

Lecture Notes in Networks and Systems 957

Héctor Quintián · Emilio Corchado ·
Alicia Troncoso Lora · Hilde Pérez García ·
Esteban Jove · José Luis Calvo Rolle ·
Francisco Javier Martínez de Pisón ·
Pablo García Bringas · Francisco Martínez Álvarez ·
Álvaro Herrero Cosío · Paolo Fosci *Editors*

International Joint Conferences

17th International Conference
on Computational Intelligence in
Security for Information Systems
(CISIS 2024) and 15th International
Conference on European Transnational
Education (ICEUTE 2024)

Series Editor

Janusz Kacprzyk , *Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland*

Advisory Editors

Fernando Gomide, *Department of Computer Engineering and Automation—DCA, School of Electrical and Computer Engineering—FEEC, University of Campinas—UNICAMP, São Paulo, Brazil*

Okyay Kaynak, *Department of Electrical and Electronic Engineering, Bogazici University, Istanbul, Türkiye*

Derong Liu, *Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, USA*

Institute of Automation, Chinese Academy of Sciences, Beijing, China

Witold Pedrycz, *Department of Electrical and Computer Engineering, University of Alberta, Alberta, Canada*

Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

Marios M. Polycarpou, *Department of Electrical and Computer Engineering, KIOS Research Center for Intelligent Systems and Networks, University of Cyprus, Nicosia, Cyprus*

Imre J. Rudas, *Óbuda University, Budapest, Hungary*

Jun Wang, *Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong*

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 worldwide 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, EI Compendex, 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).

Héctor Quintián · Emilio Corchado ·
Alicia Troncoso Lora · Hilde Pérez García ·
Esteban Jove · José Luis Calvo Rolle ·
Francisco Javier Martínez de Pisón ·
Pablo García Bringas ·
Francisco Martínez Álvarez ·
Álvaro Herrero Cosío · Paolo Fosci
Editors

International Joint Conferences

17th International Conference on
Computational Intelligence in Security
for Information Systems (CISIS 2024)
and 15th International Conference
on European Transnational Education
(ICEUTE 2024)



Springer

Editors

Héctor Quintián 
University of A Coruña
Ferrol, La Coruña, Spain

Alicia Troncoso Lora 
Pablo de Olavide University
Seville, Spain

Esteban Jove
University of A Coruña
Ferrol, La Coruña, Spain

Francisco Javier Martínez de Pisón 
University of La Rioja
Logroño, Spain

Francisco Martínez Álvarez 
Pablo de Olavide University
Seville, Spain

Paolo Fosci 
University of Bergamo
Dalmine, Bergamo, Italy

Emilio Corchado 
University of Salamanca
Salamanca, Spain

Hilde Pérez García 
University of León
León, Spain

José Luis Calvo Rolle
University of A Coruña
A Coruña, Spain

Pablo García Bringas 
University of Deusto
Bilbao, Spain

Álvaro Herrero Cosío 
University of Burgos
Burgos, Spain

ISSN 2367-3370

ISSN 2367-3389 (electronic)

Lecture Notes in Networks and Systems

ISBN 978-3-031-75015-1

ISBN 978-3-031-75016-8 (eBook)

<https://doi.org/10.1007/978-3-031-75016-8>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

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

If disposing of this product, please recycle the paper.

Preface

This volume of Lecture Notes in Networks and Systems contains accepted papers presented at the 17th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2024) and the 15th International Conference on European Transnational Education (ICEUTE 2024), which were held in the beautiful city of Salamanca, Spain, in October 2024.

The aim of the CISIS 2024 conference is to offer a meeting opportunity for academic and industry-related researchers belonging to the various vast communities of Computational Intelligence, Information Security, and Data Mining. The need for intelligent, flexible behavior by large, complex systems, especially in mission-critical domains, is intended to be the catalyst and the aggregation stimulus for the overall event.

After a peer-review process, the CISIS 2024 International Program Committee selected 24 papers published in these conference proceedings. In this edition, one special session was organized: Artificial Intelligence for Protecting the Internet of Things.

The aim of the ICEUTE 2024 conference is to offer a meeting point for people working on transnational education within Europe. It provides a stimulating and fruitful forum for presenting and discussing the latest works and advances in transnational education within European countries. In this edition, one special session was organized: Innovation in Computer Science Higher Education.

In the case of ICEUTE 2024, the International Program Committee selected 9 papers, which are also published in these conference proceedings.

The selection of papers was extremely rigorous to maintain the high quality of the conferences. 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 CISIS and ICEUTE conferences would not exist without their help.

CISIS 2024 and ICEUTE 2024 enjoyed outstanding keynote speeches by distinguished guest speakers: Prof. Ajith Abraham at Bennett University (India) and Prof. Paulo Leitão at Polytechnic Institute of Bragança (Portugal).

CISIS 2024 has teamed up with “Logic Journal of the IGPL” (Oxford University Press) for a suite of special issues, including selected papers from CISIS 2024.

Particular thanks goes as well to the conference’s main sponsors, Startup Olé, the CIBER-OLÉ project (within the National Cybersecurity Industry Promotion Program, framed within the INCIBE Emprende program and financed by INCIBE and the University of Salamanca), 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, the Program Committees, and the Local Organizing Committee members for their hard and

highly valuable work, which has contributed to the success of the CISIS and ICEUTE 2024 events.

October 2024

Héctor Quintián
Emilio Corchado
Alicia Troncoso Lora
Hilde Pérez García
Esteban Jove
José Luis Calvo Rolle
Francisco Javier Martínez de Pisón
Pablo García Bringas
Francisco Martínez Álvarez
Álvaro Herrero Cosío
Paolo Fosci

CISIS 2024

Organization

General Chair

Emilio Corchado University of Salamanca, Spain

Program Committee Chairs

Héctor Quintián	University of A Coruña, Spain
Emilio Corchado	University of Salamanca, Spain
Alicia Troncoso Lora	Pablo de Olavide University, Spain
Hilde Pérez García	University of León, Spain
Esteban Jove	University of A Coruña, Spain
José Luis Calvo Rolle	University of A Coruña, Spain
Francisco Javier Martínez de Pisón	University of La Rioja, Spain
Pablo García Bringas	University of Deusto, Spain
Francisco Martínez Álvarez	Pablo de Olavide University, Spain
Álvaro Herrero Cosío	University of Burgos, Spain
Paolo Fosci	University of Bergamo, Italy

Program Committee

Adam Wójtowicz	Poznań University of Economics and Business, Poland
Agustín García Fischer	University of A Coruña, Spain
Agustín Martín Muñoz	Spanish National Research Council (CSIC), Spain
Alberto Peinado	University of Málaga, Spain
Álvaro Michelena Grandío	University of A Coruña, Spain
Anabel Díaz Labrador	University of A Coruña, Spain
Andrés Pérez Zabala	AZTERLAN, Basque Research and Technology Alliance (BRTA), Spain
Angel Martín Del Rey	University of Salamanca, Spain
Antonio Díaz-Longueira	University of A Coruña, Spain
Borja Sanz	University of Deusto, Spain
Carlos Pereira	ISEC, Portugal
Carmen Benavides	University of León, Spain

Ciprian Pungila	West University of Timișoara, Romania
Daniel Feito Pin	University of A Coruña, Spain
Daniel Urda	University of Burgos, Spain
Darius Galis	West University of Timisoara, Romania
Eduardo Solteiro Pires	UTAD University, Portugal
Eneko Osaba	TECNALIA Research & Innovation, Spain
Enrique Onieva	University of Deusto, Spain
Esteban Jove	University of A Coruña, Spain
Fernando Ribeiro	EST, Portugal
Fernando Tricas	University of Zaragoza, Spain
Francisco Martínez-Álvarez	Pablo de Olavide University, Spain
Francisco Zayas-Gato	University of A Coruña, Spain
Francisco Álvarez Terribas	University of Granada, Spain
Gabriele Olieri	Hamad Bin Khalifa University, Qatar
Guillermo Morales-Luna	CINVESTAV-IPN, Mexico
Hugo Sanjurjo-González	University of Deusto, Spain
Hugo Scolnik	ARSAT SA, Argentina
Igor Santos-Grueiro	HP Labs, UK
Ioannis Sorokos	Fraunhofer IESE, Germany
Isaias Garcia	University of León, Spain
Javier Nieves	Azterlan, Spain
Jesús Díaz-Verdejo	University of Granada, Spain
Joaquín Gaspar Medina Arco	University of Granada, Spain
Jose Barata	NOVA University Lisbon, Portugal
José Aveira Mata	Research Institute of Applied Sciences in Cybersecurity (RIASC), Spain
Jose A. Onieva	University of Malaga, Spain
José Antonio Gómez Hernández	University of Granada, Spain
Jose Carlos Metrolho	IPCB, Portugal
Jose Luis Calvo-Rolle	University of A Coruña, Spain
José Luis Casteleiro-Roca	University of A Coruña, Spain
José Luis Imaña	Complutense University of Madrid, Spain
Jose M. Molina	University Carlos III of Madrid, Spain
Jose Manuel Lopez-Gude	University of the Basque Country, Spain
Josep Ferrer	University of the Balearic Islands, Spain
Juan J. Gude	University of Deusto, Spain
Juan Jesús Barbarán	University of Granada, Spain
Lidia Sánchez-González	University of León, Spain
Manuel Castejón-Limas	University of León, Spain
Manuel Graña	University of the Basque Country, Spain
Manuel Rubiños Trelles	University of A Coruña, Spain
Margarita Robles Carrillo	University of Granada, Spain

Marta María Álvarez Crespo	University of A Coruña, Spain
Michał Wozniak	Wroclaw University of Technology, Poland
Michał Choraś	Poland and Bydgoszcz University of Science and Technology, Poland
Miriam Timiraos Díaz	University of A Coruña, Spain
Ovidiu Cosma	Technical University Cluj Napoca, Romania
Paula Patricia Arcano Bea	University of A Coruña, Spain
Pedro Hecht	University of Buenos Aires, Argentina
Petrica Pop	Technical University of Cluj-Napoca, Romania
Rafael Alvarez	University of Alicante, Spain
Rafael Corchuelo	University of Seville, Spain
Rafael A. Rodríguez-Gómez	University of Granada, Spain
Raúl Durán	University of Alcalá, Spain
Robert Burduk	Wroclaw University of Technology, Poland
Roberto Casado-Vara	University of Burgos, Spain
Roberto Magán-Carrión	University of Granada, Spain
Rudolf Erdei	Holisun SRL, Romania
Salvador Alcaraz	Miguel Hernandez University, Spain
Sorin Stratulat	Université de Lorraine, France
Wilson Rojas	El Bosque University, Colombia

CISIS 2024: Special Sessions

Artificial Intelligence for Protecting the Internet of Things

Program Committee

Álvaro Herrero (Organizer)	University of Burgos, Spain
Daniel Urda (Organizer)	University of Burgos, Spain
Félix Iglesias (Organizer)	TU Wien, Spain
Joaquim Gonçalves	IPCA, Portugal
Nuno Lopes	Polytechnic Institute of Cavado and Ave, Portugal

CISIS 2024 Organizing Committee Chairs

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

CISIS 2024 Organizing Committee

Álvaro Herrero Cosio	University of Burgos, Spain
José Luis Calvo Rolle	University of A Coruña, 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
Antonio Javier Díaz Longueira	University of A Coruña, Spain

ICEUTE 2024

Organization

General Chair

Emilio Corchado

University of Salamanca, Spain

Program Committee Chairs

Héctor Quintián	University of A Coruña, Spain
Emilio Corchado	University of Salamanca, Spain
Alicia Troncoso Lora	Pablo de Olavide University, Spain
Hilde Pérez García	University of León, Spain
Esteban Jove	University of A Coruña, Spain
José Luis Calvo Rolle	University of A Coruña, Spain
Francisco Javier Martínez de Pisón	University of La Rioja, Spain
Pablo García Bringas	University of Deusto, Spain
Francisco Martínez Álvarez	Pablo de Olavide University, Spain
Álvaro Herrero Cosío	University of Burgos, Spain
Paolo Fosci	University of Bergamo, Italy

Program Committee

Alessandra Raffaetà	Universita' Ca' Foscari Venezia, Italy
Álvaro Michelena Grandío	University of A Coruña, Spain
Anabel Díaz Labrador	University of A Coruña, Spain
Ana Loureiro	Polytechnic Institute of Santarém, Portugal
Ana Rosa Pereira Borges	Coimbra Polytechnic Institute, Portugal
Antonio Díaz-Longueira	University of A Coruña, Spain
Antonio Morales-Estebaran	University of Seville, Spain
Carlos Pereira	ISEC, Portugal
Carmen Benavides	University of León, Spain
Daniel Feito Pin	University of A Coruña, Spain
Daniela Zaharie	West University of Timisoara, Romania
Dragan Simic	University of Novi Sad, Serbia

Eduardo Solteiro Pires	UTAD University, Portugal
Eloy Irigoyen	University of the Basque Country, Spain
Elsa Rodrigues	Polytechnic Institute of Beja, Portugal
Esteban Jove	University of A Coruña, Spain
Francisco Martínez-Álvarez	Pablo de Olavide University, Spain
Francisco Zayas-Gato	University of A Coruña, Spain
Henrique Gil	Polytechnic Institute of Castelo Branco, Portugal
Hugo Sanjurjo-González	University of Deusto, Spain
Isabel Sofia Brito	Polytechnic Institute of Beja, Portugal
Isaias Garcia	University of León, Spain
J. David Nuñez-Gonzalez	University of Basque Country, Spain
Jairo Ortiz-Revilla	University of Burgos, Spain
João Freitas	NOVA University Lisbon, Portugal
Joao Paulo Barros	Polytechnic Institute of Beja, Portugal
Jorge Barbosa	ISEC - Instituto Superior de Engenharia de Coimbra, Portugal
Jose Luis Calvo-Rolle	University of A Coruña, Spain
José Luis Casteleiro-Roca	University of A Coruña, Spain
Jose Manuel Lopez-Guede	University of the Basque Country, Spain
José Manuel Galán	University of Burgos, Spain
José-Lázaro Amaro-Mellado	University of Seville, Spain
Juan Pavón	Complutense University of Madrid, Spain
Juan J. Gude	University of Deusto, Spain
Julián Estévez	University of the Basque Country, Spain
Laura Fernández-Robles	University of León, Spain
Laura Melgar-García	Polytechnic University of Madrid, Spain
Lidia Sánchez-González	University of León, Spain
Luis Alfonso Fernández Serantes	FH-Joanneum University of Applied Sciences, Spain
Manuel Castejón-Limas	University of León, Spain
Manuel Rubiños Trelles	University of A Coruña, Spain
María Fernanda Gambarini	Francisco de Vitoria University, Spain
Maria Jose Marcelino	University of Coimbra, Portugal
Maria Victoria Requena	University of Seville, Spain
Marta María Álvarez Crespo	University of A Coruña, Spain
Miguel Carriegos	RIASC, Spain
Miguel Ángel Queiruga-Díos	University of Burgos, Spain
Miriam Timiraos Díaz	University of A Coruña, Spain
Paola Clara Leotta	University of Catania, Italy
Paula Patricia Arcano Bea	University of A Coruña, Spain
Paulo Moura Oliveira	University of Trás-os-Montes and Alto Douro (UTAD), Portugal

Sergio Freitas	University of Brasília, Brazil
Soraya Muñoz Pérez	Francisco de Vitoria University, Spain
Sorin Stratulat	Université de Lorraine, France
Vitor Rocio	Universidade Aberta, Portugal

ICEUTE 2024: Special Sessions

Innovation in Computer Science Higher Education

Program Committee

Ángela R. Troncoso-García (Organizer)	Pablo de Olavide University, Spain
David Gutiérrez-Avilés (Organizer)	University of Seville, Spain
José F. Torres (Organizer)	Pablo de Olavide University, Spain
María Martínez-Ballesteros (Organizer)	University of Seville, Spain
Andrés Manuel Chacón Maldonado	Pablo de Olavide University, Spain
Belén Vega Márquez	University of Seville, Spain
Laura Melgar-García	Polytechnic University of Madrid, Spain
Manuel Jesús Jiménez Navarro	Pablo de Olavide University, Spain

ICEUTE 2024 Organizing Committee Chairs

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

ICEUTE 2024 Organizing Committee

José Luis Calvo Rolle	University of A Coruña, 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
Antonio Javier Díaz Longueira	University of A Coruña, Spain
Paula Patricia Arcano Bea	University of A Coruña, Spain
Manuel Rubiños Trelles	University of A Coruña, Spain
Marta María Álvarez Crespo	University of A Coruña, Spain

Contents

CISIS: General Track - CISIS Applications

Blockchain-Based Random Selection Among a Private Number of Candidates	3
<i>Idoia Gamiz, Cristina Regueiro, Eduardo Jacob, Oscar Lage, and Juanjo Unzilla</i>	
A Timeline and Analysis of Recent Bitcoin Money-Laundering Detection Methods	15
<i>Rebeca Tonu and Ciprian Pungilă</i>	
An Approach Based on LLMs for Forensic Threat Detection in Autonomous Systems	25
<i>David Sobrín-Hidalgo, Irene González-Fernández, Miguel Ángel González-Santamaría, Adrián Campazas-Vega, Claudia Álvarez-Aparicio, Ángel Manuel Guerrero-Higueras, Francisco Javier Rodríguez-Lera, and Vicente Matellán-Olivera</i>	
Smart Water Meter Network Anomaly and Failure Detection System	35
<i>Lukasz Saganowski and Tomasz Andrysiak</i>	
Design and Implementation of a Fast, Platform-Adaptive, AIS-20/31 Compliant PLL-Based True Random Number Generator on a Zynq 7020 SoC FPGA	45
<i>Oğuz Yayla and Yunus Emre Yılmaz</i>	
A Comparison of Frameworks for Heterogeneous Computing Using High-Performance Pattern-Matching for DNA Biometrics and Digital Forensics	56
<i>Lia Stratan and Ciprian Pungila</i>	
Fuzzing Robotic Software Using HPC and LLM	66
<i>Garnelo Del Río Francisco Borja, Francisco J. Rodríguez Lera, Camino Fernández Llamas, and Vicente Matellán Olivera</i>	
Performance Impact of Strengthening the Accountability and Explainability System in Autonomous Robots	77
<i>Alejandro González-Cantón, Miguel A. González-Santamaría, Enrique Soriano, Gorka Guardiola, and Francisco J. Rodríguez Lera</i>	

Generation of Industrial Protocol-Traffic via Enhanced Wasserstein GAN	88
<i>Mikel Moreno, Lander Segurola, Francesco Zola, Arantza del Pozo, and Iker Pastor</i>	
Cyber Security in Hospitals: A Network-Oriented Model for Behavioural Learning of Employees During Phishing Simulations	98
<i>Debby Bouma, Charlotte Hoffmans, Niek Jan van den Hout, Natalia Zwarts, Jan Treur, and Peter Roelofsma</i>	
Augmenting API Security Testing with Automated LLM-Driven Test Generation	112
<i>Emil Marian Pasca, Rudolf Erdei, Daniela Delinschi, and Oliviu Matei</i>	
Exploring the Use of LLMs to Understand Network Traces	122
<i>Rubén De la Torre Vico, Roberto Magán-Carrión, and Rafael Alejandro Rodríguez-Gómez</i>	
CISIS: General Track - Cyber Attacks, Blockchain and Quantum Computation	
Securing V2X Communication: A Hybrid Deep Learning Approach for Misbehavior Detection	135
<i>Tiago Costa, Cesar Analide, Antonio Costa, and Jose Cunha</i>	
Use of GAN Models and CNN Heatmaps in Malware	145
<i>Xabier Ceballos Diez, Iker González Requejo, Adrián Martínez Navarro, Eduardo Jorge Sanjurjo Aramburu, José Gaviria de la Puerta, and Iker Pastor-López</i>	
Evaluating Sniffers, IDS, and IPS: A Systematic Literature Mapping	157
<i>Raúl Bayón-Martínez, Laura Inyesto-Alonso, Adrián Campazas-Vega, Gonzalo Esteban-Costales, Claudia Álvarez-Aparicio, Ángel Manuel Guerrero-Higueras, and Vicente Matellán-Olivera</i>	
Performance Analysis of NTT Algorithms	168
<i>David García Lleyda, Víctor Gayoso Martínez, Luis Hernández Encinas, Agustín Martín Muñoz, and Óscar Castillo Campo</i>	
Exploring the Landscape of Honeypots in the Fight Against Cyber Threats: A Systematic Mapping of Literature	179
<i>Alberto Miguel-Diez, Rodrigo González-Fernández, Gonzalo Esteban-Costales, Christian Vega-González, Adrián Campazas-Vega, Vicente Matellán-Olivera, and Ángel Manuel Guerrero-Higueras</i>	

