neural network modelling and forecasting in the issues of inventory management, especially in the lot-sizing problem. Several types of neural networks are created and tested within the research, searching for the most efficient neural network architecture.

Inventory control is interesting not just for manufacturing enterprises but for telecommunication companies as well. Huge material supplies freeze cash assets. A major challenge in supply chain inventory optimisation is handling, designing, analysing, and optimizing under uncertainty. The concept of linguistic variables, a fuzzy set, as an alternative approach to modelling human thinking in an approximate manner to summarize information and express it, is presented in [13].

3 Modelling the Inventory Management

Inventory is money bound in a nonmonetary form and unless it can be sold, excess inventory can be a huge drain on profits. This includes the storage cost and the lower value of perishable or dated goods. A unit of money has more value today than a year later, even if the inflation rate is zero. The reason lies in the fact that the use of money brings about a return, which can be paid as interest. Discounting is used when interest goes back in time [14].

Inventory can be classified in several ways depending on the industry, the company's operations, and the types of inventories the company manages. Generally, inventory can be grouped into four primary classifications: (i) raw materials are inventory items used in the manufacturing process to create finished goods; (ii) work-in-progress (WIP) inventory includes items that are currently being processed; (iii) finished goods are comprised of all completed items that are ready for sale to the final customer; and (iv) maintenance repair and operations goods (MRO).

This research continues and expands the authors' previous research on inventory management in production process [4]. The previous research is based on biological swarm intelligence technique in general, and particularly, the optimisation of FFA model for modelling the inventory control in a production system. The aim of this research is fine-tuning of optimisation parameters in FFA in inventory management in order to minimize production cost according to the price of items and inventory keeping cost.

The firefly algorithm (FFA) is a relatively new swarm intelligence optimisation method introduced in [15], in which the search algorithm is inspired by the social behaviour of fireflies and the phenomenon of bioluminescent communication.

Fireflies communicate, search for pray, and find mates using bioluminescence with varied flashing patterns. Attractiveness is proportional to the brightness, which decreases with the increase in the distance between fireflies. If there are no brighter fireflies than one particular candidate solution, it will move at random in the space [16]. The brightness of a firefly is influenced or determined by the objective function. For a *maximization* / *minimization* problem, brightness can simply be *proportional* / *inversely proportional* to the value of the cost function. More details about FFA and its variants are depicted in [17].

Algorithm 1 *The algorithm of* ***firefly algorithm***

```
Begin
        Step 1: Initialization. Set the generation counter G = 1; Initialize the population of n fireflies P randomly with each firefly corresponding to a potential solution to the given problem; Define light absorption coefficient $\gamma$;
                Set control to the step size $\alpha$ and the initial attractiveness $\beta_0$ at r = 0.
        Step 2: Evaluate the cost function I for each candidate in P determined by f(x)
        Step 3: While the termination criteria are not satisfied or G < MaxGeneration do
                    for i=1:n (all n candidate solution) do
                        for j=1:n (n candidate solution) do
                            if ($I_j < I_i$),
                                move candidate i towards j;
                            end if
                            Vary attractiveness with distance r via exp[$-\gamma^{r2}$];
                            Evaluate new solutions and update cost function;
                        end for j
                    end for i
                    G = G+1;
        Step 4: end while
        Step 5: Post-processing the results and visualization;
End.
```

The basic steps of the FFA are summarized by the pseudo code revealed in Algorithm 1. The *light intensity* or attractiveness value $\boldsymbol{\beta}$ depends on the distance $\boldsymbol{r}$ between the fireflies and the media light absorption coefficient $\boldsymbol{\gamma}$. The *attractiveness* of each firefly is determined as monotonic decreasing function, where $\boldsymbol{\beta_0}$ represents the attractiveness of the firefly at $r = 0$ and it is usually called *initial attractiveness*, using the equation:

$$\beta(r) = \beta_0 e^{-\gamma r^2} \tag{1}$$

The *movement* of a firefly f_{j} from position x_{j} to new position $x_{\mathrm{j+1}}$, attracted to a brighter firefly f_{i} at position x_{i}, is established by the equation:

$$x_{j+1} = x_j + \beta_0 e^{-\gamma r_{ij}^2}\left(x_j - x_i\right) + \alpha\varepsilon_i \tag{2}$$

where r_{ij} is the distance between two fireflies, $\boldsymbol{\alpha}$ is the mutation coefficient, and ε_{i} are continuous uniform random numbers.

The dataset is taken from [18]: 1) *demand for products*; 2) *item price*; and 3) *inventory keeping cost*; these are presented from Tables 1, 2 and 3. The *maximum production capacity* is 400 units per day.

4 Experimental Results and Discussion

The cost function of the FFA optimisation inventory value is:

$$\min(ProductionCost) = \min \sum\nolimits_{i=1day}^{10days} (productionplan_i * itemprice_i) \tag{3}$$

Table 1. Demand for products

Product	Day										Total
	1	2	3	4	5	6	7	8	9	10	
1	59	34	84	69	28	25	55	65	74	36	529
2	84	34	31	46	78	36	80	64	22	40	515
3	57	56	50	79	40	54	22	90	41	37	526

Table 2. Item price

Product	Day										Average
	1	2	3	4	5	6	7	8	9	10	
1	188	138	176	104	153	133	200	189	163	181	162.5
2	149	129	117	181	196	173	182	117	188	183	161.5
3	147	173	188	182	185	103	120	171	200	154	162.3