Phishing and Spam Prevention Powered by Jetson Nano	191
<i>Jackson Diaz-Gorrin and Pino Caballero-Gil</i>	
Decentralized Edge-Based Detection of Label Flipping Attacks in Federated Learning	201
<i>Nourah AlOtaibi, Muhamad Felemban, and Sajjad Mahmood</i>	
Analysis of Post-Quantum Cryptographic Algorithms: A Systematic Literature Review	212
<i>Francisco Javier de la Puente Secades, Gonzalo Esteban-Costales, Adrián Campazas-Vega, Claudia Álvarez-Aparicio, Vicente Matellán-Olivera, and Ángel Manuel Guerrero-Higueras</i>	
CISIS: Special Session: Artificial Intelligence for Protecting the Internet of Things	
A Comparison of AI-Enabled Techniques for the Detection of Attacks in IoT Devices	227
<i>Eduardo Manuel Cabeza-Lopez, Ruben Ruiz-Gonzalez, Alejandro Merino-Gomez, Leticia Elena Curiel-Herrera, and Jaime Andres Rincon</i>	
Understanding Malware Dynamics in IoT Networks: Dataset Construction Using Mathematical Epidemiology and Complex Networks	237
<i>Leticia Sainz-Villegas, Roberto Casado-Vara, Nuño Basurto, Carlos Cambra, Daniel Urda, and Alvaro Herrero</i>	
Quality of Service Enhancement for IoT-Based Smart Office System Using Ad Hoc On-Demand Distance Vector-Smart Control Ration Control Algorithm (AODV-SRCA)	247
<i>Lawrence Nforh CheSuh, Ramón Ángel Fernández-Díaz, Jose Aveleria-Mata, Martín Bayón-Gutiérrez, Sergio Rubio Martín, and Héctor Alaiz-Moretón</i>	
Cyberattack Detector for Real-Time IoT Devices and Networks	257
<i>Antonio Díaz-Longueira, Álvaro Michelena, Francisco Zayas-Gato, Marta-María Álvarez-Crespo, Óscar Fontenla-Romero, and José Luis Calvo-Rolle</i>	
ICEUTE: General Track	
Conducting Final Programming Exams with Auto-Grading and Code Evaluation Tools	269
<i>José Luis Subirats, Gabriel Luque-Polo, and Rafael M. Luque-Baena</i>	

Educational Metaverse	279
<i>Alberto Martínez-Gutiérrez, Javier Díez-González, Rubén Ferrero-Guillén, Iván Sánchez-Calleja, Pablo Alonso-Diez, Madalena Araújo, and Hilde Perez</i>	
Cybersecurity and Cyber Defense in the Classroom	288
<i>Jorge Barbosa</i>	
ICEUTE: Special Session: Innovation in Computer Science Higher Education	
Optimization of Transport Routes Through a Social Interaction Algorithm-Based Application	301
<i>Diego Ismael Valdivia Alcalá, Ángel Miguel García Vico, and Cristóbal José Carmona del Jesús</i>	
Analysis of School Dropout Rate in Paraguay Using a Machine Learning Approach	312
<i>Diego R. Insfrán-Coronel, Enzo M. Enrique-Sánchez, Federico Beck, Aurelio López-Fernández, and Miguel García-Torres</i>	
Exploring Educational Trends: Specializations in Secondary Education in Paraguay from 2018 to 2021	322
<i>José A. Torres-Báez, Juan B. Torres-Báez, Aurelio Lopez-Fernandez, Francisco A. Gomez-Vela, and Federico J. Beck</i>	
Effects of Language of Instruction in Higher Education	332
<i>Silvia Vázquez-Noguera, Fabiola Martínez, David Becerra-Alonso, Aurelio López-Fernández, Isabel Lopez-Cobo, Perla Sosa, and Miguel García-Torres</i>	
Analysis of Socio-Economic Factors Associated to the Academic Performance in Paraguay	342
<i>Federico Javier Beck and Francisco Gómez-Vela</i>	
Interactive Telegram Bot as a Support Tool for Teaching in Higher Education	352
<i>Juan Luis Suárez-Díaz, Nuria Rodríguez-Barroso, and Guillermo Gómez-Trenado</i>	
Author Index	363

CISIS: General Track - CISIS Applications



Blockchain-Based Random Selection Among a Private Number of Candidates

Idoia Gamiz^{1,2(✉)}, Cristina Regueiro², Eduardo Jacob¹, Oscar Lage², and Juanjo Unzilla¹

¹ Department of Communications Engineering, University of the Basque Country (UPV/EHU), 48013 Bilbao, Bizkaia, Spain

² TECNALIA, BRTA, Bizkaia Science and Technology Park, 700, 48160 Derio, Bizkaia, Spain
idoia.gamiz@tecnalia.com

Abstract. Random selection processes are fundamental across diverse domains, requiring transparency and fairness in execution. However, some contexts, such as sensitive ecosystems like healthcare or education, also require an additional degree of privacy by avoiding information leakage about the number of candidates for the selection. Despite this, existing literature lacks comprehensive solutions that maintain this privacy while upholding transparency and fairness. To address this gap, the study proposes a platform that combines blockchain and Homomorphic Encryption. The main contributions include a detailed definition of the architecture and the operational procedures, and an in-depth exploration of the proposed Random Number Generator function. The study introduces a homomorphic approach for confidentially computing candidate counts and explores methods for homomorphically constraining random numbers within a private range. By combining blockchain and Homomorphic Encryption, this research offers a robust solution for transparent and fair random selection while maintaining the number of candidates private.

Keywords: Selection · Blockchain · Homomorphic Encryption · Privacy

1 Introduction

The process of random selection is a fundamental operation across various domains [1]. Depending on the use case, it can serve both as an independent value, like in a random raffle, and as an integral component within larger algorithms, as in cryptographic protocols. In both cases, ensuring transparency and fairness within this operation is essential, to avoid relying on a trusted third party. This study delves deeper into a specific use case where maintaining the number of candidates for a particular selection private is crucial.

This necessity becomes particularly evident in scenarios involving sensitive data, such as education or medical information. Imagine, for example, a random

selection process among students with low-income inside a school to receive some limited free catch-up lessons. The school will probably want to avoid making public the number of students with low-income to prevent prejudice or defamation regarding the economic situation of certain individuals, which is particularly important when dealing with minors, while still being fair in the random process. Another example could be based on a scenario in which a clinical trial is conducted for a rare disease. The objective could be to determine the initial participants for testing without revealing the total number of participants, which may be confidential. This is crucial to maintain privacy and avoid premature conclusions about potential relationships between the disease and environmental factors, such as workplace conditions. Preventing the premature release of this information helps to avoid unwarranted panic or misinterpretation. Once thorough and accurate conclusions are drawn, the information can then be made public. Similarly, in cryptographic protocols or auction systems, protecting candidate numbers prevents adversarial collusion. A cryptographic protocol may stipulate the requirement of maintaining security against a specified threshold of coalition adversaries. In auction mechanisms, a random leader could be selected to start the process. In this case, protecting the number of candidates prevents collusion and bid manipulation.

In centralized scenarios, where the Selector is the only one who possesses knowledge of the participant count, maintaining this number private is straightforward. However, relying solely on a central authority for the selection process requires complete trust to ensure transparency and fairness are not compromised. Thus, achieving transparency, fairness, and privacy becomes essential. The adoption of blockchain technology emerges as a reliable solution for such selections. Moreover, advanced encryption techniques, such as Homomorphic Encryption (HE) [2], can be utilized to ensure privacy during computation over encrypted data. This research explores the combination of these technologies.

The primary aim of this article is to present a platform capable of randomly selecting a candidate with transparency and fairness, while maintaining the number of candidates private. To achieve this, the main contributions are:

- Definition of a complete platform. This includes an architectural design of the platform and the definition of operational procedures for specific selections, including a step-by-step scheme guiding users through the process.
- A detailed proposal of the Random Number Generator function, which serves as the core functionality. The process involves three steps: generating a random number transparently within blockchain (or within an oracle), computing the number of candidates in encrypted form, and mapping the previously generated random number into the calculated private range.

The paper is structured as follows. Section 2 introduces recent work in the literature. Section 3 proposes and describes the platform. Finally, Sect. 4 concludes the paper and discusses avenues for future work.

2 Related Work

In the past, random selections were often conducted by relying on a trusted third party, which allowed that party to keep the number of candidates private from

the rest. However, this approach does not guarantee transparency and fairness, as the process depend on the honesty of the third party. The use of blockchain ensures that the results are truly random and cannot be manipulated, positioning blockchain as a promising solution. Table 1 classifies related work based on their ability to make a transparent and fair random selection, while maintaining the number of candidates private.

Table 1. Related work.

Ref.	Technique	Transp.	Fairness	Privacy
Baseline	Random [Trusted third party]	✗	✗	✓
[3, 5, 6]	Blockchain + Random	✓	✓	✗
[7, 8]	Blockchain+Random+HE [Random Beacon]	✓	✓	~
This work	Blockchain+Random+HE [Private Counts]	✓	✓	✓

Several works propose methods for computing random numbers in blockchain, with different degrees of randomness. Some works [3] propose to use the block time or block-hash to initialize the PRNG directly in the SC. However, these approaches are susceptible to manipulation by adversarial miners [4]. Alternatively, secure random number generation can be achieved by sending a seed to an off-chain oracle, such as Chainlink VRF [4, 5], which provides tamper-proof and verifiable randomness, or NativeVRF [6]. While these approaches offer valuable insights, none address the specific challenge of calculating a random number in a private range. Moreover, some works integrate HE into blockchain to create a Random Beacon. HERB [7] and a similar approach by Nguyen-Van et al. [8] employ HE for collaborative private random number generation. Even if they combine HE with blockchain, they do not use HE to protect the range of the random number, they use it to create a private random number collaboratively. All this work could be applied as a part of this work, but they do not tackle the whole problem as such.

In summary, existing literature provides useful approximations, but none directly address the challenge of preserving the number of candidates private. Thus, while these works contribute to the broader landscape, this study fills a critical gap by proposing a novel platform that addresses this specific requirement.

3 Proposal

3.1 Requirements

To ensure that the proposal addresses the use case introduced in Sect. 1, it must meet a series of requirements. This section introduces these conditions and explains why certain technologies have been selected as the basis for the solution.

- Transparency. In the context of a random selection, *transparency* refers to ensuring that the selection process is visible and understandable to all parties involved, allowing participants to verify the integrity of the process and confirm there is no manipulation. This includes making the process auditable, without the need of relying on a trusted third party.
- Fairness. In the same context, *fairness* involves providing all candidates with the same probability of being selected, ensuring there are no biases or preferences in the process. It also means executing the process impartially, free from external influences that could alter the results, and maintaining equitable conditions for all participants to ensure equal chances.
- Privacy. In this case, *privacy* refers to maintaining the number of candidates private. As explained before, in scenarios involving sensitive data, such as education or medical information, maintaining the number of candidates confidential is essential.

To ensure *transparency* and *fairness*, this work proposes using blockchain. The decision to leverage blockchain as the backbone for leader selection is based on its characteristics to ensure a trustworthy orchestration. In fact, blockchain is a decentralized and transparent technology that records all transactions on a distributed network, ensuring data integrity and the immutability of records. This allows *transparency*, by ensuring that the selection process is visible to all participants and by eliminating the need for a trusted third party. Additionally, the decentralized nature of blockchain minimizes the influence of external factors, ensuring equitable conditions for all participants and maintaining *fairness*.

For *privacy* concerns, the proposal uses Homomorphic Encryption (HE). In scenarios where maintaining the number of candidates private is paramount, the blockchain approach lacks privacy by default. To address this limitation, encryption can be a good approach to safeguard stored information. Therefore, HE is used to compute over encrypted data and maintain the number of candidates private.

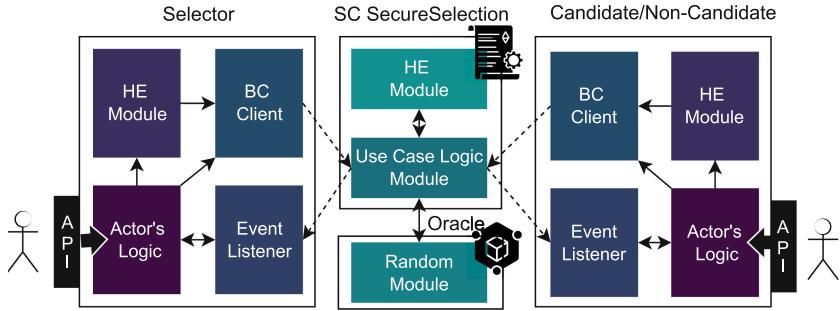
By combining HE with blockchain, there are no alterations to the fundamental properties of the blockchain, maintaining *transparency* and *fairness*. Moreover, the data in the blockchain can be encrypted, effectively addressing the privacy apprehensions often linked with public blockchains [9].

3.2 Architecture

There are different alternatives in the literature for creating random numbers in blockchain, as presented in Sect. 2. While a comprehensive analysis of different options is planned as future work, the architecture and operation of the platform is explained for generating the random number in an oracle, which is not susceptible to manipulation by adversarial miners [4].

The platform comprises three types of users, a Smart Contract (SC) deployed on the blockchain and an external Oracle for generating random numbers. The components for each element are shown in Fig. 1 and are explained below.

- SC SecureSelection. It allows interaction between the participants.

**Fig. 1.** Architecture

- Use Case Logic Module. It is essential for defining the rules and logic of the selection process.
- HE module. It is used for performing private operations within the SC, such as private computation of the total number of candidates and the operation to map the random number into that range.
- Oracle. It allows the generation of random numbers off-chain.
 - Random Module. It is used to generate an immutable and verifiable random number.
- Selector (S). The Selector is the person that initiates a *Selection*. S needs the following components:
 - API. It is used to interact with the platform.
 - Actor's Logic. It is essential for defining the logic of the actor and interacting with the rest of the modules.
 - BC client-wallet. It is used to interact with the blockchain.
 - Event listener. It is used to monitor relevant events on the blockchain.
 - HE module. It is used for performing private operations.
- Candidates/Non-Candidates. The Candidates are the participants that decide to present themselves as a candidate for a particular *Selection*. The Non-Candidates are the ones who are listening to the events but decide not to present. Depending on the use case, these participants may receive incentives through various mechanisms. It is worth emphasizing that not all Non-Candidates are required to participate.

The Selector, the Candidates and the Non-Candidates share the same components. The reason is that individuals can change the role on different *Selections*. However, depending on the role, the components would be used for performing some operations or others. From now on, n represents the number of Candidates, m the number of Non-Candidates and $N = n + m$.

3.3 Operation

The whole process is orchestrated by the SC SecureSelection. Tables 2 and 3 summarize the functions and the events that are used, respectively. The steps are explained below and are illustrated in Fig. 2.

Table 2. Function of SC SecureSelection.

Function	Description	Input Param.	Return	Caller
publishSelection()	Publishes the type of Selection and emits event E1	typeofSelection HEscheme publicKey timeLimit	—	S
registerCandidate()	Registers if they want or no to be a candidate (encrypted)	privateInput	—	C, NC
startSelection()	Returns a dictionary with the private inputs	—	privateDict	S
createRandom()	Initializes the Secure Random Number Generator and emit event E2 when it is finished	reassignedDict	—	S
[Optional] assignmentError()	Notifies and erroneous assignment and emits event E3	errorMessage	—	C, NC

Table 3. Events of SC SecureSelection.

ID	Event	Description	Parameters	Listener
E1	SelectionRequest()	Broadcast information about the type of Selection	typeofSelection HEscheme publicKey timeLimit	C, NC
E2	SecureResult()	Broadcast the selected result	selectedAddress privateN reassignedDict	C, NC
E3	AbortSelection()	Broadcast the errorMessage	errorMessage	C, NC

0. All the participants are enrolled in the SC via the “Event listener”.
 1. *S* registers a *Selection* by calling the function **publishSelection()** by the “BC client”. The input parameters of the function are:
 - typeofSelection: An explanation of the characteristics of the *Selection*, including how the candidates must represent their candidature. Participants must understand how to represent and encrypt their candidacy status with a degree of randomness. This is crucial to prevent other participants from deducing it by examining the input data on the blockchain [See Sect. 3.4 for further details].
 - HEscheme: The HE scheme that they must use during the process.
 - publicKey: The public key of *S*, pks , to encrypt their input, after representing the “Yes” or “No” as explained in the typeofSelection. This key is stored in the SC to later conduct private operations with the “HE module”.
 - timeLimit: A time limit for the registration of the candidacy.
- The function emits the event SelectionRequest, but does not return any value.

2. The participants that are subscribed would automatically receive the notification of the event **SelectionRequest** by the “Event listener”. The parameters of the event are the input parameters of the function publishSelection: typeOfSelection, HEscheme, publicKey and timeLimit.
3. The participants that are interested, the Candidates, represent “Yes” as explained and encrypt their representation with pk_S . Similarly, the participants that are not interested, the Non-Candidates, represent and encrypt “No”.
4. The Candidates and the Non-Candidates invoke the function **registerCandidate()** using their “BC client”. The input parameter of the function is:
 - privateInput: A *string* with the encrypted representation of “Yes” or “No” (for Candidates and Non-Candidates, respectively).

The inputs are kept in a dictionary (or a mapping) that matches the BC address of each participant with their register. This function does not have any return value.
5. When the timeLimit is exceeded, S invokes the function **startSelection()**. The function has no input, but has a return value:
 - privateDict: The dictionary where the privateInput’s of all the participants are kept.
6. S decrypts each privateInput and then calculates the total number of candidates, n , based on the private representation. Subsequently, S randomly assigns an encrypted integer within the range $[1, n]$ to each Candidate and a random number, always outside the range $[1, n]$, to each Non-Candidate. Finally, S creates a new dictionary that maps the BC address to the assigned encrypted value.
7. S invokes the function **createRandom()** to initialize the Random Number Generator in blockchain. It is computed through four steps:
 - (a) Generate a random number in the Oracle by the “Random Module”.
 - (b) Privately calculate the number of candidates, $enc_{pk_S}(n)$, using the “HE Module”.
 - (c) Constrain the random number to the range $[1, enc_{pk_S}(n)]$ using the “HE Module”.
 - (d) Verify which is the BC linked with that value using the “HE Module”.

The function has an input parameter:

 - reassignedDict: The dictionary where the assigned numbers for all Candidates and Non-Candidates are kept.

The function has a return value that would be emitted by the event SecureResult:

 - selectedAddress: The BC address associated with the random number in the range $[1, enc_{pk_S}(n)]$.

Note: The steps of this function are further explained in Sect. 3.4.
8. The participants receive the event **SecureResult** by the “Event listener”. The parameters are: selectedAddress, privateN ($enc_{pk_S}(n)$) and reassignedDict. The last two are emitted for verification purposes.
 - * Optionally, those participants that do not trust S can use their “HE module” to check if their assigned encrypted number is correct:

- For Candidates: They must homomorphically prove that $\text{assignedNumber} \in [1, \text{enc}_{pk_S}(n)]$
- For Non-Candidates: They must homomorphically prove that $\text{assignedNumber} \notin [1, \text{enc}_{pk_S}(n)]$

In case there is any unusual behaviour, the participant can invoke the function **assignmentError()** that broadcasts the event **AbortSelection** to notify the rest of the participants.

3.4 Secure Random Generator

This section gives a more detailed explanation of the random generation procedure, which is computed by the function **createRandom()**. The process is divided into four steps:

1. Random number in blockchain: r . This random number is calculated in clear and it falls within the range $[0, p)$, where p represents the supported range of numbers depending on the specific method used for generation.
2. Calculate the number of Candidates: $\text{enc}_{pk_S}(n)$. The Candidates and the Non-Candidates publish their encrypted representation on the blockchain (as explained in Step 4, Sect. 3.3). Therefore, an HE operation is needed to discover which inputs represent a “Yes” and count the number of Candidates.
3. Map the random number. The number must lie in the range $[1, \text{enc}_{pk_S}(n)]$, so it is necessary to define an HE operation to map the random number calculated in Step 1, r , to the range $[1, \text{enc}_{pk_S}(n)]$.
4. Link the BC address. The selected random number must be linked with a BC address in the reassignedDict, by computing homomorphic comparisons.

For Steps 2–4 it is necessary to create SCs with HE. There are various options in the literature, from implementing HE operations from scratch [10], to using existing libraries such as fhEVM (Zama) [11], smartFHE [12], and ZeeStar [13].

During the next subsection, the work analyses some options for each step. The suitability of each option may vary depending on the selected blockchain and the HE scheme.

Random Number in Blockchain. Generating a preliminary random number, r , within the blockchain is crucial for ensuring its unpredictability and transparency to all the participants. This section outlines the procedure for the well-known Chainlink VRF [4, 5]. In this case, $p = 2^{256}$, so the random number would fall within the range $[0, 2^{256})$ and the steps are:

1. The SC SecureSelection initiates a request for randomness.
2. Chainlink generates a random value and a cryptographic proof of how that value was determined. This proof is sent to the VRF contract.
3. The VRF contract verifies the authenticity of the generated randomness.
4. Upon verification, the SC SecureSelection receives the verified randomness.