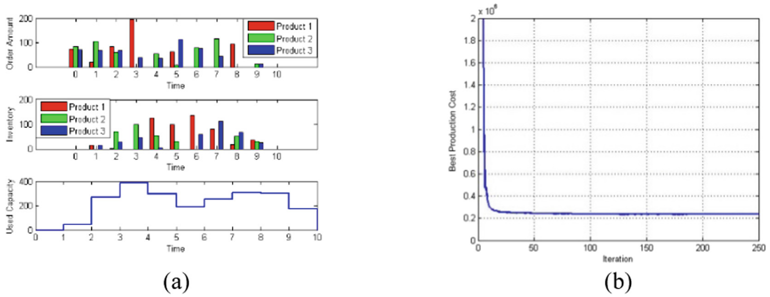
Table 3. Inventory keeping cost

Product	Day										Average
	1	2	3	4	5	6	7	8	9	10	
1	3	6	10	2	4	7	3	7	3	10	5.5
2	8	10	4	8	5	5	5	4	7	6	6.2
3	3	9	5	7	8	8	8	4	3	5	6.1

4.1 Previous Research

Experimental results of previous research [4], for the FFA method, number of planning days, total production per each day, and production cost are presented in Table 4. The experimental results for firefly optimisation algorithm: (a) used capacity, inventory and order account in time and (b) *Best Production Cost* on the number of iterations are presented in Fig. 1. In this experiment, the following value of parameters: *maximum number of iterations* = 500; *number of fireflies* = 25; *mutation coefficient* $\boldsymbol{\alpha} = 0.3$; *initial attractiveness* $\boldsymbol{\beta_0} = 2$; and *media light absorption coefficient* $\boldsymbol{\gamma} = 0.4$ are used in the firefly optimisation process.

According to the nature of the FFA algorithm, the experiment is repeated 100 times with the defined dataset and experimental results for minimizing production cost, as summarized and presented in Table 4.



(a) (b)

Fig. 1. Experimental results for *firefly optimisation algorithm*: (a) used capacity, inventory and order account in time; (b) best production cost in a number of iterations

Table 4. The best production cost, number of days (*Day*), total production per day (*Total*), total production in 10-day horizon

Prod.	Day										Total	Prod. cost
	1	2	3	4	5	6	7	8	9	10		
1	67	31	79	171	73	30	0	0	78	0	529	**234400**
2	94	52	127	2	7	85	22	126	0	0	515	
3	85	87	0	71	39	60	74	75	0	35	526	

The minimum value for the production cost for FFA optimisation for 100 experiments is 234,400 monetary units, and it presents the *Best Production Cost*. The second production cost is 239,332; and the third is 240,413 monetary units. The average value for *Production Cost* for FFA optimisation for 100 experiments is 241,895.4 monetary units.

4.2 Computational Results and Analysis

The aim of this research is fine-tuning of parameters in FFA in inventory management in order to minimize production cost according to the price of items and inventory keeping cost. In this experiment, the following values of parameters are kept from previous research: *maximum number of iterations* = 500 and *number of fireflies* = 25. On the other side, the following optimisation parameters are changed: (i) *mutation coefficient* $\boldsymbol{\alpha}$ = [0.1 0.3 0.8]; (ii) *initial attractiveness* $\boldsymbol{\beta_0}$ = [1 2 4]; and (iii) and *media light absorption coefficient* $\boldsymbol{\gamma}$ = [0.4 0.7 1]; these are used in the firefly optimisation process, and production costs are calculated.

In other words, in this experiment, 27 different combinations are possible, and every one is repeated 10 times. For every experiment, 92 KB result is generated and stored, which is about 2.5 MB for the entire experiment. Only a fraction of experimental results is summarised, and the most important among them for minimising productions cost are presented in Table 5.

Experimental results for production cost for: maximum number of iterations = 500; number of fireflies $n = 25$, mutation coefficient – $\alpha = [0.1\ 0.3\ 0.8]$, initial attractiveness – $\beta_0 = [1\ 2\ 4]$, media light absorption coefficient – $\gamma = [0.4\ 0.7\ 1]$, number of iteration - no. it, and elapsed time are presented in Table 5. The number of iterations is always less than the proposed maximum number of iterations. In our experiment, additional termination criterion is that *production cost* does not change in the last 20 iterations.

Since the most important calculated value is the production cost, Table 5 exhibits the minimal, maximum and average value. The minimum *average value* for production cost is 233,683, and standard deviation 3,680 monetary units for the following parameters: $\alpha = 0.8$, $\beta_0 = 2$, and $\gamma = 1$, which is marked bold-underline in Table 5, column 7, line 24. The second *average value* for production cost is 236,931 monetary units for the following parameters: $\alpha = 0.8$, $\beta_0 = 2$, and $\gamma = 0.7$, which is marked bold in Table 5, column 7, line 15. And, the third *average value* for production cost is 237,967 monetary units for the following parameters: $\alpha = 0.8$, $\beta_0 = 2$, and $\gamma = 0.4$, which is marked bold in Table 5, column 7, line 6.

The minimum *minimum value* for production cost is 226,688 monetary units for the following parameters: $\alpha = 0.8$, $\beta_0 = 2$, and $\gamma = 1$, which is marked bold-underline in Table 5, column 5, line 24. Therefore, 226,688 monetary units is *Best Production Cost* in this optimisation process. The second *minimum value* for production cost is 231,318 monetary units for the following parameters: $\alpha = 0.8$, $\beta_0 = 2$, and $\gamma = 0.4$, which is marked underline in Table 5, column 5, line 6. And, the third *minimum value* for production cost is 232,516 monetary units for the following parameters: $\alpha = 0.8$, $\beta_0 = 2$, and $\gamma = 0.7$, which is marked underline in Table 5, column 5, line 15.