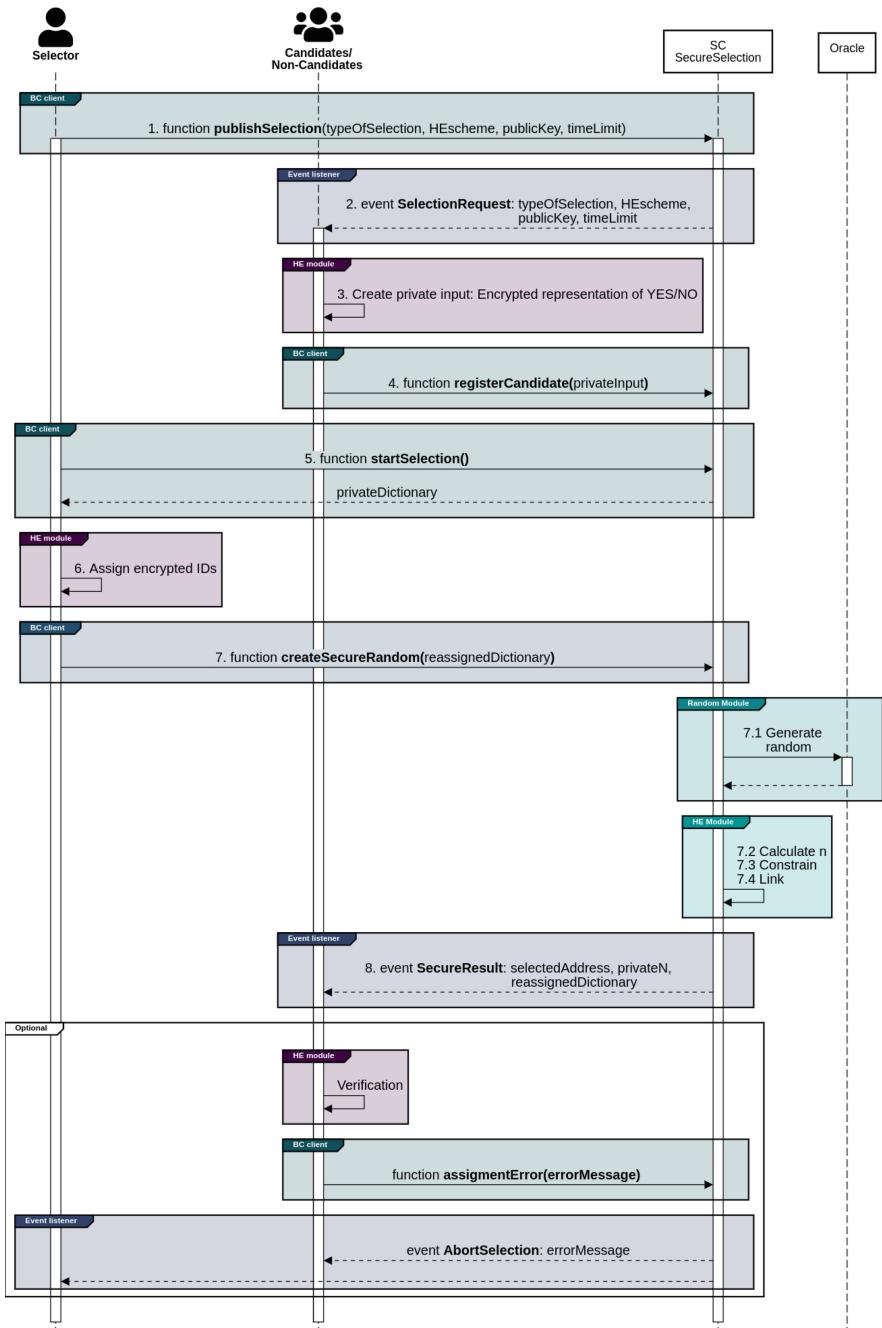


Fig. 2. Sequence diagram of the operations involved in the platform (For the case where the random number is generated in the SC).

This approach ensures that the proof of randomness is published and validated on-chain before being utilized in the SC SecureSelection. Therefore, the results cannot be manipulated by any single entity including oracle operators, miners, users, or SC developers [5].

Calculate the Number of Candidates. The participants can not simply represent the “No” and the “Yes” as a 0 and a 1, encrypt it and register it on the blockchain. Both the HE scheme and the public key are accessible to everyone. Therefore, a malicious participant can discover an input by simply encrypting the 0 and 1 (there would only exist two types of ciphertexts).

Depending on the supported data type by the HE scheme, this work considers two types of representations. If the scheme allows positive and negative integers, Candidates can provide a positive number and Non-Candidates a negative number (always encrypted). If only positive integers are allowed, it is possible to fix a number l so that Candidates input a number $> l$ and Non-Candidates a number in the range $(0, l]$. To mitigate the risk of adversaries deducing the encrypted number through trial and error, it is advisable to set l as large as possible. A practical approach could involve setting $l = p/2$, ensuring that half of the numbers in the space are available for Candidates to respond affirmatively, while the other half is reserved for Non-Candidates. For simplicity, this section explains the procedure for the first option with an example of $N = 8$ and $n = 5$.

1. The computation starts with the list of privateInput’s of all the candidates.
 $L = [enc_{pk_S}(-2), enc_{pk_S}(3), enc_{pk_S}(5), enc_{pk_S}(1), enc_{pk_S}(-1), enc_{pk_S}(10), enc_{pk_S}(12), enc_{pk_S}(-6)]$
2. The elements of the list are homomorphically compared (greater than) with $enc_{pk_S}(0)$. The result is a new list with 0s and 1s representing True or False.
 $L = L > enc_{pk_S}(0) = [enc_{pk_S}(0), enc_{pk_S}(1), enc_{pk_S}(1), enc_{pk_S}(1), enc_{pk_S}(0), enc_{pk_S}(1), enc_{pk_S}(1), enc_{pk_S}(0)]$
3. The elements of the list are added.
 $sum(L) = enc_{pk_S}(5)$

The obtained result is the number of candidates encrypted with the public key of S , i.e., $enc_{pk_S}(n)$. It is important to note that N is directly related to the number of HE comparisons, which could significantly affect the overall efficiency. For this reason, the system is more suitable for small values of N . For larger values, while feasible, it is important to consider that this could lead to a loss of efficiency, which will vary depending on the specific HE framework.

Map the Random Number. To obtain the desired random number, the previously generated number, r , must be mapped to the range $[1, enc_{pk_S}(n)]$. The $enc_{pk_S}(n)$ was homomorphically calculated in the previous step, so to make the number lie in that range this work proposes to use HE again. To do so:

1. Compute the modulo operation so that the number lies in the range $[enc_{pk_S}(0), enc_{pk_S}(n - 1)]$.
2. Add 1 so that the number lies in the range $[enc_{pk_S}(1), enc_{pk_S}(n)]$.

While computing the modulo operation in HE may pose challenges, existing literature offers insights into schemes that exclusively output integers to handle this problem. Authors in [14] use the HElib implementation of a variant of the BGV scheme. They adopt the “Non-restoring division algorithm” with bit-wise division. Given the public key, the cipher dividend A , and the cipher divisor B the result is the quotient Q and the remainder R . In this case, A would be the encrypted form of the random number generated in the blockchain, $\text{enc}_{pk_S}(r)$, and B would be $\text{enc}_{pk_S}(n)$. After applying the division, it is possible to obtain the remainder R that would lie in the range $[0, \text{enc}_{pk_S}(n) - 1]$. Alternatively, in [15] they present a similar approach, but instead of employing bitwise division, they use integer-wise division. It is important to note that these are just two examples from the literature, and further exploration of alternative approaches will be conducted in future work, particularly after evaluating the most suitable HE scheme for the selected blockchain platform.

Link the BC Address. After obtaining the selected private number in the range $[1, \text{enc}_{pk_S}(n)]$, it must be compared with all the values of the reassigned-Dict. The selected BC address would be the one whose assigned value matches the private random number. This comparison process is executed using HE computations.

4 Conclusions

This work proposes a platform designed to facilitate the random selection of individuals from a pool of candidates, while maintaining the number of candidates private. This necessity is particularly evident when the selection is integrated into larger cryptographic protocols, which may themselves require the confidentiality of the candidate count. Failure to maintain this privacy could empower malicious adversaries to form collusions and gain access to information they should not have. Through the utilization of Homomorphic Encryption and the definition of private input representation and HE operations, the platform ensures that the number of candidates remains private. Moreover, integration with blockchain safeguards transparency and fairness in the selection process.

Several potential future steps can be identified within the context of the ongoing investigation. Among these, the following are notable:

- Conduction of a comprehensive analysis of various combinations of blockchain platforms, HE schemes, and random number modules to identify the most suitable configurations for the defined operations. This analysis will encompass evaluating the performance, security, and compatibility of different combinations to ensure optimal outcomes for the application of this work.
- Development of an incentive mechanism. While Candidates naturally have an interest in the result of the operation, it is equally crucial to incentivize Non-Candidates as their participation is essential for maintaining the platform’s integrity and functionality.

Acknowledgements. This work has been partially funded by Tecnalia and the University of the Basque Country [grant number PIFTEC21/05] and by the Spanish Ministry of Science project TED2021-129975B-C22/ AEI/10.13039/501100011033/ Unión Europea NextGenerationEU/PRTR.

References

1. Kavousi, A., Wang, Z., Jovanovic, P.: SoK: Public Randomness (2023)
2. Yi, X., Paulet, R., Bertino, E.: Homomorphic encryption. In: Homomorphic Encryption and Applications. SCS, pp. 27–46. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-12229-8_2
3. Prajapati, S.: Random number generation on blockchain. In: GAMMASTACK (2020). <https://medium.com/gammastack/random-number-generation-on-blockchain-d3141e1efc2>
4. Breidenbach, L., Cachin, C., Chan, B., et al.: Chainlink 2.0: next steps in the evolution of decentralized oracle networks. In: Chainlink Labs, pp. 1–136 (2021)
5. Chainlink: Chainlink VRF. Documentation (2024). <https://docs.chain.link/vrf>
6. Werapun, W., Karode, T., Suaboot, J., et al.: NativeVRF: a simplified decentralized random number generator on EVM blockchains. In: Systems, p. 326 (2023). <https://doi.org/10.3390/systems11070326>
7. Cherniaeva, A., Shirobokov, I. Y.: Homomorphic encryption random beacon. In: IACR Cryptol, ePrint Arch., p. 1320 (2019). <https://doi.org/10.1109/HPDC.2001.945188>
8. Nguyen-Van, T., Nguyen-Anh, T., et al.: Scalable distributed random number generation based on homomorphic encryption. In: 2019 IEEE International Conference on Blockchain, pp. 572–579 (2019). <https://doi.org/10.1109/Blockchain.2019.00083>
9. Gautham: Homomorphic encryption and smart contracts: privacy and transparency. In: NEWSBTC (2018)
10. SolidityAcademy: Homomorphic Encryption in Blockchain: Safeguarding Data with Mathematical Sorcery. Solidity (2023). <https://medium.com/@solidity101/homomorphic-encryption-in-blockchain-safeguarding-data-with-mathematical-sorcery-cbaa1e668484>
11. Zama: Private smart contracts on the EVM using homomorphic encryption (2023). <https://github.com/zama-ai/fhevm>
12. Solomon, R., Weber, R., Almashaqbeh, G.: smartFHE: Privacy-Preserving Smart Contracts from Fully Homomorphic Encryption (2023). <https://doi.org/10.1109/EuroSP57164.2023.00027>
13. Steffen, S., Bichsel, B., Baumgartner, R. et al.: ZeeStar: private smart contracts by homomorphic encryption and zero-knowledge proofs. In: 2022 IEEE Symposium on Security and Privacy, pp. 179–197 (2022). <https://doi.org/10.1109/SP46214.2022.9833732>
14. Chen, Y., Gong, G.: Integer arithmetic over ciphertext and homomorphic data aggregation. In: 2015 IEEE Conference on Communications and Network Security, CNS 2015, pp. 628–632 (2015). <https://doi.org/10.1109/CNS.2015.7346877>
15. Okada, H., Cid, C., Hidano, S., Kiyomoto, S.: Linear depth integer-wise homomorphic division. In: Blazy, O., Yeun, C.Y. (eds.) WISTP 2018. LNCS, vol. 11469, pp. 91–106. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-20074-9_8



A Timeline and Analysis of Recent Bitcoin Money-Laundering Detection Methods

Rebeca Tonu^(✉) and Ciprian Pungilă

West University of Timisoara, Timișoara, Romania
`{rebeca.tonu,ciprian.pungila}@e-uvt.ro`

Abstract. Money laundering poses a significant threat to the integrity of financial systems worldwide and with the development of blockchain technology, new avenues for illegal activities have emerged. The accelerated evolution of blockchain technology has outpaced the progression of corresponding regulatory frameworks, creating a fruitful environment conductive to financial crimes. Current anti-money laundering (AML) techniques often struggle to adapt to the complex and dynamic nature of blockchain transactions, continuously proving the need to reassess both technological and legislative approaches. This paper undertakes a critical analysis of the methods used to detect illegal actions performed on the Bitcoin blockchain. We discuss detection taxonomy, as well as its challenges and opportunities, while also providing a timeline of past and current research.

Keywords: blockchain · cryptocurrency · financial crime · money laundering · illicit transactions · detection methods · review study

1 Introduction

This paper focuses on exploring money laundering techniques within the virtual cyberspace, with particular focus on the Bitcoin blockchain, conducting a critical analysis by reviewing related works and their approaches to identifying and combating illicit transactions. While the methods proposed herein refer to the specifics of the Bitcoin blockchain, to some degree many approaches could be extended to other blockchains as well. Some transaction aspects (i.e. multiple inputs mapped to multiple outputs) are however specific to the Bitcoin blockchain. At the same time, we propose a detection taxonomy of the detection methods so far, identifying the proper challenges that these come with, as well as expose the opportunities we have identified in the process. Particularly, Sect. 2 discusses a timeline and relevant research, while Sect. 3 focuses on identifying challenges. We discuss conclusions in Sect. 4, and summarize our findings.

2 Related Work

Services offering false documents [1] are becoming more widespread in the cyberspace, with providers creating replicas of ID cards, driver's licenses, and passports that can easily deceive authenticity systems. Additionally, some platforms offer various illegal services [2], such as compromising social media or banking accounts, launching DDoS attacks, spreading ransomware, and stealing data. Privacy-focused networks like Tor provide access to services like Bitcoin Fog [3], which mix Bitcoins from multiple users to obscure transaction traces and maintain anonymity.

J. West et al. [44], presented a timeline dedicated to the findings related to fraud detection-based related work in the span of 2003–2013. In Fig. 1, we analyze and meticulously select the Bitcoin fraud detection-based related work, placing a placeholder for each, in the 2013–2024 time frame. Figure 2 presents a detection taxonomy of the same work as above, based on the methods the authors implemented. Weber et al. introduced EvolveGCN [26], an advanced neural network designed to handle dynamic graph data over time, making it ideal for financial forensics in cryptocurrency. They create separate GCN models for each time step, linked by a recurrent neural network similar to a GRU, to capture temporal dependencies. By dynamically updating GCN weights using key node embedding, EvolveGCN adapts to changing transaction patterns, improving its ability to detect illicit activities as they evolve.



Fig. 1. Timeline of works analyzed

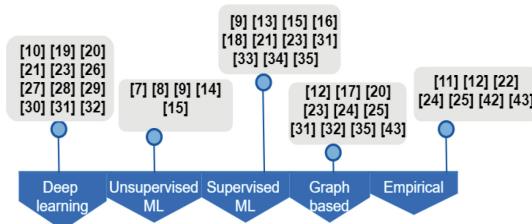


Fig. 2. Classification of approaches studied

Table 1. Classification of papers according to Machine Learning methods.

Machine Learning			
Supervised		Unsupervised	
Regression	[34]	Clustering	[7,8]
Classification	[9, 15, 16, 18, 21, 23, 31, 33–35]	Anomaly detection	[15]
		Dimension reduction	[9, 14, 15]

Table 2. Classification of papers according to Deep Learning methods.

Deep Learning				
GCN	GNN	GAT	Autoencoder	FNN
[26–30]	[20, 23, 31, 32]	[29, 30]	[10]	[21]

Table 3. Visualization of time complexity for methods used in Machine Learning and Deep Learning approaches.

Unsupervised ML	Supervised ML	Deep Learning
Time complexity of studied works		
* $O(n'kt)$: [7–9]	* $O(dn_{trees})$, $O(p)$: [9]	* $O(n^2)$: [10]; $O(n^2)$: [20]
* $O(n'kt)$, $O(pn_{sv})$: [14]	* $O(dn_{trees}), O(pn_{sv})$: [13]	* $O(V FF' + E F') + O(n)$: [19]
* $O(n^2)$, $O(lp)$: [15]	* $O(dn_{trees})$, $O(np)$: [15, 33, 34]	* $O(1)$ [21]
	* $O(d)$, $O(dn_{trees})$: [16]	* $O(V + E)$ [27]
	* $O(dn_{trees})$: [18, 21, 23, 31]	$O(n(4d_hd_x + 3d_h + d_h^2 + 1))$: [28]
	* $O(dn_{trees})$, $O(p)$: [35]	* $O(V FF' + E F')$: [29, 30]

In Table 3, n_{sv} = number of support vectors; p = number of features or columns in dataset; n_{trees} = number of trees in the ensemble algorithm, d = depth of tree, l = number of eigen vectors considered, n' = number of patterns, k = number of clusters, t = number of iterations, $|V|$ = number of nodes, F = number of input features, F' = no. of computing features, $|E|$ = number of edges, d_h = number of features in the hidden state, d_x = number of input features at each time step. Fields marked with * indicate that the complexity of the approach presented in the papers could not be determined or validated. Instead, an approximation has been made using complexity based on the classical methods used as those works' starting point.

Alarab et al. [28], describe and propose the implementation of a temporal Graph Convolutional Networks (GCN), a model consisting of combining Long short-term memory (LSTM) and Temporal Graph Convolutional Networks (TGCN). The Topology Adaptive Graph Convolutional Networks (TAGCN) method is used, which is derived from the actual GCN model. N. Pocher et al. [29] are proposing two approaches, while making use of graphs-based methods, namely GCN and Graph Attention Networks (GAN). The dataset the authors used is offered by Elliptic. The GCN approach seems to slightly outperform the

GAN efficiency. Another approach based on GAT and GCN is described in [30]. The authors propose 5 methods of implementing the approaches mentioned earlier, namely: 2 layer GCN, 4 layer GCN, 1 GAT, 1 hard voting GAT, and 1 soft voting GAT.

Weng Lo et al. [31] are using a self-supervised Graph neural Networks (GNN) framework based on a self-supervised Deep Graph Infomax (DGI) and Graph Isomorphism Network, and combining Random Forest (RF). During the experimental phase, the proposed framework proves significant performance, outperforming the other approaches tackled by the other authors. The best variant used is AF+DNE, which combines node embedding generated from DGI with all original raw features (AF). Alarab et al. [32] are proposing a method based on a Recurrent Graph neural Networks (RecGNN) methodology. RecGNN is formed by combining Long Short-Term Memory (LSTM) network with a Graph neural Networks (GNN). This method tries to achieve a higher efficiency in node classification. The temporal behavior is captured using a modified LSTM (M-LSTM), which processes the transaction data over time. This is followed by a GNN, which aggregates information from neighboring nodes to enhance the prediction capabilities. Alarab et al. [33] employed a comprehensive methodology: using a dataset provided by Elliptic, they outlined the specifications of a proposed ensemble learning method designed to work alongside supervised algorithms to classify transactions. In [34], a case study is presented on de-anonymizing illicit transactions using Bitcoin and identifying their type, using the implementation of supervised learning algorithms on the dataset provided by Chainalysis. Here, the Gradient Boosting, Random Forest, Extra Trees, and Bagging Classifier algorithms are the most efficient when it comes to classifying transactions.

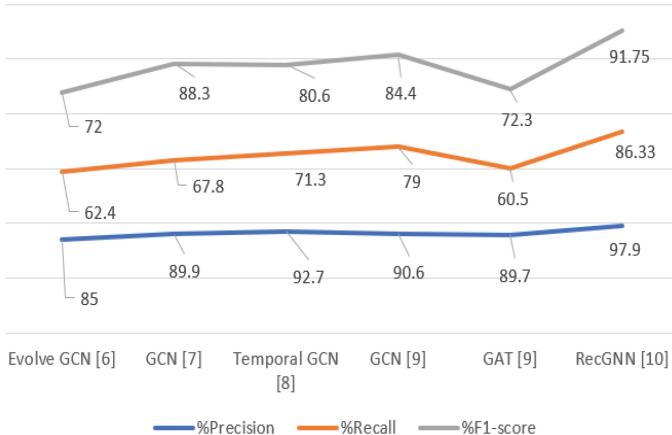


Fig. 3. Comparison of performance of methodologies analyzed

In another approach [35], authors constructed a Flow Transaction Graph, Actor Interaction Graph, Address-Transaction Graph, and User Entity Graph.

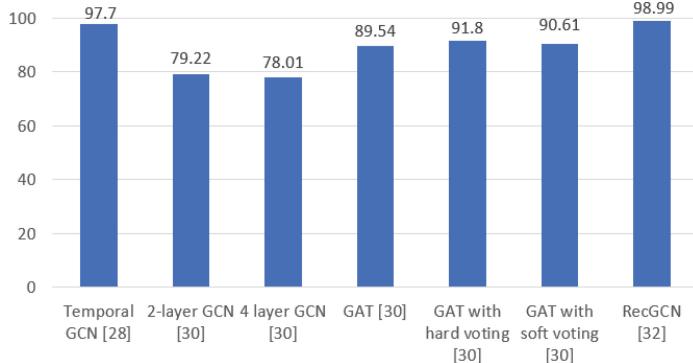


Fig. 4. Comparison of accuracy (%) of methodologies analyzed

During the experimental phase of finding illicit transaction, the authors used the following ML methods: Random Forest, Multilayer Perceptrons, Long Short-Term Memory, and Extreme Gradient Boosting. They also used the Ensemble methods by combining all but the Long Short-Term Memory method.

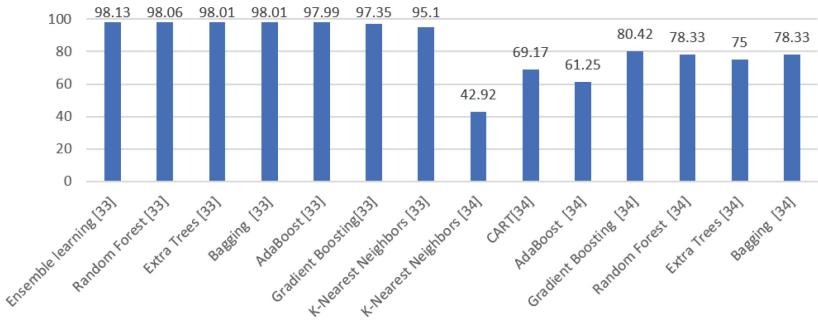


Fig. 5. Comparison of accuracy (%) for ML-based methods

3 Challenges

Detecting money laundering in blockchain systems is challenging due to technical and legislative issues. Real-time tracking is costly, and varying Bitcoin classifications affect AML law application. Regulatory actions are typically reactive, following suspicious activities. The lack of standardized cross-border legal terms and regulations causes enforcement confusion. Addressing this needs a multifaceted approach with technical, economic, and international law perspectives. Effective money laundering detection on the Bitcoin

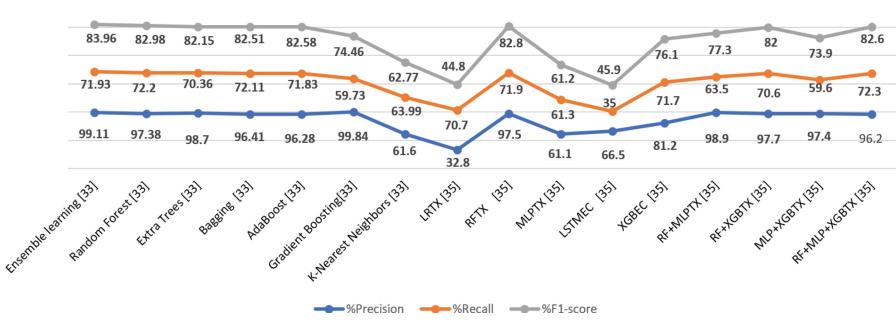


Fig. 6. Comparison of precision, recall and F1-score (%) for ML-based methods

blockchain requires standardized AML regulations, better international cooperation, privacy-transparency balance, and updated legal frameworks.

In terms of scalability, the Bitcoin blockchain's large daily transaction volume makes real-time analysis difficult, especially for historical data. Researching it requires parsing the entire chain, which is system-dependent and measures around 560 GB [36]. Parsing Bitcoin data is challenging due to its sequential nature, requiring processing from the first block to the last. This task, done via the Bitcoin core daemon service or third-party APIs, is often slow and difficult. In Fig. 7 we compare the efficiency of the functionality found within Bitcoin Core, and a C-based parser of the chain [37]. We downloaded the Bitcoin-based blockchain data onto a Samsung T7 series 4 TB external SSD and conducted further queries on blocks stored on it. For our experiment, we chose a block that contained only 774 transactions. For our experiment, we choose a block that contained only a mere 774 transactions. We ran the test on an Intel Core i7-12700H CPU, with 16 GB DDR5 memory and Windows 10 as the operating system. Bitcoin Core took 23 s to parse the entire block through its API, yet the chosen C parser performed better, achieving 10 s. Despite the C method [37] being efficient, it retrieves only 21 attributes, while the Bitcoin Core method retrieves all the attributes found within the blocks. In essence, these methods will take around 8 months, respectively 3 months to parse the entire Bitcoin blockchain, making real-time processing impossible.

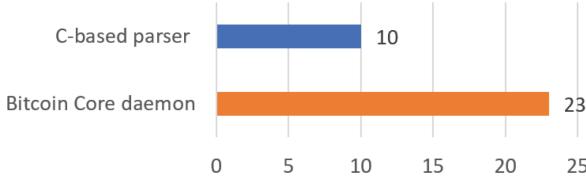


Fig. 7. Parsing results (in seconds) for a block of 774 transactions used for analysis

Traditional Machine Learning algorithms are mostly making use of the CPU, yet this approach is considered a bottleneck when it comes to the pre-processing step, discussed in several papers. In one such paper [38], authors revealed that the results of ML training on public data sets showed that data pre-processing takes up to 65% of the epoch time. Nowadays, implementing methods such as deep learning or neural networks through GPUs, become more popular, because they are specifically designed for massively parallel processing making use of thousands of threads that can handle the execution of multiple operations simultaneously. The scalability of GPUs also enables the training of large, complex models, making them essential for advancements in AI [39]. Despite traditional ML methods mostly have CPU-centric implementations, some of them can be processed on GPUs [40]. Nvidia also developed the cuML [41] library suite, to accelerate ML implementations on CUDA-enabled GPUs (Table 4).

Table 4. Machine learning algorithms and the platforms that they are known to be implemented on.

CPU	GPU
Regression algorithms	K-means
K-clustering	Random Forest
Support Vector Machines	XGBoost
Principal Component Analysis	Gradient Boost
Deep Learning	Regression algorithms
neural networks	Deep Learning
	neural networks

4 Conclusions

The Bitcoin blockchain is already fitting the criteria of processing “big data”. One needs to consider how to reduce the time and space complexities: immediate future trends, of which some are in progress even in the authors’ backlog, involve parallelization of the Bitcoin processing mechanism(s), as well as of the detection methods using heterogeneous architectures. In this paper we discussed recent works in the money-laundering detection space, as well as propose a timeline for them. We produced a thorough review study of the recent approaches, discussing feasibility and challenging aspects of the detection of illicit transactions all throughout. We discussed detection taxonomy and outlined challenges identified. While we are aware this is a rather small but important step in this niche, we hope it brings further clarity for researchers, and we express confidence that future technical, legal and economical changes and the overcoming of the challenges identified will be greatly contributing to the successful outcome for future fraud detection.

Acknowledgements. This work has been partially supported by virtuaLedger [45] and RoNaQCI, part of EuroQCI, DIGITAL-2021-QCI-01-DEPLOY-NATIONAL, 101091562.

References

1. Onion Identity Services. <http://endtovmbc5vokdpnxrhajcwgkfbkfz4wbyhbj6ueisai4prtvencheyd.onion/info.php>. Accessed 28 May 2024
2. Rent a Hacker. <http://jn6weomv6klvnwdwcu55miabpwklsmmya5qrkt4miif4shrqmvdhqd.onion/>. Accessed 28 May 2024
3. Bitcoin Fog. <https://bitcoinfog.info>. Accessed 28 May 2024
4. What is Anti Money Laundering (AML)? <https://internationalsales.lexisnexis.com/glossary/compliance/anti-money-laundering-aml>. Accessed 27 May 2024
5. What is know-your-customer (KYC) in crypto, and why do exchanges require it? <https://notabene.id/crypto-travel-rule-101/kyc-crypto>. Accessed 27 May 2024
6. Know Your Transaction (KYT). <https://ciphertrace.com/glossary/know-your-transaction-kyt/>. Accessed 27 May 2024
7. Deepak, Z., Shah, A.: Analysis of Bitcoin Network Dataset for Fraud. Unpublished Report 27 (2013)
8. Monamo, P., Marivate, V., Twala, B.: Unsupervised learning for robust Bitcoin fraud detection. In: Information Security for South Africa (ISSA), Johannesburg, South Africa, pp. 129–134 (2016). <https://doi.org/10.1109/ISSA.2016.7802939>
9. Monamo, P.M., Marivate, V., Twala, B.: A multifaceted approach to bitcoin fraud detection: global and local outliers. In: 15th IEEE International Conference on Machine Learning and Applications (ICMLA), Anaheim, CA, USA, pp. 188–194 (2016). <https://doi.org/10.1109/ICMLA.2016.0039>
10. Nan, L., Tao, D.: Bitcoin mixing detection using deep autoencoder. In: Proceedings of the 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), pp. 280–287 (2018). <https://doi.org/10.1109/DSC.2018.00047>
11. Wegberg, R., Oerlemans, J.-J., van Deventer, O.: Bitcoin money laundering: mixed results? An explorative study on money laundering of cybercrime proceeds using bitcoin. *J. Financ. Crime* **25** (2018). <https://doi.org/10.1108/JFC-11-2016-0067>
12. Prado-Romero, M.A., Doerr, C., Gago-Alonso, A.: Discovering bitcoin mixing using anomaly detection. In: Mendoza, M., Velastín, S. (eds.) CIARP 2017. LNCS, vol. 10657, pp. 534–541. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-75193-1_64
13. Jobse, F.: Detecting suspicious behavior in the bitcoin network. M.S. thesis, Tilburg University (2017)
14. Pham, T., Lee, S.: Anomaly detection in bitcoin network using unsupervised learning methods. [arXiv:1611.03941](https://arxiv.org/abs/1611.03941) (2017)
15. Lorenz, J., Silva, M.I., Aparício, D., Ascensão, J.T., Bizarro, P.: Machine learning methods to detect money laundering in the bitcoin blockchain in the presence of label scarcity. [arXiv:2005.14635](https://arxiv.org/abs/2005.14635) (2021)
16. Nerurkar, P., Busnel, Y., Ludinard, R., Shah, K., Bhirud, S., Patel, D.: Detecting illicit entities in bitcoin using supervised learning of ensemble decision trees. In: Proceedings of the 10th International Conference on Information and Communication Management (ICICM), Paris, France, pp. 25–30 (2020). <https://doi.org/10.1145/3418981.3418984>

17. Tharani, J.S., Charles, E.Y.A., Hóu, Z., Palaniswami, M., Muthukkumarasamy, V.: Graph based visualisation techniques for analysis of blockchain transactions. In: 2021 IEEE 46th Conference on Local Computer Networks (LCN), Edmonton, AB, Canada, pp. 427–430 (2021). <https://doi.org/10.1109/LCN52139.2021.9524878>
18. Ashfaq, T., et al.: A machine learning and blockchain based efficient fraud detection mechanism. *Sensors* **22**(19), 7162 (2022). <https://doi.org/10.3390/s22197162>
19. Zheng, Y.: GRU-GAT model for blockchain bitcoin abnormal transaction detection. In: 2022 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS), Dalian, China, pp. 666–674 (2022). <https://doi.org/10.1109/TOCS56154.2022.10016137>
20. Nicholls, J., Kuppa, A., Le-Khac, N.-A.: FraudLens: graph structural learning for bitcoin illicit activity identification. In: Proceedings of the 39th Annual Computer Security Applications Conference, pp. 324–336. Association for Computing Machinery, New York (2023). <https://doi.org/10.1145/3627106.3627200>
21. Ghosh, C., Chowdhury, A., Das, N., Sadhukhan, B.: Enhancing financial fraud detection in bitcoin networks using ensemble deep learning. In: 2023 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS), New Raipur, India, pp. 1–6 (2023). <https://doi.org/10.1109/ICBDS58040.2023.10346590>
22. Trivedi, G.R., Bolla, J.V., Sireesha, M.: A Bitcoin transaction network using cache based pattern matching rules. In: 2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, pp. 676–680 (2023). <https://doi.org/10.1109/ICSSIT55814.2023.10061064>
23. Ouyang, S., Bai, Q., Feng, H., Hu, B.: Bitcoin money laundering detection via subgraph contrastive learning. *Entropy* **26**(3) (2024). <https://doi.org/10.3390/e26030211>. Art. no. 211
24. Meiklejohn, S., et al.: A fistful of bitcoins: characterizing payments among men with no names. In: Proceedings of the 2013 Conference on Internet Measurement Conference, pp. 127–140 (2013). <https://doi.org/10.1145/2504730.2504747>
25. Möser, M., Böhme, R., Breuker, D.: An inquiry into money laundering tools in the bitcoin ecosystem. In: 2013 APWG eCrime Researchers Summit, San Francisco, CA, USA, pp. 1–14 (2013). <https://doi.org/10.1109/eCRS.2013.6805780>
26. Weber, M., et al.: Anti-money laundering in bitcoin: experimenting with graph convolutional networks for financial forensics. *arXiv:1908.02591* (2019)
27. Alarab, I., Prakoonwit, S., Nacer, M.I.: Competence of graph convolutional networks for anti-money laundering in bitcoin blockchain. In: Proceedings of the International Conference on Machine Learning Technologies, pp. 23–27 (2020). <https://doi.org/10.1145/3409073.3409080>
28. Alarab, I., Prakoonwit, S.: Graph-based LSTM for anti-money laundering: experimenting temporal graph convolutional network with bitcoin data. *Neural Process. Lett.* **55**, 1–19 (2022). <https://doi.org/10.1007/s11063-022-10904-8>
29. Pocher, N., Zichichi, M., Merizzi, F., Shafiq, M.Z., Ferretti, S.: Detecting anomalous cryptocurrency transactions: an AML/CFT application of machine learning-based forensics. *Electron. Markets* **33**(1), 37 (2023). <https://doi.org/10.1007/s12525-023-00654-3>
30. Pan, Y.: Enhancing predictive models for illicit activities in the Bitcoin transaction network using advanced graph analytical techniques. *Appl. Comput. Eng.* **48**, 78–86 (2024). <https://doi.org/10.54254/2755-2721/48/20241181>
31. Lo, W.W., Kulatilleke, G.K., Sarhan, M., Layeghy, S., Portmann, M.: Inspection-L: self-supervised GNN node embeddings for money laundering detection in bitcoin. *Appl. Intell.* **53**(16), 19406–19417 (2023)

32. Alarab, I., Prakoonwit, S.: Robust recurrent graph convolutional network approach based sequential prediction of illicit transactions in cryptocurrencies. *Multimedia Tools Appl.* **83**, 1–16 (2023). <https://doi.org/10.1007/s11042-023-17323-4>
33. Alarab, I., Prakoonwit, S., Nacer, M.I.: Comparative analysis using supervised learning methods for anti-money laundering in bitcoin. In: Proceedings of the International Conference on Machine Learning Technologies, pp. 11–17 (2020)
34. Yin, H., Langenheldt, K., Harlev, M., Mukkamala, R.R., Vatrapu, R.: Regulating cryptocurrencies: a supervised machine learning approach to de-anonymizing the bitcoin blockchain. *J. Manage. Inf. Syst.* **36**, 37–73 (2019). <https://doi.org/10.1080/07421222.2018.1550550>
35. Elmougy, Y., Liu, L.: Demystifying fraudulent transactions and illicit nodes in the bitcoin network for financial forensics. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023 (2023)
36. Bitcoin Blockchain size. https://ycharts.com/indicators/bitcoin_blockchain_size. Accessed 6 June 2024
37. Blockchain-parser. <https://github.com/rmull/blockchain-parser>. Accessed 2 June 2024
38. Mohan, J., Phanishayee, A., Raniwala, A., Chidambaram, V.: Analyzing and Mitigating Data Stalls in DNN Training. CoRR, vol. abs/2007.06775 (2020)
39. Why GPUs are Great for AI. <https://blogs.nvidia.com/blog/why-gpus-are-great-for-ai/>. Accessed 30 May 2024
40. NVIDIA Glossary. <https://www.nvidia.com/en-us/glossary/>. Accessed 30 May 2024
41. RAPIDS cuML: A Library For GPU Accelerated Machine-Learning. <https://developer.download.nvidia.com/video/gputechconf/gtc/2019/presentation/s9817-rapids-cuml-a-library-for-gpu-accelerated-machine-learning.pdf>. Accessed 30 May 2024
42. Liu, M., Chen, H., Yan, J.: Detecting roles of money laundering in bitcoin mixing transactions: a goal modeling and mining framework. *Front. Phys.* **9** (2021)
43. Hu, Y., Seneviratne, S., Thilakarathna, K., Fukuda, K., Seneviratne, A.: Characterizing and Detecting Money Laundering Activities on the Bitcoin Network (2019)
44. West, J., Bhattacharya, M., Islam, R.: Intelligent financial fraud detection practices: an investigation. In: Tian, J., Jing, J., Srivatsa, M. (eds.) *SecureComm 2014, Part II*. LNICST, vol. 153, pp. 186–203. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23802-9_16
45. virtuaLedger. <https://virtualedger.com>. Accessed 18 July 2024



An Approach Based on LLMs for Forensic Threat Detection in Autonomous Systems

David Sobrín-Hidalgo^(✉), Irene González-Fernández,
Miguel Ángel González-Santamaría, Adrián Campazas-Vega,
Claudia Álvarez-Aparicio, Ángel Manuel Guerrero-Higuera,
Francisco Javier Rodríguez-Lera, and Vicente Matellán-Olivera

Robotics Group, University of León, Campus de Vegazana, s/n, 24071 León, Spain
`{dsobh, igonf, mgons, acamv, calvaa,
am.guerrero, fjrrol, vicente.matellan}@unileon.es`

Abstract. In a world where interaction between humans and robotic systems is increasingly common, cybersecurity is essential to protect the integrity of both. Security measures to protect humans from possible accidents or cyber-attacks on a robotic system are essential. However, after an unexpected event, whether caused by a system malfunction or the influence of a malicious actor, it must be possible to explain what happened. In this paper, we propose to combine the fields of explainability and cybersecurity by exploring the potential of an explainability system in robotics based on LLMs as a forensic tool. We performed a denial of service attack on the mobile robot's controller. After, we analyzed the logs generated by the robot using the proposed explainability system. The results obtained show that this system can detect the presence of a potential threat affecting the robot's behavior.

Keywords: Cybersecurity · Explainability · Forensics · LLM · Robotics · ROS 2

1 Introduction

The integration of robots into the real world is becoming increasingly common, from the use of industrial robots and autonomous cars to the use of remotely operated surgical robots. These systems interact directly with humans; therefore cyber-attacks against these robots can have serious consequences. An attack on the behavior of one of these robots can lead to material loss and, in the worst case, human loss.

For this reason, all robots that interact with humans in any way (human-robot interaction or HRI) must incorporate robust security measures to protect users from any incident. However, it is not always possible to avoid such situations and it is necessary to analyze what happened after a cyber incident. This

knowledge is essential for assigning responsibility, identifying errors, and closing potential security gaps. In addition, all this information helps to strengthen systems against new attacks, thus preventing a cyber incident from happening again.

Explainability can be understood as the ability of an autonomous or intelligent system to explain its own decisions. This field of study strongly impacts the world of artificial intelligence, known as eXplainable Artificial Intelligence (XAI) [1]. On the other hand, the presence of AI within the world of cybersecurity has favored the development of the concept of explainability within this field [2].

However, explainability does not only affect learning models but also any system that works autonomously, such as social or service robots. The variant of explainability applied to the robotic world is known as *eXplainable Autonomous Robots* (XAR) [3], and it is from this perspective that the present work will be approached.

There has been a growing interest in explainability in official regulations in recent years, suggesting a latent need for any autonomous system to incorporate explainability tools. But can such a system be used as an additional safety tool?

A robust accountability system should collect all relevant data produced by the robot. This data could be analyzed to determine the origin of behavior and to identify the reason for a system problem. A system that fulfills these characteristics could also serve as a forensic analysis tool.

LLMs have transformed all areas of computing, including cyber security. The use of LLMs presents two perspectives, as they can be used maliciously as tools to develop cyber-attacks or as defensive tools. The work [4] examines the history and background of some of the most popular LLMs, as well as their relationship to cyber defense, and explores their possible malicious use. The authors of [5] identify several types of attacks, including social engineering, phishing, and ransomware generation, that could benefit by leveraging language models.

The high text processing and text generation capabilities of LLMs make them an efficient tool for such attacks. On the other hand, [6] emphasizes this ability to perform cyber defense tasks, such as analyzing source code for vulnerabilities or generating reports. Similarly, in [5] the authors state:

“ChatGPT can process large volumes of log data and efficiently detect anomalies or security issues within access logs.”