Table 5. Experimental results for Best Production Cost (BPC) for: maximum number of iterations = 500; number of fireflies $\boldsymbol{n} = 25$, media light absorption coefficient - $\boldsymbol{\gamma}$, initial attractiveness - $\boldsymbol{\beta_0}$, mutation coefficient - $\boldsymbol{\alpha}$, number of iteration - **no. it**, and elapsed time for experiment

No.	γ	β_0	α	Best production cost			no. it	Elapsed time (s)
				Min	Max	Average	Average	Average
1	0.4	1	0.1	1213585	3187922	2077716	127	53
2	0.4	1	0.3	272051	336738	289056	178	70
3	0.4	1	0.8	238169	251142	*244822*	210	85
4	0.4	2	0.1	242971	259355	249281	185	71
5	0.4	2	0.3	236669	249015	242906	239	149
6	0.4	2	0.8	231318	246309	**237967**	261	153
7	0.4	4	0.1	244344	267604	253779	157	81
8	0.4	4	0.3	242164	267268	248682	195	125
9	0.4	4	0.8	239571	248007	*243566*	278	173
10	0.7	1	0.1	577917	3746089	2126847	127	108
11	0.7	1	0.3	264431	527318	323738	180	151

(*continued*)

Table 5. (*continued*)

No.	γ	β_0	α	Best production cost			no. it	Elapsed time (s)
				Min	Max	Average	Average	Average
12	0.7	1	0.8	240266	251453	*247867*	216	153
13	0.7	2	0.1	235252	258196	245623	182	92
14	0.7	2	0.3	238152	250639	244004	248	136
15	0.7	2	0.8	232516	240968	**236931**	287	112
16	0.7	4	0.1	238825	260381	250243	174	68
17	0.7	4	0.3	234200	262246	245676	226	91
18	0.7	4	0.8	239000	250870	*244137*	250	99
19	1	1	0.1	1133366	4079682	2620837	126	51
20	1	1	0.3	280240	548074	331371	180	71
21	1	1	0.8	242343	251584	*245300*	221	88
22	1	2	0.1	239772	259634	247328	175	71
23	1	2	0.3	238309	245195	241772	237	90
24	1	2	0.8	**226688**	238307	**233683**	256	99
25	1	4	0.1	234043	260643	247616	183	72
26	1	4	0.3	232807	249368	243851	209	85
27	1	4	0.8	236400	248212	*242918*	267	104

After these observations, it could be concluded that, after fine-tuning, the following optimisation parameters in firefly algorithm in inventory management are the best: mutation coefficient $\alpha = 0.8$; initial attractiveness $\beta_0 = 2$; and media light absorption coefficient $\gamma = 1$. It can be concluded that mutation coefficient $\boldsymbol{\alpha}$ is very significant for final results, though it needs a little longer elapsed time. The mutation coefficient $\boldsymbol{\alpha} = 0.1$ does not provide satisfactory results.

There is not so much research and scientific papers published in journals or conference proceedings which discuss optimal parameters in FFA. The research paper [19] proposes maximum number of iterations = 500; the population size = 25; $\alpha = 0.01$; $\beta_0 = 0.6$; and $\gamma = 0.2$. On the other hand, well known and many times cited research paper [20] proposes maximum number of iterations = 500; the population size = 20; $\alpha = 0.25$; $\beta_0 = 0.2$ and $\gamma = 1$. Compared with our experimental results for optimal and best production cost in inventory management: maximum number of iterations = 500; population size = 25; $\alpha = 0.8$; $\beta_0 = 2$ and $\gamma = 1$, it is very difficult to make final decision. It can also be concluded that mutation coefficient $\boldsymbol{\alpha}$ is very significant for final results, though it needs a little longer elapsed time.

5 Conclusion and Future Work

The aim of this research is fine-tuning of parameters in FFA in inventory management in order to minimize production cost according to the price of items and inventory keeping cost. The experimental results are generated and stored, amounting to approximately 2.5 MB. These experimental results could be analysed in more details in the future work. The experimental results demonstrate that it is possible to select values of FFA parameters that significantly reduce production cost in inventory management.

Experimental results encourage the authors' further research. As the optimisation method, the firefly has several parameters that determine its behaviour and efficacy. The future work could focus on extending research on good choice of parameters for various optimisation scenarios which should help the production manager achieve different solutions and make better operational solutions and better production results with less effort. Then, this model could be tested with an original very large real-world dataset obtained from the existing different manufacturing companies.

Acknowledgment. This research (paper) has been supported by the Ministry of Education, Science and Technological Development through project no. 451-03-68/2022-14/ 200156 "Innovative scientific and artistic research from the FTS (activity) domain".

References

1. Jones, E.C.: Supply Chain Engineering and Logistics Handbook - Inventory and Production Control. CRC Press, Boca Raton (2019)
2. Lewis, C.: Demand Forecasting and Inventory Control: A Computer Aided Learning Approach. Wiley, New York (1998)
3. Bartmann, D., Beckmann, M.J.: Inventory Control: Models and Methods. Springer, Heidelberg (1992). https://doi.org/10.1007/978-3-642-87146-7
4. Simić, D., Ilin, V., Simić, S.D., Simić, S.: Swarm intelligence methods on inventory management. In: Graña, M., et al. (eds.) SOCO'18-CISIS'18-ICEUTE'18 2018. AISC, vol. 771, pp. 426–435. Springer, Cham (2019). https://doi.org/10.1007/978-3-319-94120-2_41
5. Simić, D., Svirčević, V., Corchado, E., Calvo-Rolle, J.L., Simić, S.D., Simić, S.: Modelling material flow using the Milk run and Kanban systems in the automotive industry. Expert. Syst. **38**(1), e1254 (2021). https://doi.org/10.1111/exsy.12546
6. Simić, D., Simić, S.: Hybrid artificial intelligence approaches on vehicle routing problem in logistics distribution. In: Corchado, E., Snášel, V., Abraham, A., Woźniak, M., Graña, M., Cho, S.-B. (eds.) HAIS 2012. LNCS (LNAI), vol. 7208, pp. 208–220. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28942-2_19
7. Simić, D., Svirčević, V., Simić, S.: A hybrid evolutionary model for supplier assessment and selection in inbound logistics. J. Appl. Logic **13**(2), Part A, 138–147 (2015). https://doi.org/10.1016/j.jal.2014.11.007
8. Simić, D., Kovačević, I., Svirčević, V., Simić, S.: 50 years of fuzzy set theory and models for supplier assessment and selection: a literature review. J. Appl. Logic **24**, Part A, 85–96 (2017). https://doi.org/10.1016/j.jal.2016.11.016
9. Simić, D., Kovačević, I., Svirčević, V., Simić, S.: Hybrid firefly model in routing heterogeneous fleet of vehicles in logistics distribution. Logic J. IGPL **23**(3), 521–532 (2015). https://doi.org/10.1093/jigpal/jzv011