ROS (Robot Operating System) has become the most popular middleware for building robotic applications. The successor of ROS, called ROS 2, has been used in this research [7]. In terms of security, ROS 2 has improved the communication security over its predecessor by including cryptographic functions. However, these are disabled by default, and enabling them can affect system performance [8].

Despite the improvements introduced in ROS 2, the middleware is still not free of vulnerabilities. In [9], the authors expose vulnerabilities in the communication protocol through a man-in-the-middle (MITM) attack. Moreover, in [10],

authors demonstrate the possibility of capturing messages exchanged by nodes within a network using sniffing network tools such as Wireshark. Another of the most common attacks in robotics is the attempt to cause a Denial of Service (DoS). An analysis of this type of attack on remotely controlled robots can be found in [11].

The main objective of this work is to test the potential of the explainability system proposed in [12] to detect anomalous behavior in a mobile robot built over ROS 2. The explainability system is built on a ROS implementation that uses Large Language Models (LLMs) to analyze the robot logs. In this way, the LLM is responsible for interpreting these logs and explaining what happened after the execution.

The article is structured as follows: in Sect. 2 the materials used are presented and the experiment carried out is defined. The results obtained from the experiment are collected and discussed in Sect. 3. Finally, Sect. 5 presents the conclusions and future work.

2 Materials and Methods

This article proposes the use of an explainability system integrated into a robotic system to facilitate and accelerate forensic analysis tasks after an incident has occurred. To address this problem, an LLM-based explainability system [12], has been used as a starting point. This system leverages natural language processing technology to analyze large sets of logs generated during the robot's execution.

2.1 Materials

A series of resources and materials, both software and hardware, have been used to carry out the experiment that guides this research. These materials are listed below.

- **ROS 2.** Both the robot and the software components developed for the experiment are built on ROS, specifically on ROS 2 Humble, the latest stable ROS 2 distribution.
- **Langchain [13].** A library that facilitates the development of applications based on LLMs. It also offers the possibility of implementing the RAG method simply.
- **Liminerity/M7-7b-GGUF.** Model in charge of processing the user's question and the most relevant logs to generate an explanation.
- **Mixedbread-ai/mxbai-embed-large-v1.** Model responsible for generating the embedding of each of the logs received through the topic `/rosout`.
- **TIAGo.** A mobile manipulator robot manufactured by PAL Robotics¹. During the experiment, it is teleoperated to navigate within Leon@Home Testbed.
- **Leon@Home Testbed².** The environment in which the experiment has been carried out is certified by the ERL (European Robotics League).

¹ <https://pal-robotics.com/es/>.

² <https://robotica.unileon.es/index.php?title=Testbed>.

The basic operation of ROS 2 is that of a Publisher/Subscribe paradigm, in which communication is carried out through software components called nodes that exchange messages through information channels called topics. The advantage of this type of communication is the ease of monitoring the messages exchanged. All the information that reaches the topics, can be recorded.

The explainability system uses this message log and tries to reason what has happened. It works by using the messages that have been sent to the main ROS 2 log (`/rosout`) as context for the LLM in charge of generating the explanations. However, the number of messages published in `/rosout` may be a context too large for the system, and cause reasoning to be inconsistent and inaccurate. For this reason, a smaller context is created from all the recorded messages. This context is built on the most relevant logs to build a concrete explanation. This new context is achieved through the Retrieval Augmented Generator (RAG) method and using a vector database to store the logs. Figure 1 illustrates the operation of the explainability system using the RAG method.

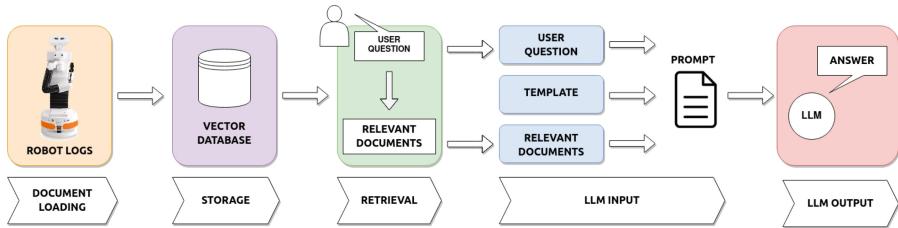


Fig. 1. Operation of the explainability system where the RAG method is shown. Image extracted from [12].

The models used have been obtained through Hugging Face³. For selection, the position of their respective leaderboards in Hugging Face was taken into consideration; [14] for the explainability model and [15] for the model in charge of the embeddings. Other influential characteristics in the choice have been the size of the model and its quantization.

2.2 Methods

The experiment proceeds as follows. A user teleoperates a service robot (TIAGo) via a remote controller to move it in the environment (Leon@Home Testbed). Controller publishes messages on the topic `/input_joy/cmd_vel`, used to exchange messages about the robot's motion in `/geometry/geometry_msgs/Twist`⁴ format.

³ <https://huggingface.co/>.

⁴ http://docs.ros.org/en/noetic/api/geometry_msgs/html/Twist.html.

A ROS 2 node has been developed to monitor the activity of topic `/input_joy/cmd_vel`. The node calculates the Hz at which it is publishing and logs a trace, in `/rosout`, in case the publishing rate increases or decreases significantly.

The explainability system subscribes to `/rosout` topic, where the monitor node publishes relevant changes of the publishing rate of `/input_joy/cmd_vel`. The system will take as context the logs generated during teleoperation. Upon finishing the robot's activity, the user can ask the system to know what has happened during execution.

The evaluation has been carried out on two executions (*E1* and *E2*). In each execution, the teleoperator's goal is to move the robot from one point to another within the environment, where these points are always the same. The route followed by the robot is shown in Fig. 2.

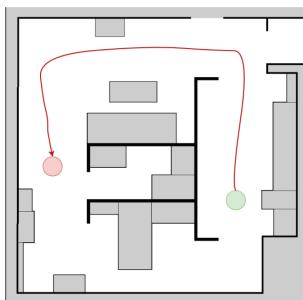


Fig. 2. Plan of Leon@Home Testbed showing the route followed by the robot with a red arrow. The green circle represents the starting point of the navigation, while the red circle represents the destination point.

E1 The robot completes the navigation without any attack.

E2 The robot suffers an attack with a publication rate much higher than normal (500 Hz).

During *E2*, a Denial of Service (DoS) attack was performed by publishing additional messages on the topic that controls the robot's movement. The way ROS 2 communication works, it is easy to flood a topic with malicious messages that disrupt the operation of the robot.

The robot controller has a publish rate of 10 Hz, so it is possible to flood the topic by publishing at a much higher rate, for example, 500 Hz. The information published during the attack are messages of the type `/geometry/geometry_msgs/Twist` with 0 values in all components (linear and angular velocity). This will cause the controller messages to be lost and the robot to stop.

The data generated during tests were recorded using Rosbags, a ROS data format that allows storage messages exchanged during robot execution.

The Rosbags obtained were analyzed using the explainability system designed for ROS 2 in [12]. The data of the topic `/rosout` stored in the Rosbag are processed by the LLM. The explanation system used acts as a dialogue manager between the user and the LLM. Thus, to get an explanation of what happened, the user has to ask the model a question. The questions asked are part of the LLM prompt and their expression and structure are fundamental to obtaining coherent results. The explainability system should therefore answer the same question for both executions.

Question Do you observe any problems in the cmd_vel? If so, could it be a cyberattack, and what type of cyberattack?

In addition to the question, it is necessary to define a template prompt, to give certain information and context to the model. The template used is shown in the Listing 1.1.

```
"You are an explainability AI tool for ROS 2 mobile robots. "
"You goal is to detect possible cyberattacks. "
"You have to detect anomalous behavior within the robot's data. "
"You have to interpret the robot's data paying attention to the publication
    rates of cmd_vel, that is between 8.0Hz and 23.0Hz.\n\n"

"Relevant data:\n"
"{logs}\n\n"

"USER: "
"Given the context information and not prior knowledge, answer the query: {"
    "question}\n"
"ASSISTANT:"
```

Listing 1.1. Prompt template used for the LLM

The final prompt that will serve as input to the LLM is composed of the template, the user's question, and the 10 most relevant logs obtained through the RAG method. The logs collect the publishing rate of the `/cmd_vel` topic. The template states how the system should behave, like an Intrusion Detection System (IDS), and specify the range of publish rate during the normal execution of the robot. Due to the impact of the context and verbosity of the data on the behaviour of the model. This is necessary for the LLM to produce a reasonable answer.

3 Results

This section provides the experimental results. It displays the output of the monitoring node for each execution, along with the question-answer pairs gen-

Table 1. Rosbag information for the different executions. Includes the total number of messages logged, the duration of the execution, and the size of the rosbag.

Execution ID	# Messages	Duration (s)	Size (MB)
E1	7120	75.6	6.6
E2	10355	40.2	3.5

erated by the explanation system. In addition, Table 1 shows information on the generated rosbags.

3.1 Execution 1 (*E1*)

In *E1*, the robot navigates without incident. Traces of the monitoring node are shown in the Listing 1.2. The user’s question and the LLM’s answer are collected via a ROS 2 service, see Listing 1.3. The video⁵ shows the execution of this experiment.

```
Average publication rate of cmd_vel: 9.00Hz.
Average publication rate of cmd_vel: 9.14Hz.
Average publication rate of cmd_vel: 9.50Hz.
Average publication rate of cmd_vel: 12.00Hz.
Average publication rate of cmd_vel: 12.86Hz.
Average publication rate of cmd_vel: 14.22Hz.
Average publication rate of cmd_vel: 19.56Hz.
Average publication rate of cmd_vel: 19.54Hz.
Average publication rate of cmd_vel: 20.67Hz.
Average publication rate of cmd_vel: 20.93Hz.
Average publication rate of cmd_vel: 21.73Hz.
Average publication rate of cmd_vel: 21.73Hz.
Average publication rate of cmd_vel: 22.39Hz.
Average publication rate of cmd_vel: 21.47Hz.
Average publication rate of cmd_vel: 20.62Hz.
Average publication rate of cmd_vel: 20.01Hz.
Average publication rate of cmd_vel: 20.01Hz.
Average publication rate of cmd_vel: 20.01Hz.
```

Listing 1.2. Frequency of publication during Execution 1

```
$ ros2 service call /question explicability_msgs/srv/Question "{question: 'Do
    you observe any problem in the cmd_vel? If so, could it be a cyberattack
    , and what type of cyberattack?'}"
response:
explicability_msgs.srv.Question_Response(answer=" Based on the provided data
    alone, focusing solely on the publication rates of cmd_vel within the
    expected range of 8.0Hz to 23.0Hz, there do not appear to be any obvious
    anomalies or signs of a cyberattack. The rates fluctuate naturally, but
    they remain within the acceptable range. However, it's important to note
    that a comprehensive analysis would require more context and a broader
    set of metrics to accurately detect potential cybersecurity threats.")
```

Listing 1.3. Explanation generated by the LLM

3.2 Execution 2 (*E2*)

In *E2*, a DoS attack is performed by flooding the /cmd_vel topic while the robot is remotely controlled. This attack causes the robot to stop, ignoring instructions from its human controller. Traces generated by the monitor node are shown in Listing 1.4. The user’s question and the LLM’s answer are collected in the Listing 1.5. The video⁶ shows the execution of this experiment.

⁵ https://www.youtube.com/watch?v=fov5Jwoh_vQ.

⁶ <https://www.youtube.com/watch?v=LSVOb0mBxPE>.

```
Average publication rate of cmd_vel: 8.50Hz.
Average publication rate of cmd_vel: 8.57Hz.
Average publication rate of cmd_vel: 10.25Hz.
Average publication rate of cmd_vel: 11.41Hz.
Average publication rate of cmd_vel: 12.23Hz.
Average publication rate of cmd_vel: 36.32Hz.
Average publication rate of cmd_vel: 96.57Hz.
Average publication rate of cmd_vel: 146.37Hz.
Average publication rate of cmd_vel: 170.00Hz.
Average publication rate of cmd_vel: 170.00Hz.
```

Listing 1.4. Frecuencia de publicación durante la Ejecución 1

```
$ ros2 service call /question explicability_msgs/srv/Question "{question: 'Do
you observe any problem in the cmd_vel? If so, could it be a cyberattack?
, and what type of cyberattack?'}

response:
explicability_msgs.srv.Question_Response(answer=" In the provided data, there
are significant spikes in the average publication rates of cmd_vel,
which exceed the expected range of 8.0Hz to 23.0Hz. At
1710853730.8051765, the rate jumps to 36.32Hz, and from 1710853735.805152
to 1710853745.805236, the rates are abnormally high, reaching up to
170.00Hz. While these extreme fluctuations might indicate a problem with
the system, it's crucial to note that solely based on this data, we
cannot definitively conclude it as a cyberattack. Anomalous behavior
could be due to various reasons, including software glitches or
misconfigurations. To determine if it's a cyberattack, more context and
evidence would be necessary. If confirmed as a malicious act, one might
speculate a Denial of Service (DoS) attack or a manipulation attempt to
compromise the robot's control, but further investigation is required for
a conclusive assessment.")
```

Listing 1.5. Explicación generada por el LLM

4 Discussion

The main objective of this work is to analyze the responses generated by the explainability system to assess whether its use can be beneficial for forensic analysis. If we analyze the first case *E1*, shown in the Listings 1.2 and 1.3, we can observe how the model considers logs within the normal behavior.

In the second execution, *E2*, a DoS by flooding the topic that controls the robot's movement has been done. Results obtained for this execution are shown in Listings 1.4 and 1.5. In that case, the system correctly identifies the increments of the publish rate of the topic. The system indicates that the publication frequency becomes “abnormally high, reaching up to 170.00 Hz”. It also associates the logs generated with a possible attack. The system successfully deduces that a DoS attack to compromise the control of the robot.

In both executions, the system remarks the need for the case to be studied by a human expert, making it clear that its response is not conclusive. It is necessary to highlight this behavior because we propose to use this type of technology as a complement to the analyst, but never as a replacement. The final goal is to speed up the forensic analysis process. Related to Table 1, we can see that for executions lasting around one minute, a large number of logs are generated. *E1* generated 7120 messages and *E2* generated a total of 10355 messages. The

system can warn of a possible threat and guide the analyst through the data to start the research.

5 Conclusions

We present a mechanism to use an explainability system based on LLMs as a complementary tool to perform forensic analysis tasks on robotic systems. Results demonstrate the capacity of this system to infer events occurring during a cyber-attack. This contributes to enhanced comprehension of the attack. Furthermore, it reduces the time spent on the forensic analysis.

As future work, the system's validity will be assessed with other attacks commonly carried out on robots, such as a MitM attack[16]. We plan to improve the quality of our experimentation by defining metrics to evaluate the system's performance. Additionally, to enrich the system evaluation, we will select a forensic tool as a baseline and perform a comparison, using the defined metrics, between our system and the selected tool. Furthermore, the system will be modified to extend its functionality creating a real-time incident alert system. In this way, the system will not only focus on generating explanations but will also have IDS functionalities.

Acknowledgments. This research is part of the project TESCAC, financed by “European Union NextGeneration-EU, the Recovery Plan, Transformation and Resilience, through INCIBE”. Moreover, it is part of the Grant PID2021-126592OB-C21 funded by MCIN/AEI/10.13039/501100011033 and by “ERDF A way of making Europe”. Finally, David Sobrín-Hidalgo thanks the University of León for funding his doctoral studies.

References

1. Došilović, F.K., Brčić, M., Hlupić, N.: Explainable artificial intelligence: a survey. In: 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) 2018, pp. 0210–0215 (2018)
2. Capuano, N., Fenza, G., Loia, V., Stanzione, C.: Explainable artificial intelligence in cybersecurity: a survey. *IEEE Access* **10**, 93 575–93 600 (2022)
3. Sakai, T., Nagai, T.: Explainable autonomous robots: a survey and perspective. *Adv. Robot.* **36**(5–6), 219–238 (2022). <https://doi.org/10.1080/01691864.2022.2029720>
4. Kucharavy, A., et al.: Fundamentals of generative large language models and perspectives in cyber-defense (2023)
5. Gupta, M., Akiri, C., Aryal, K., Parker, E., Praharaj, L.: From ChatGPT to ThreatGPT: impact of generative AI in cybersecurity and privacy. *IEEE Access* **11**, 80 218–80 245 (2023)
6. Motlagh, F.N., Hajizadeh, M., Majd, M., Najafi, P., Cheng, F., Meinel, C.: Large language models in cybersecurity: state-of-the-art. arXiv preprint [arXiv:2402.00891](https://arxiv.org/abs/2402.00891) (2024)
7. Macenski, S., Foote, T., Gerkey, B., Lalancette, C., Woodall, W.: Robot operating system 2: design, architecture, and uses in the wild. *Sci. Robot.* **7**(66), eabm6074 (2022). <https://doi.org/10.1126/scirobotics.abm6074>

8. Kim, J., Smereka, J.M., Cheung, C., Nepal, S., Grobler, M.: Security and performance considerations in ROS 2: a balancing act. arXiv preprint [arXiv:1809.09566](https://arxiv.org/abs/1809.09566) (2018)
9. Teixeira, R.R., Maurell, I.P., Drews, P.L.: Security on ROS: analyzing and exploiting vulnerabilities of ROS-based systems. In: 2020 Latin American Robotics Symposium (LARS), 2020 Brazilian Symposium on Robotics (SBR) and Workshop on Robotics in Education (WRE) 2020, pp. 1–6 (2020)
10. McClean, J., Stull, C., Farrar, C., Mascarenas, D.: A preliminary cyber-physical security assessment of the robot operating system (ROS). In: Unmanned Systems Technology XV, vol. 8741, pp. 341–348. SPIE (2013)
11. Bonaci, T., Yan, J., Herron, J., Kohno, T., Chizeck, H.J.: Experimental analysis of denial-of-service attacks on teleoperated robotic systems. In: Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems, pp. 11–20 (2015)
12. Sobrín-Hidalgo, D., González-Santamaría, M.A., Guerrero-Higueras, Á.M., Rodríguez-Lera, F.J., Matellán-Olivera, V.: Explaining autonomy: enhancing human-robot interaction through explanation generation with large language models (2024)
13. Chase, H.: LangChain (2022). <https://github.com/langchain-ai/langchain>
14. Beeching, E., et al.: Open LLM leaderboard (2023). https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard
15. Muennighoff, N., Tazi, N., Magne, L., Reimers, N.: MTEB: massive text embedding benchmark. arXiv preprint [arXiv:2210.07316](https://arxiv.org/abs/2210.07316) (2022). <https://arxiv.org/abs/2210.07316>
16. Díez, A.M., Vega, A.C., de Castro, B.J., Aparicio, C.Á., Martínez, I.S.C., Higueras, Á.M.G.: Evaluación de la seguridad en el robot cuadrúpedo a1 de unitree robotics. In: Actas de las VIII Jornadas Nacionales de Investigación en Ciberseguridad: Vigo, 21 a 23 de junio de. Universidade de Vigo 2023, pp. 133–138 (2023)



Smart Water Meter Network Anomaly and Failure Detection System

Łukasz Saganowski and Tomasz Andrysiak

Institute of Telecommunications and Computer Science, Bydgoszcz University of Science and Technology, Al. Prof. S. Kaliskiego 7, 85-796 Bydgoszcz, Poland
{lulasz.saganowski,tomasz.andrysiak}@pbs.edu.pl

Abstract. In the article we present a solution for analyzing water consumption from a multi-family buildings. In the first stage, data from the tested water meters are collected, aggregated and prepared for initial processing. In the next step, we check whether there are any gaps in the time series and, if so, we fill them using interpolation. After this time, the time series is processed by machine learning algorithms in real time. Traffic features from the smart meter network are collected within one minute and processed in sliding time windows. Anomaly or failure conditions take into account the number of times the values of the current signal extracted on the line exceed the prediction interval per unit time. Finally, a report is submitted to the maintenance services about possible problems with a selected part of the water infrastructure. The obtained experimental results confirmed the effectiveness of the presented method.

Keywords: Intelligent water meters · time series analysis · prediction process · computational intelligence methods · anomaly/failure detection

1 Introduction

A significant challenge for suppliers of water intended for human consumption is to reduce water losses in the distribution network and internal water supply installations of recipients. For this purpose, there is a need for proper diagnosis based on authenticity and accurate data. Then the solutions should be based on the idea of building intelligent water supply systems, where diagnostic knowledge comes from reliable monitoring and proper data analysis using data mining [1]. Insertion of Intelligent Water Network would allow to improve the quality of undertaken decisions about designing and operating water supply systems. On the other hand, challenges connected with applying such systems concern dense and automatic meter reading (Automatic Meter Reading), along with machine learning methods to analyze, predict and detect anomalies in recorded time series in real time [2, 3].

Effective prediction and detection of anomalies in one-dimensional time series using classic machine learning algorithms like ARIMA (Autoregressive Integrated Moving Average) or Linear regression models [4] is usually not a simple and easy task [5]. This is due to the fact that the search for sophisticated learning algorithms is the key to success

in achieving good prediction and anomaly detection results for one-dimensional time series. In this context, machine learning and deep learning algorithms can achieve better results in applications where the signal under study has a complex, sophisticated and time-varying structure, strong noise, or missing observations (as in the case of water supply measurement networks) [6, 7].

The main goal of the article was to develop algorithms for forecasting water demand, which constitute the basis for detecting anomalies. We propose anomaly and failure detection system with the use of artificial intelligence algorithms that helps maintenance staff of intelligent water system to find anomalies and failures in water supplying system. In the first stage, a campaign was carried out to record water consumption in selected facilities and the results of water flow rate measurements that were obtained from the infrastructure operator. The next stage was to develop a tool based on a learning machine, pre-process the collected data and test artificial intelligence models to select the most appropriate one. The dynamics of the developed system is then expressed through quick detection of anomalies in almost real time. The basis is the solution implemented at the stage of forecasting water demand, and an increase in error between the predicted and observed values indicates the occurrence of an anomaly, i.e. a failure or incorrect consumption.

The article has the following structure: after the introduction, Sect. 2 presents in detail: forecasting time series of water flow consumption and the method of prediction and anomaly detection, as well as the methodology for calculating artificial intelligence models. Section 3 provides implementation details and experimental results. Then the conclusions were presented.

2 Methodology for Anomaly and Failure Detection in Smart Water Meters Network

Test bed metering network as well as smart meter was designed by Authors during an industrial research project. Data from test metering network were gathered in the period one year. In the article we analyzed water consumption from a multi-family building. In Fig. 1 we presented main steps of proposed methodology for anomaly and failure system applied to network of smart water meters.

2.1 The Proposed Solution

In the first step data from smart water meters network are collected, aggregated and prepared for the next preprocessing step. Subsequently water flow [liter/min] traffic feature is extracted. Values of traffic feature are collected in a form of univariate time series (univariate time series are defined as a time series where values are sampled in constant time periods). In the next step we check if there are any gaps in time series and these gaps are filled using interpolation methodology. After that time series are processed by machine learning algorithm.

Time series are analyzed by machine learning algorithms in real time. Traffic feature from smart meter networks is gathered in one minute time period and are processed in sliding time windows. Time required for fulfilling samples for the next sliding analysis

window is sufficient to perform any required calculations that are necessary in our methodology in real time.

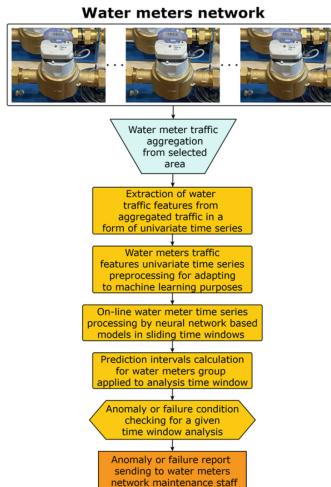


Fig. 1. The main steps of anomaly and failure methodology for network of smart water meters.

Time series representing values from smart meter network are then processed by neural network based artificial intelligence algorithms (see Sect. 2.2 for algorithm details). Afterwards mean prediction (10 step prediction) and prediction intervals based on TimesNet [8], NHITS [9], and TSMixer [10] AI algorithms are calculated. Next anomaly or failure condition is checked calculated with present on-line extracted values from smart water meter network. Anomaly or failure condition takes into consideration the number of times the current on-line extracted signal values exceed the prediction interval per unit of time (details are presented in Sect. 2.3). Finally, the report is passed to the maintenance staff about possible problems concerning selected part of water infrastructure.

2.2 The Water Flows Time Series Forecasting Using Machine Learning Models

In the processes of forecasting future values of analyzed time series, approaches based on various types of computational intelligence are increasingly used, which acquire the ability to predict as a result of learning processes.

It was selected three AI algorithms with different characteristic. In analyzed water flow time series signals both daily, weekly periodicity and intricate temporal variations are observed. TimesNet algorithm captures two types of temporal variations defined as an intraperiod-variation and interperiod-variation. Intraperiod-variation indicates to short term patterns within period while interperiod-variation may represent long-term trends between different periods [8]. Second TSMixer model was used because of his performance in case of long-term predictions and computational performance because of MLP Multilayer Perceptron usage across time and feature dimensions. Such operations corresponds to time-mixing and feature-mixing operations, which captures both

temporal patterns and cross-variate information [10]. Last N-HITS model was designed as univariate model utilizing such techniques as: hierarchical interpolation, multi-rate input sampling in order to provide long-horizon and short-term predictions. Time series are sampled in different rates so model can learn and mix predictions of time series in different scales [9].

Temporal 2D-Variation Modeling for General Time Series Analysis (TimesNet)

The described method focuses on modeling time variability, which is a common key problem in case of extensive analytical tasks. Typically, the methods used attempt to do this directly from the 1D time series level, which is extremely difficult, due to complex time patterns. Based on the observation of multi-period nature, in the proposed method the time series were divided into numerous intra- and inter-temporal changes in terms of complex time changes. To address the limitations of 1D time series representation capabilities, we extended temporal variation analysis to 2D space by transforming the 1D time series into a set of 2D tensors based on multiple periods. Then, the implemented transformation allowed us to embed intra- and intertemporal variations in the columns and rows of 2D tensors, respectively, so that 2D variations can be easily modeled using 2D kernels. Technically, TimesNet with Times Block is proposed as a general framework for time series analysis tasks. Times Block can adaptively discover multi-temporality and extract complex temporal variations from transformed 2D tensors with a parameter-efficient seed block. In this context, the proposed solution, i.e. the TimesNet network, provides a consistent and modern standard in the main tasks of time series analysis, especially in short- and long-term forecasting [8].

An All-MLP Architecture for Time Series Forecasting (TSMixer)

Real-world time-series datasets are often multivariate with complex dynamics. To capture this complexity, high capacity architectures like recurrent- or attention-based sequential deep learning models have become popular. In our solution, we propose TSMixer, a novel architecture for time series forecasting, designed using MLP instead of the commonly used RNN attention mechanism, to achieve better generalization with a simple architecture. TSMixer is based on mixing operations along both the time and feature dimensions to extract information efficiently [10]. On popular academic benchmarks, the simple-to-implement TSMixer is comparable to specialized state-of-the-art models that leverage the inductive biases of specific benchmarks. On the challenging and large scale M5 benchmark, a real-world retail dataset, TSMixe demonstrates superior performance compared to the state-of-the-art alternatives [11].

Neural Hierarchical Interpolation for Time Series Forecasting (NHITS)

Long-horizon forecasting is a very difficult task. Two common challenges affecting the task are the volatility of the predictions and their computational complexity. NHITS model addresses both challenges by incorporating novel hierarchical interpolation and multi-rate data sampling techniques. These techniques enable the proposed method to assemble its predictions sequentially, emphasizing components with different frequencies and scales while decomposing the input signal and synthesizing the forecast. It is the proved that the hierarchical interpolation technique can efficiently approximate arbitrarily long horizons in the presence of smoothness [9].

2.3 The Condition of Anomaly and Failure Detection

Taking into account the nature and character of water flow, significant variability of data in the analyzed time series can be expected. The reasons for this phenomenon are usually related to possible seasonal changes in water consumption by residents or the occurrence of unpredictable external factors [12]. Therefore, the following statistical condition should be formulated, the fulfillment of which will indicate the detection of an anomaly or failure for the new data set

$$x_i \notin (\mu - 3\sigma, \mu + 3\sigma) \quad i = 1, 2, \dots, n, \quad (1)$$

where $\{x_1, x_2, \dots, x_n\}$ is time series limited by n elements' analysis window, μ is estimated from forecasts of the machine learning models in the analysis window, and σ is standard deviation of elements of the examined time series in reference to such mean. Detection condition was formulated based on reference of anomaly events detected during work of test bed network.

3 Experimental Smart Water Meter Network and Results Achieved for Proposed Solution

Test smart water meter network and smart water meter construction were part of research and development project realized with cooperation with Municipal Waterworks and Sewerage in Bydgoszcz, Poland company [13]. Test smart water networks consist of meters communicating through embedded GPRS/LTE modules and consist of accumulators charged by current provided by micro water turbine ensuring independent power supply provided over the life of connected batteries. Test bed network consisted of 100 water meters localized in Bydgoszcz (Poland) – 46 m and Świdnica (Poland) – 56 m. Meters were usually installed in the basements of buildings. In these locations 2023 people were registered. Most of people lived there in multi-family buildings.

In the article water consumption from multi-family buildings participating in the project was analysed. Data was gathered from test network in one-year period of time. Time series where were created from values sampled in one-minute period of time. After steps connected with data gathering and preprocessing (see Sect. 2.1 for more information) time series representing water flow [liter/minute] were passed to the input of neural network based algorithms. Achieved results rely on characteristic of selected neural network based algorithms and very strong on characteristic of examined time series signal. Properties of examined neural network based algorithms in the context of analyzed signal were presented in more details in Sect. 2.2. Analyzed signal in test network representing water flows is very demanding for prediction algorithms because of noise-like sophisticated and time varying characteristics without obvious patterns in signal structure.

In Fig. 2, Fig. 3, Fig. 4 it can observed water flow signals from the same period of time in normal working day captured in different year seasons: winter (Fig. 2), spring (Fig. 3) and summer (Fig. 4).

Three neural network based algorithms NHITS, TimesNet and TSMixer (see Sect. 2.1 for details) algorithms were tested for possible usage in our anomaly or failure detection

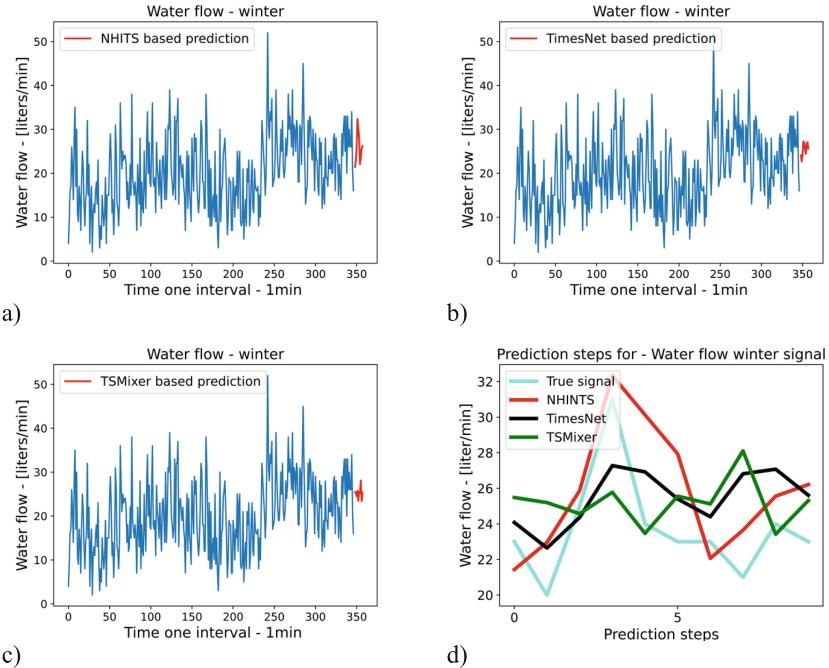


Fig. 2. Example of 10 step prediction intervals applied to winter water flow time series for an example of multi-family building, achieved by three artificial intelligence models NHITS a), TimesNet b) and TSMixer c) used by proposed anomaly and failure methodology together with magnification view comparing 10-step prediction results for three artificial intelligence methods presented in d).

methodology. Predictions of water flow signals were calculated in real time. Measurements were gathered in 1 min time period so it was possible to calculate mean prediction and prediction intervals for selected AI algorithms in real time.

Analysis time window has been adjusted to 500 samples length. Prediction interval was set to 10 one minute steps. In order to predict future values of time series supervising learning approach was used. First daily water usage time series were split into train set (80% of data), test (10% of data) and validation data (10% of data). Data have to be split into multiple input sequences. Based on these sequences model will learn and generalize across. In the next step selected models were evaluated using walk-forward validation. In case of walk-forward validation model do a forecasts for each observation in the test dataset one at a time. Real observations consisting of water flow measurements are then added to the input time series in order to make subsequent predictions on the next time step. Prediction period is sufficient from maintenance point of view and allows for acceptable time reaction. In Fig. 2a, Fig. 2b, Fig. 2c–Fig. 4a, Fig. 4b, Fig. 4c 10-step prediction values respectively for NHITS, TimesNet and TSMixer artificial intelligence algorithms can be observed. Magnification view comparison results achieved for every prediction method can be observed together with original signal in Fig. 2d (winter signal),

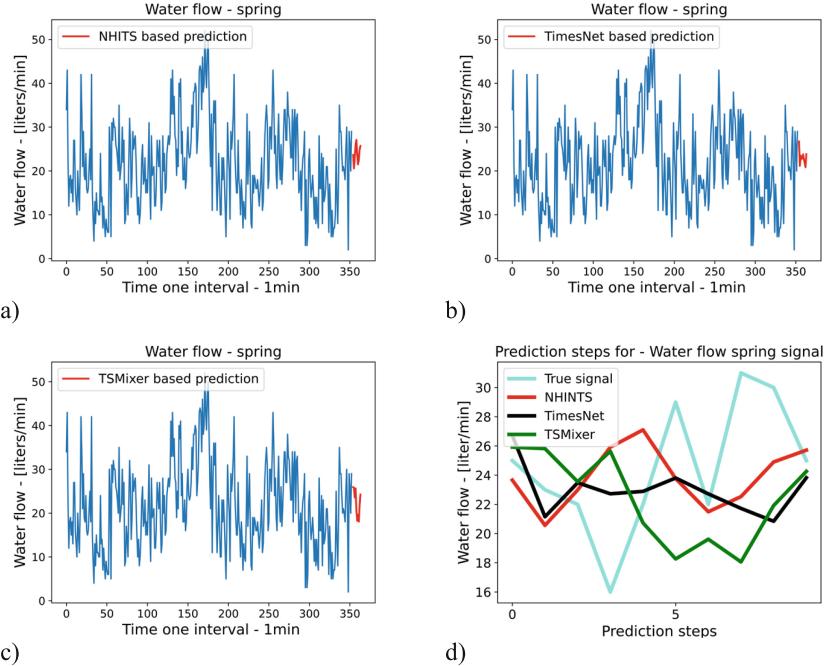


Fig. 3. Example of 10 step prediction intervals applied to spring water flow time series for an example of multi-family building achieved by three artificial intelligence models NHITS a), TimesNet b) and TSMixer c) used by proposed anomaly and failure methodology together with magnification view comparing 10-step prediction results for three artificial intelligence methods presented in d).

Fig. 3d (spring signal), Fig. 4d (summer signal). Accuracy prediction comparison for NHITS, TimesNet, TSMixer artificial intelligence models are presented in Table 1.

Result of RMSE and MAPE are based on four consecutive algorithms calculations applied to a given signal representing normal working day for all seasons in a year. In case of our signal representing water flows the best results were achieved for NHITS algorithm for summer and autumn signal while for winter and spring signal NHITS was comparable or very slightly worse than TimesNet neural network model. It can be stated that for this kind of signal in most cases NHITS provided the more accurate predictions. Similar relationship to prediction accuracy can be noticed in case of detection rate and false positive results concerning anomaly or failure events registered during analyzed period of time.

Results of detection rate and false positive presented in Table 2 were divided into all year's periods. Detection rate is defined as: DR [%] = number of real anomalies detected/number of all anomalies while FP [%] = number of anomaly indications that appears to be anomaly free/number of all anomaly free indications.

Anomalies in examined test network had different sources:

- water leakage (e.g. because of broken pipes or water valves),
- problems with wireless communications from smart water meters,

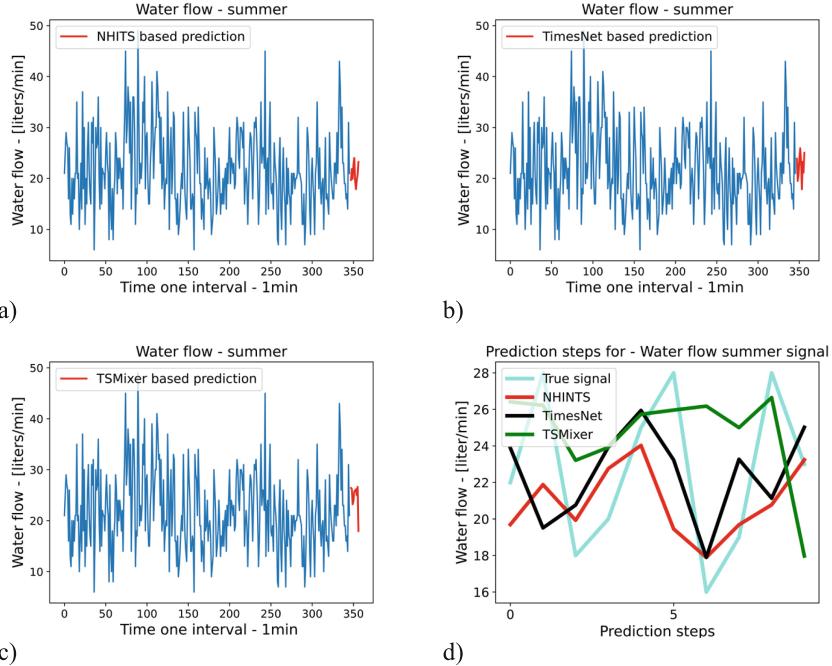


Fig. 4. Example of 10-step prediction intervals applied to summer water flow time series for an example of multi-family building achieved by three artificial intelligence models NHITS a), TimesNet b) and TSMixer c) used by proposed anomaly and failure methodology together with magnification view comparing 10-step prediction results for three artificial intelligence methods presented in d).

Table 1. Prediction accuracy for NHITS, TimesNet, TSMixer artificial intelligence models evaluated by: Root mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE).

AI based model	Winter RMSE	Winter MAPE[%]	Spring RMSE	Spring MAPE[%]	Summer RMSE	Summer MAPE[%]	Autumn RMSE	Autumn MAPE[%]
NHITS	3.10	11.35	5.08	17.31	4.29	13.35	4.87	10.35
TimesNet	2.98	11.30	5.03	15.40	4.42	16.23	6.15	13.54
TSMixer	3.57	12.42	6.78	20.85	4.86	20.71	6.45	13.36

- EMI Electromagnetic interferences from different local sources that had an impact on transmission, smart water meters failures (e.g. broken electronic controller),
- not sufficient amount of energy from water micro turbine.

Taking into account all year seasons, the best results were achieved for NHITS neural network algorithm. TimesNet algorithm had very slightly better results than NHITS for winter and spring signals but for the rest of examined period of time TimesNet neural network had significantly worse results than NHITS. It can be noticed here that

Table 2. Detection rate and False Positive results for proposed anomaly and failure detection methodology with three versions utilizing NHITS, TimesNet and TSMixer artificial intelligence models.

AI based model	Winter DR[%]	Winter FP[%]	Spring DR[%]	Spring FP[%]	Summer DR[%]	Summer FP[%]	Autumn DR[%]	Autumn FP[%]
NHITS	97.42	4.34	95.56	5.53	96.67	5.15	98.11	4.25
TimesNet	97.56	4.21	96.63	5.12	95.72	6.08	95.14	6.52
TSMixer	94.46	6.44	93.67	6.85	93.36	7.58	94.88	7.02

these results can't be approximated automatically to any kind of univariate time series signals. Especially in case of univariate time series algorithms have to be carefully selected. Sometimes more sophisticated neural network algorithms doesn't provide the best prediction results.

Proposed in article solution achieved promising results. Detection rate and false positive for NHITS based solution changes respectively 98.11%–95.56% and 4.25%–5.53%. Second best model TimesNet achieved 97.56%–9.14% for detection rate and 4.21%–6.52% for false positive.

Solution proposed in this article was evaluated with the use of real world smart metering network and helped maintenance staff to shorten reaction time in case of anomaly or water infrastructure failure. Therefore, it helps to reduce costs related to leaks and shorten repair times. As a result, a higher quality of operation of the water supply network will be ensured.

4 Conclusions

In this article solution for anomaly and failure detection for smart water meter network was proposed. Both the anomaly detection method and the intelligent water meter and the network were proposed and implemented as part of the research project. Proposed solution based on artificial intelligence algorithms helps maintenance staff to manage smart water metering network. Solution with three artificial intelligence methods NHITS, TimesNet and TSMixer were proposed. The best results were achieved for NHITS (DR 98.11%–95.56% and FP 4.25%–5.53%) and TimesNet (DR 97.56%–9.14% and FP 4.21%–6.52%). As a consequence of proposed system usage, process of seeking and repairing water infrastructure can be shorten, so people have more uninterrupted access to very important critical infrastructures.

In the future works it is planned to investigate how model ensemble algorithms can improve prediction efficiency and detection rate of anomaly events in intelligent water system. Additionally it is planned to add measurements from other infrastructure elements like hydrophore plants.