10. Samanta, B., Al-Araimi, S.A.: An inventory control model using fuzzy logic. Int. J. Prod. Econ. **73**(3), 217–226 (2001). https://doi.org/10.1016/S0925-5273(00)00185-7
11. Madamidola, O.A., Daramola, O.A., Akintola, K.G.: Web-based intelligent inventory management system. Int. J. Trend Sci. Res. Dev. **1**(4), 164–173 (2017). https://doi.org/10.31142/ijtsrd107
12. Šustrová, T.: A suitable artificial intelligence model for inventory level optimization. Trends Econ. Manage. **25**(1), 48–55 (2016). https://doi.org/10.13164/trends.2016.25.48
13. Zhivitskaya, H., Safronava, T.: Fuzzy model for inventory control under uncertainty. Central Eur. Res. J. **1**(2), 10–13 (2015)
14. Keynes, J.M.: The General Theory of Employment, Interest, and Money. (Reprint edition) Macmillan and Co., London (1949)
15. Yang, X-S.: Firefly algorithm, Lévy flights and global optimization. In: Bramer, M., Ellis, R., Petridis, M. (eds.) Research and Development in Intelligent Systems XXVI. Springer, London (2010). https://doi.org/10.1007/978-1-84882-983-1_15
16. Yang, X.-S.: Cuckoo search and firefly algorithm: overview and analysis. In: Yang, X.-S. (ed.) Cuckoo Search and Firefly Algorithm. SCI, vol. 516, pp. 1–26. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-02141-6_1
17. Yang, X.-S. (ed.): Cuckoo Search and Firefly Algorithm. SCI, vol. 516. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-02141-6
18. Yarpiz. Inventory Control using PSO in MATLAB (2022). https://www.mathworks.com/matlabcentral/fileexchange/53142-inventory-control-using-pso-in-matlab. MATLAB Central File Exchange. Accessed 11 June 2022
19. Saha, S.K., Kar, R., Mandal, D., Ghoshal, S.: Optimal stable IIR low pass filter design using modified Firefly algorithm. In: Panigrahi, B.K., Suganthan, P.N., Das, S., Dash, S.S. (eds.) SEMCCO 2013. LNCS, vol. 8297, pp. 98–109. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-03753-0_10
20. Yang, X.-S.: Firefly algorithms for multimodal optimization. In: Watanabe, O., Zeugmann, T. (eds.) SAGA 2009. LNCS, vol. 5792, pp. 169–178. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04944-6_14

Security Centric Scalable Architecture for Distributed Learning and Knowledge Preservation

Rudolf Erdei(✉), Daniela Delinschi, and Oliviu Matei

Technical University of Cluj Napoca, North University Centre of Baia Mare,
Cluj-Napoca, Romania
{rudolf.erdei,oliviu.matei}@cunbm.utcluj.ro

Abstract. This article presents the architecture, design and validation of a distributed learning approach, that provides support for knowledge preservation. The architecture is able to provide support for Collaborative Data Mining, Context Aware Data Mining, but also Federated Learning. Improving User Experience, providing support for research activities and providing a framework for production-grade machine learning pipeline automations were the primary objectives for the design of the proposed architecture. Third party service support is available out of the box, maintaining the loose-coupling of the system. Obtained results are promising, the system being validated with a use case on Collaborative Data Mining.

1 Introduction

Nowadays, *Distributed Learning (DL)* [1,6] is one of the key research areas in the field of *Artificial Intelligence* and *Data Science*. The need for DL arises from the ever evolving need for *data security*, *data privacy*, *resources optimization*, and, in general, *cost control*. Traditional (centralized) Machine Learning (ML) methodologies use a single node to implement the entire *ML pipeline*, that has to be executed over the entire dataset, in order to obtain a working model. This working model can later be used by the Edge Nodes (or third-party locations) to obtain certain estimations or classifications for other input data.

This approach raises some important problems: training data must be uploaded from the Edge Nodes to the Cloud Component, that on the other hand has to be able to retain all that information while also possessing the computing power needed to process it. Time consuming pipeline operations are executed over the entire dataset at once.

While the problem of data-privacy can be partially mitigated using *Homomorphic Encryption* [22], this operation does not resolve the bandwidth problem. The latter could be mitigated by using *Dimensionality Reduction* and *Data Reduction* [8] techniques, or even newer formats for data compression (like Parquet files [4]).

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 655–665, 2023.
https://doi.org/10.1007/978-3-031-18050-7_64

The question arises if both the privacy and the bandwidth problems could be optimally mitigated, by taking advantage of the steady increase in *low-energy high-power computing* that is currently available on the Edge Nodes. *Federated Learning* [17,26] creates and trains partial models on the edge nodes, uploading them to the Cloud Component, that only has to integrate these partial models. This approach will mostly eliminate the need for a central node with high storage and processing requirements. Secure Computing [11] employs different strategies for providing Data Security, while also retaining the full power of Cloud Computing and delivering consistent results.