References

1. Li, J., Yang, X., Sitzenfrei, R.: Rethinking the framework of smart water system: a review. *Water* **12**(2) (2020)
2. Billewicz, K.: Smart metering. PWN (2020)
3. Blokdyk, G.: Advanced Metering Infrastructure (AMI), 5STARCooks (2022)
4. Ji, M., Yi, G., Jung, J.: Central prediction system for time series comparison and analysis of water usage data. *IEEE Access* **8**, 10342–10351 (2020)
5. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: Statistical and machine learning forecasting methods: concerns and ways forward. *PLoS ONE* **13** (2018)
6. Xie, M., Han, S., Tian, B., Parvin, S.: Anomaly detection in wireless sensor networks: a survey. *J. Netw. Comput. Appl.* **34**(4), 1302–1325 (2011)
7. Heo, S., Nam, K., Loy-Benitez, J., Yoo, C.: Data-driven hybrid model for forecasting wastewater influent loads based on multimodal and ensemble deep learning. *IEEE Trans. Industr. Inf.* **17**(10), 6925–6934 (2021)
8. Haixu, W., Tengge, H., Yong, L., Hang, Z., Jianmin, W., Mingsheng, L.: TimesNet: temporal 2D-variation modeling for general time series analysis. In: International Conference on Learning Representations (2022)
9. Challu, C., Olivares, K.G., Oreshkin, B.N., Ramirez, F.G., Canseco, M.M., Dubrawski, A.: NHITS: neural hierarchical interpolation for time series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence (2023)
10. Chen, S., Li, Ch., Yoder, N.C., Arik, S.O., Pfister, T.: TSMixer: an all-MLP architecture for time series forecasting. *Trans. Mach. Learn. Res.* **09** (2023)
11. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: M5 accuracy competition: results, findings, and conclusions. *Int. J. Forecast.* **38**(4), 1346–1364 (2022)
12. Cichoń, T., Królikowska, J., Nachlik, E.: Gospodarka wodomierzowa. Wyzwania i możliwości. Wydawnictwo Politechnika Krakowska (2020)
13. Miejskie Wodociągi i Kanalizacja w Bydgoszczy – sp. z o.o.. <https://mwik.bydgoszcz.pl/>. Accessed 07 May 2024



Design and Implementation of a Fast, Platform-Adaptive, AIS-20/31 Compliant PLL-Based True Random Number Generator on a Zynq 7020 SoC FPGA

Oğuz Yayla¹ and Yunus Emre Yılmaz^{1,2(✉)}

¹ Institute of Applied Mathematics, Middle East Technical University,
06800 Ankara, Turkey
oguz@metu.edu.tr

² Aselsan Inc., Ankara, Turkey
seylmz@gmail.com

Abstract. Phase-locked loops (PLLs) integrated within field-programmable gate arrays (FPGAs) or System-on-Chip FPGAs (SoCs) represent a promising approach for generating random numbers. Their widespread deployment, isolated functionality within these devices, and robust entropy, as demonstrated in prior studies, position PLL-based true random number generators (PLL-TRNGs) as highly viable solutions for this purpose. This study explicitly examines PLL-TRNG implementations using the ZC702 Rev1.1 Evaluation Board featuring the Zynq 7020 SoC from Xilinx, utilizing a configuration involving three such boards for experimental validation. Parameters governing the PLL-TRNG are optimized using a backtracking algorithm. Additionally, a novel methodology is proposed to enhance the rate of random data bit generation while preserving entropy characteristics. Performance metrics are rigorously evaluated against the criteria set by the German Federal Office for Information Security (BSI) AIS-20/31 Tests, accompanied by detailed descriptions of the implementation process.

Keywords: random number generation · PLL-TRNG · AIS-20/31

1 Introduction

Random numbers are critical and also the most required elements in the majority of cryptographic systems. For these numbers, excellent statistical quality is required, and that is provided by true random number generators (TRNGs) that are based on some physical random phenomena to guarantee unpredictability. Different types of entropies, concerning the application and the chosen technology, can be harvested. In this research, the jitter is selected as the entropy source, and the jitter can be defined as the fluctuation of the period in the clock in the time domain.

In this study, Phase-Locked Loops (PLLs) are selected as the source of the entropy harvester. These PLLs, which vary in number and properties, can be found in any field-programmable gate array (FPGA) or System-on-Chip FPGA (SoC). PLLs are feedback control systems that automatically adjust the phase of a locally generated signal to align with the phase of an input signal. Due to their internal structures and inherent noise, the clocks within PLLs exhibit jitter. This jitter serves as a source of randomness, forming the basis for PLL-based True Random Number Generators (PLL-TRNGs). The PLL-TRNG boasts a straightforward and comprehensive design, utilizing coherent sampling. Furthermore, PLLs have isolated locations in FPGAs and SoCs, and previous works about PLL-TRNG conclude that PLL-TRNG has good cryptographic properties. The PLL-TRNG design using one PLL is explained in [1–3], while the design with two PLLs is explained in [4]. Moreover, in [5], the stochastic model of the PLL-TRNG is presented. In addition to these works, details of PLL-TRNG implementation are analyzed in [6] and [7].

The primary challenge in PLL-TRNG design is the selection of optimal PLL settings from a vast configuration space. The chosen parameters must yield both a sufficient entropy rate and an adequate output bit rate. This study adopts the parameter determination process outlined in [10]. Notably, the backtracking algorithm employed in [10] offers significant advantages over previous search algorithms for PLL-TRNG design, as presented in [1,8,9].

While it is advantageous to implement PLL-TRNG considering its high entropy and isolated locations of PLL, one of the main drawbacks of the PLL-TRNG is its relatively low random data output speed. A new PLL-TRNG method using four PLLs is proposed to overcome this disadvantage. For this purpose, we chose Xilinx Zynq 7020 SoC instead of an FPGA. Since they offer higher integration, lower power, smaller board sizes, and higher bandwidth communication between the processor and FPGA.

Once random numbers are generated, their quality must be assessed. For this evaluation, the AIS-20/31 standard [11], a methodology proposed by the German Federal Office for Information Security (BSI), is chosen.

This research culminates in the implementation of a four-PLL True Random Number Generator (4-PLL TRNG) on the Xilinx Zynq 7020 SoC. To elucidate the design progression, a referenced configuration utilizing two PLLs and two intermediate configurations utilizing three PLLs are developed. PLL parameter optimization is achieved through a backtracking algorithm as outlined in [10]. Subsequently, four distinct PLL-TRNG configurations are implemented on the Xilinx (AMD) ZC702 Evaluation Kit Rev1.1, incorporating the Zynq 7020 SoC. To ensure design independence from the specific board, three evaluation kits are employed. Rigorous testing against AIS-20/31 standards is conducted on the generated random data. Finally, a comparative analysis with existing PLL-TRNG implementations is performed.

Our work presents two primary contributions:

- We design a new structure of PLL-TRNG that is adaptive to new FPGAs or SoCs, which can be used to increase the bit rate without worsening cryptographic properties.
- We conduct our new, fast, and adaptive design implemented on Xilinx Zynq 7020 SoC with respect to AIS-20/31 Tests and compare our results with previous works.

The paper is organized as follows. Section 2 provides the basic background information to explain how PLL-TRNG works. In Sect. 3, the implementation details of our proposed PLL-TRNG are explained. In Sect. 4, the test results of one referenced and three proposed PLL-TRNGs are presented and compared with previous works. In the end, Sect. 5 concludes the paper.

2 Background Information About PLL-TRNG Implementation

2.1 Basics of PLL

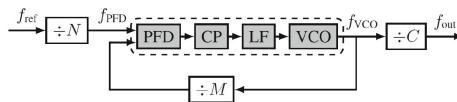


Fig. 1. Block diagram of a PLL (PFD: phase frequency detector, CP: charge pump, LF: loop filter, VCO: voltage-controlled oscillator) [10]

Table 1. Table of ranges of possible values for the PLL parameters and frequencies for Zynq-7000 SoC [14, 15]

Parameters	Xilinx Zynq-7000	
	Min	Max
f_{ref} (MHz)	19	800
P_{VCO}	1	1
M	2	64
N	1	56
C	1	128
f_{PFD} (MHz)	19	450
f_{VCO} (MHz)	800	1600
f_{out} (MHz)	6.25	464

A phase-locked loop (PLL) is a circuit (as depicted in Fig. 1) that uses an input signal to synchronize a signal from an embedded oscillator on it. The grey blocks represent the analog components, which cannot be parameterized, whereas the M , N , and C integer division coefficients, depicted in white blocks, need to be configured. These coefficients are essential for calculating the output frequency of the PLL (f_{out}) from the reference frequency (f_{ref}), as described in Eq. (1).

$$f_{out} = f_{ref} \times \frac{M}{N \times C} \quad (1)$$

2.2 Random Bit Generation Principle of the PLL-TRNG

The working principle of the PLL-TRNG with one PLL, and also two PLL versions of PLL-TRNG, is presented in Fig. 2.

The jittered clock signal clk_1 from the PLL is sampled by a D flip-flop (D-FF) using the reference clock signal clk_0 . The 1-bit counter records the number of samples that equal one. Due to the frequency relationship established by the PLL, a pattern with a period $T_Q = K_D \times T_0 = K_M \times T_1$ emerges at the flip-flop output. As a result, some samples are consistently one (shown as blue in Fig. 2 and these are 4th and 7th dots), some are always zero (shown as green and these are 2nd and 5th dots), and others are random (shown as red and these are 1st, 3rd, 6th, and 8th dots). By applying the coherent sampling principle and rearranging the samples based on their positions, the waveform of one period of clk_1 can be reconstructed [1, 16]. This work adopts a two PLL-TRNG architecture as a reference model due to its better performance characteristics. The incorporation of two PLLs significantly enhances design flexibility by expanding the practical operating ranges for critical parameters, K_M and K_D , consequently increasing attainable bit and entropy rates. Moreover, this configuration substantially reduces autocorrelation between output bits. While incurring increased implementation costs, these can often be mitigated through resource sharing with other system components, as proposed in [8].

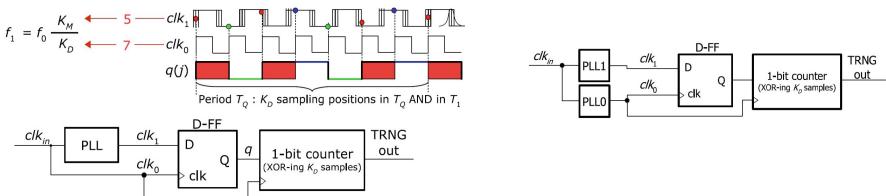


Fig. 2. [Left Figure]: Principle of the PLL-TRNG with one PLL [16] [Right Figure]: PLL-TRNG with two PLLs Configuration [16]

In this two PLLs case, firstly, as it is stated in Fig. 2:

$$\frac{f_1}{f_0} = \frac{K_M}{K_D} \quad (2)$$

where K_M and K_D are integer values representing frequency multiplication and division factors, depending on the configuration of PLLs. Each PLL has its multiplication and division factors. Moreover, they are related to K_M and K_D as:

$$K_M = K_{M_1} \cdot K_{D_0} \quad (3)$$

$$K_D = K_{M_0} \cdot K_{D_1} \quad (4)$$

The output (Q) of DFF in the left part of Fig. 2 has a pseudo-random pattern with a certain period. After XORing that pattern in the decimator or 1-bit counter, the bit rate of the PLL-TRNG is defined as follows:

$$R = \frac{f_0}{K_D} = \frac{f_1}{K_M} \quad (5)$$

The entropy rate per bit at generator output depends on the parameters of the jitter and on the parameters of the generator, which are characterized by its sensitivity to the jitter:

$$S = \Delta^{-1} = f_0 \cdot K_M = f_1 \cdot K_D \quad (6)$$

The design of PLL-TRNG relies on choosing appropriate PLL multiplication and division factors. However, selecting these factors can be challenging due to the physical constraints of the PLL, such as the maximum and minimum values of N , M , C , and the input, output, PFD, and VCO frequency range. Consequently, determining these values is an optimization problem, and our solution to this problem is explained in Sect. 3.2 for Zynq 7020 SoC values listed in Table 1.

3 PLL-TRNG Implementation Details

3.1 Implemented PLL-TRNG Configurations

A primary limitation of PLL-TRNGs is their comparatively low output data rate. To address this constraint, this work proposes a methodology to enhance output capacity by leveraging additional PLLs available within the SoC. The Zynq 7020 SoC, featuring four PLLs, represents the upper bound for this implementation. However, prior to full-scale implementation, intermediate configurations employing three PLLs are investigated to facilitate a systematic design process. This study elucidates the design rationale for the four-PLL system by providing detailed explanations of these intermediate steps. Consequently, four distinct PLL-TRNG configurations are presented in Table 2 and visually depicted in Fig. 3:

Table 2. Configurations of PLL-TRNG Implementations

Codes of PLL-TRNG Designs Depicted in Fig. 3	Number of PLLs	Purpose of Use of PLL		PLL-TRNG Design Type
		As Reference Clock	As Jittered Clock	
(a)	2		1	Referenced Design
(b)	3	1	2	Intermediate Step for Proposed Approach
(c)	3	2	1	Intermediate Step for Proposed Approach
(d)	4	2	2	Proposed Design

3.2 Determining PLL-TRNG Parameters

In this work, as the parameter search algorithm, the backtracking algorithm in [10] is selected. Given a set of variables explained in Sect. 2.1 and Sect. 2.2 and constraints listed in Table 1, this backtracking method iteratively investigates potential solutions. Unlike a brute-force approach, it promptly eliminates any variable values that fail to meet a constraint, then backtracks to explore other possible values until all valid solutions are identified. The algorithm detailed in [10] involves determining the PLL-TRNG parameters that comply with both physical constraints and application requirements.

The code in the backtracking is open-source shared in [19]. Hence, we can modify it for Zynq 7020 SoC parameters provided in Fig. 1. Table 1 is generated with PLL properties of the SoC except for f_{out} which is determined concerning the maximum frequency value of BUFG clock buffer in Zynq 7000 Series [15]. BUFG must be used in the SoC design, and hence, it restricts f_{out} value for the search algorithm.

After generating results for our case, the results of the algorithm are ordered with respect to three different configurations. Those are the maximum bit rate (max. R), the maximum sensitivity to jitter (max. S), and the maximum $R \cdot S$ value as the optimization between max. S and max. R .

After obtaining the candidate results for three different configurations, those results must be tested with one more criterion. The sampling process of the jittered clock with the reference clock is illustrated in Fig. 2. In order to obtain random numbers at the output of this PLL-TRNG, at least one sample is required to be affected by the jitter. This necessitates that the distance between any edge of clk_0 and its corresponding edge on clk_1 must be less than Δ . This condition is met if the following condition holds [4, 7]:

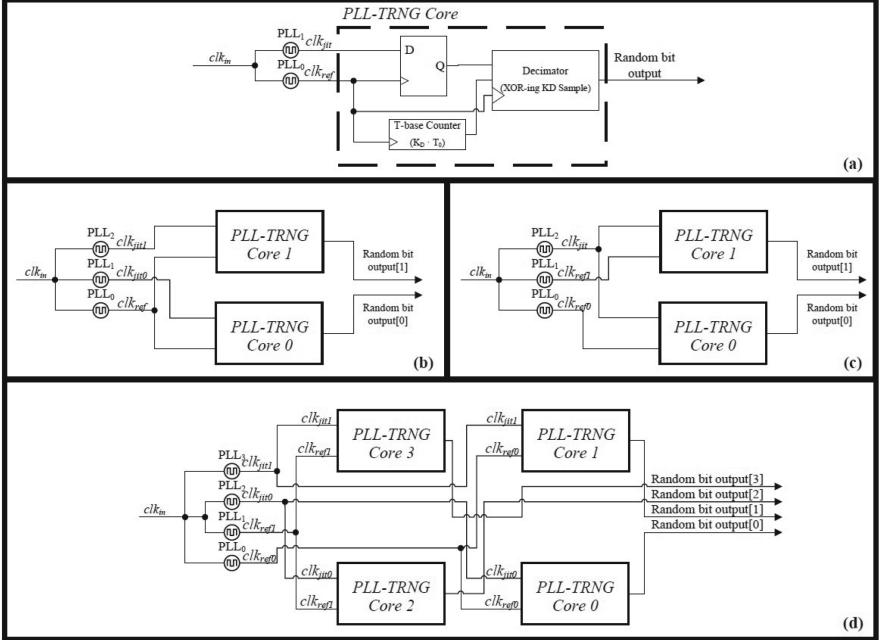


Fig. 3. Implemented PLL-TRNG Configurations: (a), (b), (c), and (d)

$$\sigma_{jit} > \max(\Delta T_{min}) \quad (7)$$

where σ_{jit} is the standard deviation of the jitter at the output of the PLL, and $\max(\Delta T_{min})$ is the largest distance between the two closest edges of clk_0 and clk_1 . This can be computed as [4, 7]:

$$\max(\Delta T_{min}) = \frac{T_{clk_0}}{4K_M} \gcd(2K_M, K_D) = \frac{T_{clk_1}}{4K_D} \gcd(2K_M, K_D) \quad (8)$$

where \gcd is the greatest common divisor of two integers.

Upon executing the backtracking algorithm and obtaining results for the selected SoC, the maximum value of $\max(\Delta T_{min})$ can be determined. However, accurately measuring or estimating σ_{jit} presents significant challenges. At this juncture, the estimation tool named *Clocking Wizard* in Vivado 2019.1 can be utilized. This tool provides an estimation of the jitter at the PLL's output clock, given the PLL parameters. Consequently, the results from the backtracking algorithm are first examined, and max. R , max. S and the max. $R \cdot S$ are identified. These three candidates are then evaluated against Eq. (7). Candidates failing to satisfy the equation are discarded, and alternative candidates from the backtracking results are considered.

The results of the search algorithms are listed in Table 3. As it can be seen, all the selected configurations satisfy Eq. (7).

Table 3. Determined Parameters for the PLL-TRNG Implementations

Configuration	f_{ref} (MHz)	M_0	N_0	C_0	M_1	N_1	C_1	f_o (MHz)	f_a (MHz)	K_M	K_D	R (Mbit/s)	S (ps^{-1})	R-S	σ_{jit}	$\max(\Delta T_{min})$
Max R	125	51	4	4	11	1	3	398.438	458.333	176	153	2.60417	0.07013	0.18263	76.706	3.56506
Max S	125	51	4	4	32	3	3	398.438	444.444	512	459	0.86806	0.204	0.177084	100.882	1.22549
Max R-S	125	37	5	2	32	3	3	462.5	444.444	320	333	1.38889	0.148	0.20556	100.882	1.68919

3.3 Implementation Setup

Xilinx/AMD ZC702 Rev1.1

Evaluation Board

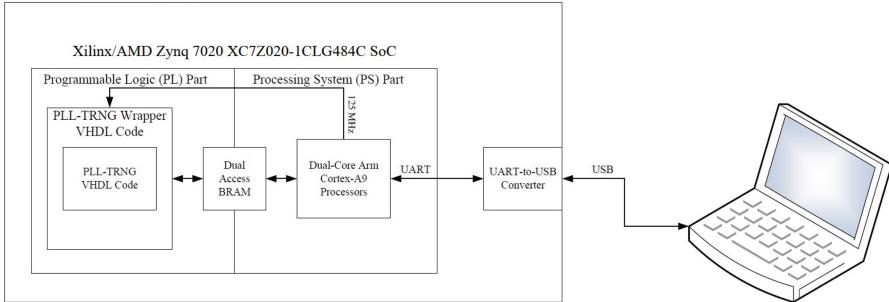


Fig. 4. Block Diagram of Implementation Setup

The implementation setup employed in this study is illustrated in Fig. 4. It utilizes the ZC702 Rev1.1 Evaluation Board [12], which incorporates the Zynq 7020 XC7Z020-1CLG484C SoC to facilitate the implementation of four distinct PLL-TRNG configurations as detailed in Sect. 3.1. In the Programmable Logic (PL) section, four distinct designs, specified in Table 2, are developed using Vivado 2019.1 [13] in VHDL. To enable real-time transmission of generated random numbers to a personal computer (PC), a dual-access Block RAM (BRAM) is employed. One port of this BRAM is connected to the PL, while the other is connected to the processing system (PS) section. The requisite code for the PS section is written in the C programming language. The PL section generates random numbers and writes a predefined value to a specific BRAM address to indicate that the random bits are ready. Once this indication is given, the software in the PS section outputs the random bits to the UART serial port, which are then converted to USB and transmitted to the PC. The received bits on the PC are saved in their ASCII-coded hexadecimal form and later converted to

binary form offline to serve as input for AIS-20/31 Tests [17]. Both Procedure A and Procedure B Tests of AIS-20/31 are conducted for each result. Given that these tests require ~ 7 Mb of random bits, each output file is generated to have a size of ~ 7.2 Mb. Additionally, a 125 MHz clock frequency is selected for the system's main clock (clk_{in}) due to timing constraints inherent in the SoC.