The structure of this article is as follows: Sect. 2 describes the latest work in the field of *Federated Learning* (FL), Sect. 3 presents the use-case that inspired the work, Sect. 4 presents the functional and non-functional requirements for the system, Sect. 5 discusses the architecture, Sect. 7 presents advantages/disadvantages of the current architecture and also a comparison to a similar work, and lastly Sect. 8 discusses conclusions and future work.

2 Related Work

As *Distributed Systems* (DS) progress turning the Internet into a huge scale organism, the importance of software and platform architectures grows. In this context, the *Pareto Principle* [24] enhances the importance of a well-designed architecture, that will save on implementation time later on in the life-cycle of the software platform.

Matei et al. [20] have developed a flexible architecture that makes use of *Serverless Components* for a face-recognition app, with very good results, thus demonstrating the effectiveness of the *Serverless Architecture*, and also a multilayer architecture for soil moisture prediction [18] that is done in a *Context-Aware* methodology. Both these architectures make use of flexibility as well as layering in software components, in order to optimize all aspects of development and practical exploitation.

Data Silos are used heavily in *Distributed Learning* (actually every edge node is treated as a *Data Silo*). Durrant et al. [10] discusses about the role of DL in the Agri-Food sector. A soybean yield forecasting use-case is used for demonstration purposes, with good results, as the resulting model performed better than models trained on single data-sources.

Gupta et al. [11] present a methodology that uses *Machine Learning Agents* which work on subsets of the data, generating partial *Deep Neural Networks* (DNN) and then centralizing them and producing a final integrated model. The only data transfer that occurs between the central component and the edge nodes is composed of tensor parameters and gradients.

Li et al. [16] propose a modified version of *Federated Learning* architecture, where the system works on an overall model at once, agents being able to compute gradients and send them back to the central node. The system can handle *byzantine workers*, that send corrupt responses, due to local data corruption or any other causes (including maleficent behavior).

DL can also be regarded as a Bias Averaging System (reduction of overall bias in models by averaging multiple biased models). Alireza et al. [9] discuss about a methodology that brings *Adaptive Bias Compensation* in an energy efficient way. Their system uses a two-stage Analog to Digital Converter (ADC) with a two-fold adaptive feedback loop for signal processing and preparing the application of Data Mining Algorithms. Biases can thus be mitigated, if not eliminated completely.

Federated Learning Architecture for Smart Agriculture (SA) is discussed by Kumar et al. [15] in their article about the *PEFL* Framework, that securely raises the accuracy of the predictions up to 98%.

Model Trust is a topic discussed by Chakraborty et al. [5], where they propose storing the models within a blockchain. This approach, however, will affect the *privacy* of the model, as blockchains are inherently public, and also any encryption method will eventually be hacked in the future. The inability to remove blockchain entries can be detrimental for some use-cases.

3 Agricultural Use-Case for Distributed Learning

While data in the *Agri-Food* field may not be as sensitive as in other fields (e.g.: the medical field), the sheer amount of sensor data generated by sensors of all kinds (often multiple sensor stations spread over a large area) can mean a huge bandwidth and storage space would be required just to upload and store the required data. Model parameters, however, would require much less storage and bandwidth, making this choice preferred, over the former.

MUSHNOMICS[1] is a three-stage holistic platform that enables mushroom growers to *1. prepare the substrate*, *2. grow the mushrooms* and *3. dispose the spent substrate* in an economic and eco-friendly way. The digital platform assesses the entire infrastructure at all times, through several sensing components, allowing the overseer to accurately predict the final yield of the growing system, in order to better plan the required logistics.

As mushroom growers often work with private data and there is some competition between them, the platform needs to be able to work with client specific models. This constraint, however, leads the way to generalization of the platform for all kinds of other Agri-Food use-cases, not just mushroom specific, that can be applied in the same platform, thus raising the flexibility of the result. Collaborative models for data-mining can also be implemented, if consortia of producers will agree to share their models (not their data), in order to better transfer new knowledge within the group. Also, science use-cases will benefit from the new found knowledge, as the system should have a knowledge preserving system (ontologically modeled).

[1] https://mushnomics.org/.

4 Functional and Non-Functional Requirements for the Distributed Learning Architecture

Within the project, requirements are elicited after brain-storming sessions that also included project beneficiaries as well as relevant stakeholders. After the initial idea generation, the structuring of requirements was done by using the *MoSCoW* methodology [2], that prioritizes features with most benefits for the end-result, and also using the *ATAM* Method [13]. Assessment of user expectations and the market demand for a privacy-centric solution were some other factors that were taken into consideration when designing these requirements.

Functional Requirements are formulated as high-level platform functionality. Each element was then split in more specific and atomic requirements needed for the development, testing and validation phases. From these, the most important ones are:

- FR1 - Ability to use multiple DL Methodologies, including *Federated Learning*, but also provide support for a classic (centralized) ML pipeline;
- FR2 - The ability of the system to provide both research support [3] as well as production grade automatic pipelines [21];
- FR3 - The ability of the system to accommodate several agri-producers in a secure, data-private way, but also enable support for consortia of agri-producers;
- FR4 - The system's ability to provide its services in a privacy centric way;
- FR5 - Ability to retain discovered knowledge, per project;
- FR6 - Ability to connect 3^{rd} party client applications/integrations to the system.

An excerpt from the **Non-Functional Requirements**, as performance and security aspects that the platform needs in order to provide quality services, are:

- NFR1 - *Security*, *Scalability* and *Stability* of the resulting system;
- NFR2 - *Platform Resilience*, as the ability to *manage and recover several failure* classes within the system;
- NFR3 - *Low-spec hardware requirements* for the Cloud Component (leveraging Serverless Components);
- NFR4 - *High-Performance* and *Flexible API* for 3^{rd} party integrations;
- NFR5 - *Portability/Accessibility*, as the possibility of the platform to be accessed from various operating systems, including mobile;
- NFR6 - *Flexibility*, as the platform should be used both for scientific as well as production purposes.

5 Distributed Learning System Architecture

DL methodologies have a huge benefit of using the edge nodes for generating a partial model. But, in order for the *Model Federation* to be obtainable, the models themselves have to be *compatible* [14]. The usual way of federating models is by means of averaging the parameter values over all the partial models.

Model characteristics are sent as metadata, so the Cloud Component can estimate an importance factor for it, factor that will be translated into the model's weight in the average. These operations ensure the robustness of the model and initial resistance against problems that can arise, including edge node byzantine failures [27].

Model compatibility is obtained in two steps:

1. **Model Build Characteristics** (like number of hidden layers and/or number of tensors per layer, for Neural Networks) and an initial model are obtained in the *research* part of the working methodology. This provides the best settings for the model to be built. These characteristics, that are unique per project, are then sent to the Edge Nodes;
2. Edge Nodes use the received characteristics and local data, to generate a **partial model** that will be uploaded to the Cloud Component. By using the same settings, averaging the models can be done in a more straightforward way.

In the case of FL, the Cloud Component does not permanently store the partial models (as this would pose some privacy risks), rather use the partial models to update the global model and then discard the used partial model.

The high level architecture is presented in Fig. 1. We can observe the multiple *Edge Nodes*, that provide required project inputs to the *Cloud Component*, and also receive model parameters and other platform settings (like sensor read frequency). The *Cloud Component* centralizes either the full data (for classical ML methodologies) or the resulting partial models, in order to provide analytics, data/model storage, knowledge preservation and an advanced *model creation pipeline* that can either work in a classical manner, or in a decentralized (federated) one. Platform access can be provided either through the *Web UI*, or through an *API*.

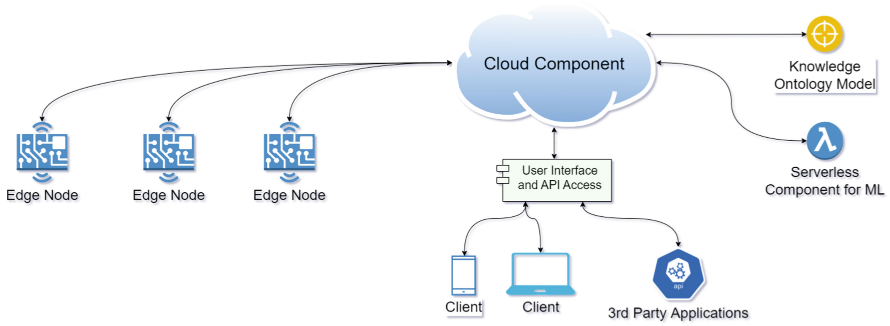


Fig. 1. High level architecture for distributed learning

The detailed architecture is presented in Fig. 2.

The roles and responsibilities of the three large components of the system are:

The **Edge Nodes** are responsible for the local data gathering devices, centralizing and pre-processing data from sensors. Depending on the methodology, the data will be used either for generating/enhancing a local model, or uploading it to the *Cloud Component*, via the `Data Export Module`. If needed, the *Edge Nodes* can have a local administration UI, that will also present local statistics and *Node Health*.

The **Cloud Platform** Component is responsible for aggregating either data or partial models, generating the initial model scheme, providing support for *Edge Nodes Management*, *System Overview* and *Knowledge Management*.

The **Platform UI** Component is responsible with managing the communications with the clients, through the `Web UI Interface`, so its modules are focused solely on this task.

5.1 Cloud Component Architecture

It is composed of several subsystems that provide important services for the platform. These subsystems have very specialized roles, each composed of one or several modules that will cooperate in order to fully implement the designated requirements. A short description for the subsystems is:

- **Edge Communication Manager** - involves all the modules that receive or send information to/from the edge components, and is governed by the *Access Control Module*. The modules also provide the required encryption/decryption mechanisms that the platform will use;
- **Multi-paradigm Data Ingestion Pipeline** - manages the *import* of data/partial models into the platform, checking the *data validity* and calculating *model quality* (see Sect. 8);
- **Data Persistence** - assesses if the received data can/must be persisted, and acts on this decision. Also, forwards received models to the FL methodology, that will trigger the *General Model Update*;
- **Knowledge Preservation** - gathered knowledge will enhance/update the already existing knowledge (the ontological model). These modules will keep the knowledge on a project basis, allowing the ontologies to be published or shared;
- **Multi-paradigm Machine Learning Pipeline** - has multiple methodologies that can be selected per project, that use the data to generate models, or use partial models to update a general model.

6 Architecture Validation

Validation of the architecture was done by using *Software Architecture Analysis Method* (SAAM) [25] and *Active Reviews for Intermediate Designs* (ARID) [7] methodologies, that were used in architecture review meetings with the involved stakeholders.

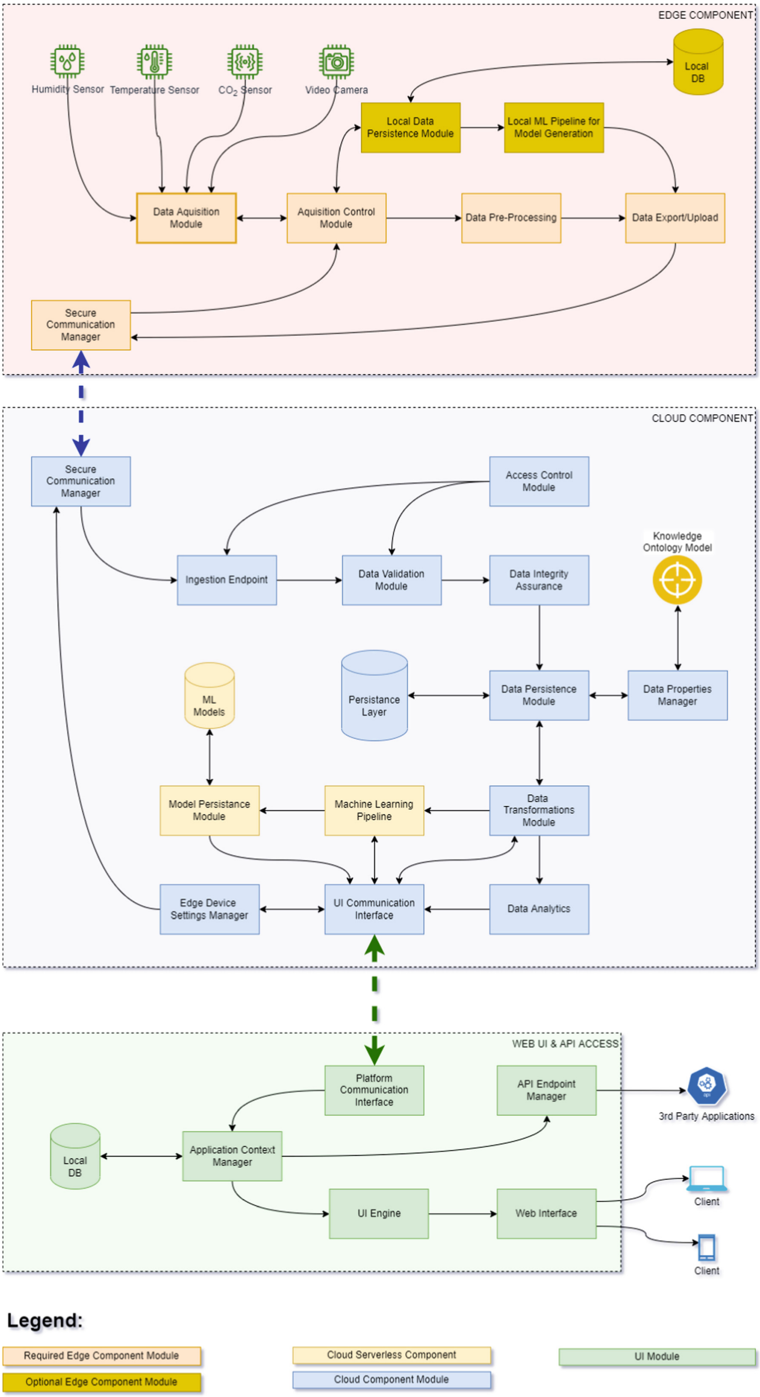


Fig. 2. Detailed architecture for the distributed learning system

The mapping between platform components and requirements was thoroughly analyzed and discussed, dissecting any raised concerns done by stakeholders and mitigating any potential issues. These methodologies lead to an Incremental design for the final architecture, having at least 4 increments until the architecture was finished. At the end of the validation process, the final architecture was proven to support all the requirements that the business case has required (Table 1).

Table 1. Correlation between the Functional requirements and modules withing the architecture

Functional requirement	Modules that are used for functionality implementation
FR1	ML Modules within Edge Components, Model Integration pipeline in the Cloud Component
FR2	Flexible Model Generation Pipeline in the Cloud Component, UI Component, Communication with Edge Components
FR3	Access Control Module, Model Persistence Module
FR4	Federated Learning Methodology within the Machine Learning Pipeline
FR5	Access Control Module, Model Persistence Module, Knowledge Ontology Model
FR6	Access Control Module, API Endpoint Manager

7 Discussions

The proposed architecture provides the required architectural flexibility in order to integrate several Data Mining paradigms, including Federated Learning, which is the main focus of this research. The platform also provides secure communications, by encrypting all data transfer between edge nodes and the central Cloud Component. By limiting the amount of data transferred between components and changing way learning is tone, the required bandwidth is limited to a minimum, so the communication channel is a non-critical part of the system, unlike in many other ML paradigms.

Disadvantages include the inability of the system to perform realtime/time-critical operations. In some specific cases this aspect might be crucial. In our agricultural use-case, the time window for solving specific issues that the platform might predict is rather large (hours or maybe even days), so realtime performance is not needed in this case.

When compared to the *Collaborative Data Mining* (CDM) Architecture developed by Matei et al. [19], we can already observe some key differences:

- The proposed architecture will be able to also integrate CDM, that becomes in this case a local methodology;
- When using DL Methodologies (like FL), *Edge Nodes* will not need to upload data to the cloud, so privacy of user data is enhanced;

- *Physical Devices* are abstracted into *Edge Nodes*, that have greater importance and perform a larger range of *Data Transformations*;
- ML can also be done on the *Edge Nodes*;

8 Conclusions and Further Research

The presented architecture successfully integrates all functional and non-functional requirements, successfully addressing safety, security and trust issues regarding the edge nodes, over the medium of communication. Several layers of trust have been designed, in order to provide a *Multiple Point Fault Detection* system [12] providing a base for further research over *Building Trust in Distributed Machine Learning Paradigms.*

Further research will be done on the assessment for the *Quality of the Model*, as a subset of *Model Trust* [23], that has a huge importance as it defines the weight the model will have in the overall average of the project. A *Model Quality Index* is being researched, that will allow better general models to be generated from this methodology.

Acknowledgment. This research was made possible by funding from the ICT-AGRI-FOOD 2019 Joint Call. This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CCCDI-UEFISCDI, project number COFUND-ICT-AGRI-FOOD-MUSHNOMICS 205.2020, within PNCDI III.

References

1. Alavi, M., Marakas, G.M., Yoo, Y.: A comparative study of distributed learning environments on learning outcomes. Inf. Syst. Res. **13**(4), 404–415 (2002)
2. Bebensee, T., van de Weerd, I., Brinkkemper, S.: Binary Priority List for Prioritizing Software Requirements. In: Wieringa, R., Persson, A. (eds.) REFSQ 2010. LNCS, vol. 6182, pp. 67–78. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14192-8_8
3. Blockeel, H., Vanschoren, J.: Experiment databases: towards an improved experimental methodology in machine learning. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702, pp. 6–17. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74976-9_5
4. Boufea, A., Finkers, R., van Kaauwen, M., Kramer, M., Athanasiadis, I.N.: Managing variant calling files the big data way: using HDFS and apache parquet. In: Proceedings of the Fourth IEEE/ACM International Conference on Big Data Computing, Applications and Technologies, pp. 219–226 (2017)
5. Chakraborty, S., Chakraborty, S.: Proof of federated training: accountable cross-network model training and inference. arXiv preprint arXiv:2204.06919 (2022)
6. Chen, M., et al.: Distributed learning in wireless networks: recent progress and future challenges. IEEE J. Sel. Areas Commun. (2021)
7. Clements, P.C.: Active reviews for intermediate designs. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST (2000)

8. Czarnowski, I., Jedrzejowicz, P., Chao, K.-M., Yildirim, T.: Overcoming "big data". barriers in machine learning techniques for the real-life applications (2018)
9. Danaee, A., de Lamare, R.C., Nascimento, V.H.: Energy-efficient distributed learning with adaptive bias compensation for coarsely quantized signals. In: 2021 IEEE Statistical Signal Processing Workshop (SSP), pp. 61–65. IEEE (2021)
10. Durrant, A., Markovic, M., Matthews, D., May, D., Enright, J., Leontidis, G.: The role of cross-silo federated learning in facilitating data sharing in the agri-food sector. Comput. Electron. Agric. **193**, 106648 (2022)
11. Gupta, O., Raskar, R.: Distributed learning of deep neural network over multiple agents. J. Netw. Comput. Appl. **116**, 1–8 (2018)
12. Isermann, R.: Model-based fault-detection and diagnosis-status and applications. Ann. Rev. Control **29**(1), 71–85 (2005)
13. Kazman, R., Klein, M., Barbacci, M., Longstaff, T., Lipson, H., Carriere, J.: The architecture tradeoff analysis method. In: Proceedings. Fourth IEEE International Conference on Engineering of Complex Computer Systems (cat. no. 98ex193), pp. 68–78. IEEE (1998)
14. Kühne, T.: On model compatibility with referees and contexts. Softw. Syst. Model. **12**(3), 475–488 (2013)
15. Kumar, P., Gupta, G.P., Tripathi, R.: PEFL: deep privacy-encoding based federated learning framework for smart agriculture. IEEE Micro **42**, 33–40 (2021)
16. Li, L., Xu, W., Chen, T., Giannakis, G.B., Ling, Q.: RSA: byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 1544–1551 (2019)
17. Liu, J., et al.: From distributed machine learning to federated learning: a survey. Knowl. Inf. Syst. **64**(4), 885–917 (2022). https://doi.org/10.1007/s10115-022-01664-x
18. Matei, O., Anton, C., Bozga, A., Pop, P.: Multi-layered architecture for soil moisture prediction in agriculture 4.0. In: Proceedings of International Conference on Computers and Industrial Engineering, CIE, vol. 2, pp. 39–48 (2017)
19. Matei, O., Anton, C., Scholze, S., Cenedese, C.: Multi-layered data mining architecture in the context of internet of things. In: 2017 IEEE 15th International Conference on Industrial Informatics (INDIN), pp. 1193–1198. IEEE (2017)
20. Matei, O., Erdei, R., Moga, A., Heb, R.: A serverless architecture for a wearable face recognition application. In: Del Bimbo, A., et al. (eds.) ICPR 2021. LNCS, vol. 12667, pp. 642–655. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-68787-8_46
21. Olson, R.S., Moore, J.H.: TPOT: a tree-based pipeline optimization tool for automating machine learning. In: Workshop on Automatic Machine Learning, pp. 66–74. PMLR (2016)
22. Pulido-Gaytan, L.B., Tchernykh, A., Cortés-Mendoza, J.M., Babenko, M., Radchenko, G.: A survey on privacy-preserving machine learning with fully homomorphic encryption. In: Nesmachnow, S., Castro, H., Tchernykh, A. (eds.) CARLA 2020. CCIS, vol. 1327, pp. 115–129. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-68035-0_9
23. Siau, K., Wang, W.: Building trust in artificial intelligence, machine learning, and robotics. Cutter Bus. Technol. J. **31**(2), 47–53 (2018)
24. Vishal, P., Bhattacharya, S.: Application of the pareto principle in rapid application development model. Citeseer (2013)
25. Vogel, C.: SAAM (software architecture analysis method). Universität Karlsruhe, p. 1 (2008)

26. Yang, Q., Liu, Y., Cheng, Y., Kang, Y., Chen, T., Han, Yu.: Federated learning. Synth. Lect. Artif. Intell. Mach. Learn. **13**(3), 1–207 (2019)
27. Yin, D., Chen, Y., Kannan, R., Bartlett, P.: Byzantine-robust distributed learning: towards optimal statistical rates. In: International Conference on Machine Learning, pp. 5650–5659. PMLR (2018)

Author Index

P. García Bringas et al. (Eds.): SOCO 2022, LNNS 531, pp. 667–669, 2023.
https://doi.org/10.1007/978-3-031-18050-7