In conclusion, three ZC702 Rev1.1 Evaluation Boards are employed to demonstrate that the implemented PLL-TRNG configurations are not specific to a particular device. The backtracking algorithm and the elimination criteria outlined in Eq. (7) are used to determine the PLL-TRNG configuration parameters for maximizing S , R , and the product $R \cdot S$. Subsequently, five random output bit files are generated for each of the four configurations described in Sect. 3.1 and tested on the three evaluation boards. The results are stored on a PC, and AIS-20/31 Tests are conducted. The outcomes of these tests are detailed in Sect. 4.

4 Results and Comparisons

The implementation results are presented in Table 4. In this table, each row corresponds to a unique configuration defined by the number of PLLs in the PLL-TRNG and the parameter configuration. By considering our four different PLL-TRNG configurations and three different PLL parameter selections, we have twelve distinct rows, in other words, twelve different results. For each row, five different ~ 7.2 Mb random number files are generated for each of the ZC702 Boards. Hence, the arithmetic mean of these fifteen values is used for the Shannon Entropy calculation in the table.

Although the Shannon Entropies are provided by the test suite, it is not indicated in Table 4, but it must be emphasized that all four PLL-TRNG designs passed all the AIS-20/31 Tests for all three different configurations on three distinct boards for all generated files.

Table 5 provides a comparative analysis of our work with previously implemented PLL-TRNGs. The results indicate that our 4-PLL implementation sig-

Table 4. PLL-TRNG Implementation Results

PLL Configurations	Parameter Configurations	R (Mbit/s)	Output Bit Rate (Mbit/s)	S (ps^{-1})	R-S	Entropy (Shannon)
2-PLL with one reference clock and one jittered clock	Max. R	2.60417	2.60417	0.07013	0.18263	0.99999986833568
	Max. S	0.86806	0.86806	0.204	0.17708	0.99999986516413
	Max. R-S	1.38889	1.38889	0.148	0.20556	0.99999981069240
3-PLL with one reference clock and two jittered clocks	Max. R	2.60417	5.20834	0.07013	0.18263	0.99999976364641
	Max. S	0.86806	1.73612	0.204	0.17708	0.9999997771549
	Max. R-S	1.38889	2.77778	0.148	0.20556	0.99999980834110
3-PLL with two reference clocks and one jittered clock	Max. R	2.60417	5.20834	0.07013	0.18263	0.99999985962200
	Max. S	0.86806	1.73612	0.204	0.17708	0.99999961780714
	Max. R-S	1.38889	2.77778	0.148	0.20556	0.9999996076834
4-PLL with two reference clocks and two jittered clocks	Max. R	2.60417	10.41668	0.07013	0.18263	0.99999972332402
	Max. S	0.86806	3.47224	0.204	0.17708	0.99999971434251
	Max. R-S	1.38889	5.55556	0.148	0.20556	0.99999956486246

Table 5. PLL-TRNG Implementation Results Comparison with [7, 10, 18]

PLL Configurations	Parameter Configurations	Output Bit Rate (Mbit/s)	S (ps^{-1})	Entropy (Shannon)	Results from [10] for Xilinx Spartan-6			Results from [7] for Xilinx Spartan-6		
					Output Bit Rate (Mbit/s)	S (ps^{-1})	Entropy (Shannon)	Output Bit Rate (Mbit/s)	S (ps^{-1})	Entropy (Shannon)
4-PLL with two reference clocks and two jittered clocks for Zynq-7020	Max. R	10.41668	0.07013	0.999999972332402	1.042	0.094	1	0.555	0.0913	N/A
	Max. S	3.47224	0.204	0.999999971434251	0.521	0.167	0.99999	0.555	0.0913	N/A
	Max. R-S	5.55556	0.148	0.999999956486246	N/A	N/A	N/A	0.555	0.0913	N/A
PLL Configurations	Parameter Configurations	Output Bit Rate (Mbit/s)	S (ps^{-1})	Entropy (Shannon)	Results from [18] for Xilinx Spartan-6					
					Output Bit Rate (Mbit/s)	S (ps^{-1})	Entropy (Shannon)			
					1.042	0.094	1			
4-PLL with two reference clocks and two jittered clocks for Zynq-7020	Max. R	10.41668	0.07013	0.999999972332402	0.521	0.167	0.99999			
	Max. S	3.47224	0.204	0.999999971434251	N/A	N/A	N/A			
	Max. R-S	5.55556	0.148	0.999999956486246						

nificantly enhances the output bit rate of the PLL-TRNG design while maintaining robust cryptographic properties. Specifically, the table shows that a speed of approximately 10.4 Mb/s can be achieved, which is notably higher than any other reported PLL-TRNG implementation. Additionally, our results exhibit superior Shannon Entropy compared to earlier PLL-TRNG designs.

5 Conclusion

In this paper, we delineate the design and implementation procedures of an innovative and fast PLL-TRNG utilizing the coherent sampling method of jittered PLL clocks. For parameter selection, we employ the backtracking algorithm [10]. Unlike conventional designs, which typically incorporate two PLLs, our approach leverages four PLLs to enhance the output bit rate. The choice of four PLLs is constrained by the Xilinx Zynq 7020 SoC, which accommodates exactly four PLLs. Nevertheless, the methodology illustrated in Fig. 3 is adaptable to any FPGA or SoC platform, provided the target device includes at least three PLLs. This flexibility ensures the broad applicability of our approach across various hardware configurations.

We show that our proposed methods can generate random numbers with AIS-20/31 compliance. With their excellent results compared to the previous works, it can be concluded that our proposed method is promising.

The proposed PLL-TRNG approach demonstrates scalability and adaptability, rendering it suitable for integration into future FPGA and SoC designs featuring enhanced performance characteristics. Moreover, the proposed design constitutes a viable candidate for implementation as a TRNG core within an application-specific integrated circuit (ASIC) architecture.

Acknowledgements. We thank the anonymous reviewers whose comments/ suggestions helped improve and clarify this paper. Also, the authors acknowledge Aselsan Inc. for its support during the preparation of this paper and for providing three Xilinx ZC702 Evaluation Boards that were utilized during the implementation of the PLL-TRNG algorithms described in the paper.

References

1. Fischer, V., Drutarovský, M.: True random number generator embedded in reconfigurable hardware. In: Kaliski, B.S., Koç, K., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 415–430. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36400-5_30
2. Drutarovský, M., Simka, M., Fischer, V., Celle, F.: A simple PLL-based true random number generator for embedded digital systems (2004)
3. Liu, C., McNeill, J.: A digital-PLL-based true random number generator. PhD Res. Microelectron. Electron. **1**(2005), 113–116 (2005)
4. Fischer, V., Drutarovský, M., Šimka, M., Bochard, N.: High performance true random number generator in Altera Stratix FPLDs. In: Becker, J., Platzner, M., Vernalde, S. (eds.) FPL 2004. LNCS, vol. 3203, pp. 555–564. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30117-2_57
5. Bernard, F., Fischer, V., Valtchanov, B.: Mathematical model of physical RNGs based on coherent sampling (2010)
6. Petura, O.: True Random Number Generators for Cryptography: Design, Securing and Evaluation. Micro and Nanotechnologies/Microelectronics. Université de Lyon. English. NNT: 2019LYSES053. tel-02895861 (2019)
7. Allini, E.N.: Characterization, Evaluation and Utilization of Clock Jitter as Source of Randomness in Data Security. Cryptography and Security [cs.CR]. Université de Lyon. English. NNT: 2020LYSES019. tel-03207261 (2020)
8. Petura, O., Mureddu, U., Bochard, N., Fischer, V.: Optimization of the PLL-based TRNG design using the genetic algorithm (2017)
9. Allini, E.N., Petura, O., Fischer, V., Bernard, F.: Optimization of the PLL configuration in a PLL-based TRNG design (2018)
10. Colombier, B., Bochard, N., Bernard, F., Bossuet, L.: Backtracking search for optimal parameters of a PLL-based true random number generator (2020)
11. Killmann, W., Schindler, W.: A proposal for functionality classes for random number generators - Version 2.35 - DRAFT, Technical report, 2 September 2022
12. Xilinx Zynq-7000 SoC ZC702 Evaluation Kit. <https://www.xilinx.com/products/boards-and-kits/ek-z7-zc702-g.html>
13. Xilinx (AMD) Vivado 2019.1 Design Software for Xilinx (AMD) Adaptive SoCs and FPGAs. <https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/vivado-design-tools/archive.html>
14. 7 Series FPGAs Clocking Resources User Guide (UG472) (v1.14). https://docs.amd.com/v/u/en-US/ug472_7Series_Clocking
15. Zynq-7000 SoC: DC and AC Switching Characteristics (DS187) (v1.21). <https://docs.amd.com/v/u/en-US/ds187-XC7Z010-XC7Z020-Data-Sheet>
16. Fischer, V., Bernard, F., Bochard, N.: Modern random number generator design - case study on a secured PLL-based TRNG (2019)
17. BSI. Implementation of Test Procedure A and Test Procedure B for Application Notes and Interpretation of the Scheme (AIS) 20/31 Standard. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_31_testsuit_zip.zip
18. Petura, O., Mureddu, U., Bochard, N., Fischer, V., Bossuet, L.: A survey of AIS-20/31 compliant TRNG cores suitable for FPGA devices (2016)
19. The source code of the backtracking algorithm in [16]. <https://gitlab.univ-st-etienne.fr/sesam/pll-trng-constraint-programming/tree/master>



A Comparison of Frameworks for Heterogeneous Computing Using High-Performance Pattern-Matching for DNA Biometrics and Digital Forensics

Lia Stratan^(✉) and Ciprian Pungila

Faculty of Mathematics and Informatics, West University of Timisoara,
Timișoara, Romania
{lia.stratan00,ciprian.pungila}@e-uvt.ro
<https://info.uvt.ro>

Abstract. In this paper we explore popular parallel computing frameworks, used in high-performance pattern-matching applications involving heterogenous architectures, specifically based on CPU/GPU infrastructures, and with particular emphasis on the Aho-Corasick multiple pattern-matching approach and its applications to DNA sequence analysis and digital forensics. Our experiments showed that modern managed frameworks, such as SYCL, could serve as a viable replacement for dedicated frameworks such as CUDA and OpenCL, showcasing a comparable throughput, while being more flexible and developer-friendly.

Keywords: high-performance computing · heterogeneous architectures · pattern-matching · Aho-Corasick · parallel algorithms · DNA biometrics · digital forensics · CUDA · OpenCL · SYCL · GPU

1 Introduction

In the field of cybersecurity, DNA biometrics is a challenging topic, with major implications in criminal investigations specifically related to ancestry detection and traits analysis. To the same degree, digital forensics is an important aspect of cybersecurity, generally involving the finding of relevant data on large storage devices, to be used as evidence. Both fields are often coming down to pattern-matching of relevant data.

With the advent of heterogeneous computing environments, leveraging the computational power of GPGPUs (General Purpose Graphics Processing Units) has become a norm, due to their ability to handle massive parallelism. Frameworks like CUDA and OpenCL are predominantly used for this task, offering substantial performance gains over traditional CPU implementations. CUDA, developed by NVIDIA, provides a robust platform for GPU programming but is limited to NVIDIA hardware. OpenCL, on the other hand, offers portability

across different hardware platforms, although sometimes at the cost of performance due to its broader scope. However, more modern approaches, such as SYCL [16], are also now present in the field of vision of developers worldwide.

In this paper, we are focusing on performing a comparison of the popular frameworks for heterogeneous computing, with special focus on high-performance pattern-matching applications, a critical field to DNA biometrics and digital forensics. We aim to implement an efficient Aho-Corasick pattern-matching algorithm in the frameworks above, benchmarking them against each other.

2 Related Work

Since the introduction of the Aho-Corasick algorithm in 1975 [5], which expanded the Knuth-Morris-Pratt algorithm's capabilities to include multiple-pattern matching, the field of pattern matching has experienced tremendous developments. Applications for this algorithm are numerous and range from the original Linux grep utility [6] to sophisticated fields like digital forensics [13], text mining, intrusion detection [14], and DNA processing [8]. An increasing number of attempts have been made to modify the Aho-Corasick algorithm in order to meet certain domain needs due to the increasing size of datasets across various research domains. Although the use of GPUs [10] and heterogeneous architectures [11] for DNA pattern matching is widely recognized, some difficulties are still unexplored. Nishimura et al. [1] proposed rearranging the states of the Aho-Corasick finite-state machines to improve memory access patterns and cache utilization. By organizing the states according to their breadth-first traversal order, the authors achieved a 55% performance improvement on CPU implementations.

Lin et al. [2] introduced PFAC (Parallel Failure-less Aho-Corasick) - a multi-threaded approach to accelerate pattern matching on GPUs. The proposed algorithm allocates a thread for each byte of the input stream, thus eliminating the need for backtracking and failure transitions, which are part of the traditional Aho-Corasick implementation. This new version of the algorithm not only avoids the boundary detection problem but also shortens the average thread lifetimes which improves the overall efficiency. Lin et al., in a subsequent study from 2012 [3], introduced an architecture that uses perfect hashing to compress the state transition tables of pattern matching machines. This approach significantly reduces memory requirements and contributes to improved scalability and detection performance in network intrusion detection systems. Thambawita et al. [4] proposed an optimized version of PFAC algorithm specific for DNA sequence matching. Using the fact that DNA sequences only involve four nucleotides, they designed a specialized data structure that reduces the transition table size and utilizes GPU shared and texture memory for faster access. Pungila et al. [12] proposed a hybrid CPU/GPU storage model for pattern-matching applications, specifically aimed at Aho-Corasick and Commentz-Walter automata. They specifically focused on reduced video RAM storage and throughput efficiency for device-to-host and host-to-device memory transfer.

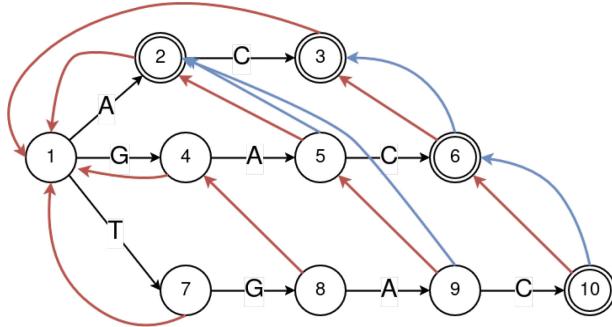


Fig. 1. The Aho-Corasick automaton for the following nucleotides AC, GAC, TGAC. The red arrows represent failure links and the blue arrows output links. The circled nodes indicate the end of a pattern.

2.1 Aho-Corasick Algorithm

The Aho-Corasick algorithm, developed by Alfred V. Aho and Margaret J. Corasick in 1975 [5], is particularly suited for pattern matching within DNA sequences due to its ability to locate multiple patterns simultaneously in linear time. This algorithm constructs a finite state machine from a set of keywords and uses it to search for occurrences of these keywords in a given text efficiently. For every new pattern to be added we follow the edges specified by the letters in the pattern, if there is no such edge, we create it. For example, if we were to add the pattern “GACT” in the automaton in Fig. 1, we would go through the edges 1–4, 4–5, 5–6 and then there would be no suitable edge, so we would create a new node - 11 with an edge “T” from node 6 to it.

Another important part of the Aho-Corasick Automaton is the presence of failure/mismatch and output links. Failure links are pointing to the longest proper suffix of the current path, where the path is an array of characters present on the links from node 1 to the current node. The path to node 9 can be thought of as “TGA”, with the longest proper suffix at node 5 - “GA”. The output links are pointing to the nodes which indicate the end of a pattern that can be reached by following the failure links from the current node. The output links exist for the purpose of taking into account patterns that get automatically matched by matching a longer pattern or a part of a longer pattern.

2.2 CUDA

CUDA (Compute Unified Device Architecture) [23] is NVIDIA’s proprietary parallel computing platform and API designed for computing on GPUs [15]. When executing a function (kernel), CUDA operates as a grid of thread blocks. The threads in a block can communicate using a fast shared memory. A group of blocks executing a function on the GPU form a grid. Each GPU has several Streaming Multiprocessors (SMs), which are capable of managing and executing

many threads simultaneously. Threads within a block are organized into warps, which are groups of 32 threads that execute instructions in lockstep. CUDA provides a highly optimized platform for running on NVIDIA GPUs but cannot run on other hardware such as CPUs or GPUs by other manufacturers.

2.3 OpenCL

OpenCL (Open Computing Language) [22] is an open standard for parallel programming of heterogeneous systems that allows execution across various types of processors, including GPUs, CPUs, and other processors. Like CUDA, OpenCL organizes execution into an architecture of compute units and work-items, grouped into work-groups. OpenCL defines a programming language (based on C99) for writing kernels, plus APIs to control the platform and execute programs on the compute devices.

OpenCL platforms consist of a host connected to one or more compute devices, each with multiple compute units, each containing multiple processing elements. Work-groups (analogous to CUDA’s thread blocks) execute concurrently on compute units, and work-items within a work-group execute on processing elements. Communication within a work-group can occur through local memory, and all work-items can read from or write to global memory. Work-items are also grouped into “wavefronts,” which are similar to CUDA’s warps, where a group of work-items executes the same operation in lockstep. OpenCL provides the flexibility to run on hardware from multiple manufacturers, unlike CUDA, which is specific to NVIDIA GPUs. This makes OpenCL a versatile choice for applications that need to run across different hardware platforms. Nonetheless, compatibility often comes at the cost of lower performance, and overall OpenCL has been empirically determined to provide slower performance than CUDA [20], while sometimes offering higher throughputs than the latter, as well [21].

2.4 SYCL

SYCL (pronounced “sickle”) [16] is a high-level programming model that builds on the concepts of OpenCL. It is a cross-platform abstraction layer that enables code for heterogeneous processors to be written in standard and modern C++ with the addition of simple template and lambda function annotations. SYCL abstracts device-specific details of parallel computation and integrates seamlessly with C++17/20 features, making it more accessible and flexible compared to lower-level APIs like OpenCL and CUDA. It supports single-source programming in which the device and host code are contained in the same file, simplifying the development process. Additionally, SYCL can inter-op with OpenCL, allowing developers to leverage existing OpenCL code.

The SYCL standard defines three primary concepts: buffers, accessors, and kernels. Buffers manage memory across devices, accessors define how kernels read and write memory, and kernels are functions that execute on the compute devices. SYCL achieves parallelism through a task graph of execution, which captures dependencies and allows efficient execution order of the tasks. This

execution model helps the compiler optimize data locality and minimize data movement. SYCL source code can be compiled to run on many different platforms using various SYCL implementations. Figure 2 shows some of these implementations and their possible targets. There are other prominent implementations such as AdaptiveCpp [17] not mentioned here in the interest of brevity. This flexibility ensures that SYCL can be used to develop applications that run efficiently on a wide range of hardware.

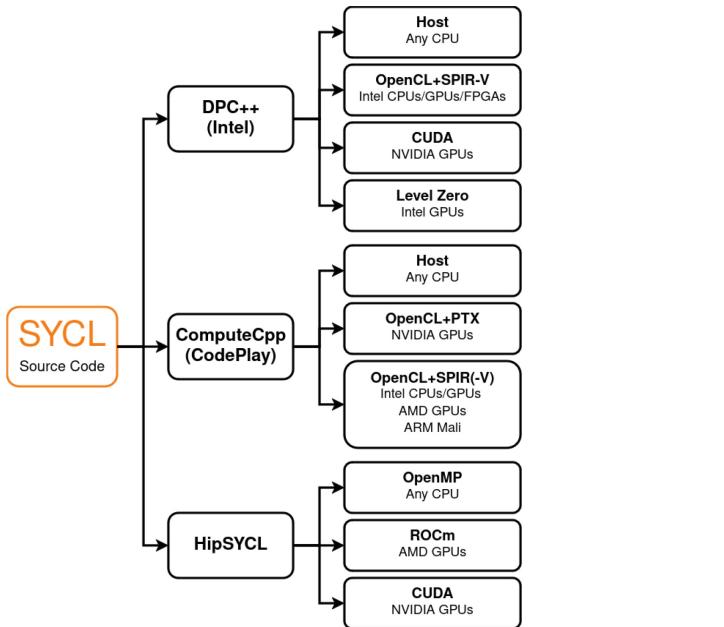


Fig. 2. Some of the SYCL implementations and their targets.

3 Implementation

For implementing the algorithm we follow the hybrid CPU/GPU storage model proposed by Pungila et al. [12]. The Aho-Corasick automaton is usually represented as a tree, where each node contains pointers to its child nodes, implemented using linked lists. While this structure is closer to the theoretical model of an automaton, it is not the most space-efficient due to the sparse pointer allocation problem, especially for large-scale DNA sequences that specific applications often deal with, including DNA biometrics and digital forensics (specifically, the file-carving process). For this reason, we are using a vector-based storage model in our implementation. The automaton is stored as a vector of nodes, where each node encapsulates not only the state information but also the transition

information in a compact format. This design choice significantly reduces the memory consumption compared to a linked-list implementation.

3.1 Node Structure

Each node in the vector contains the following information:

1. The node *bitmap* is a 4-bit array identifying children of current node, for each nucleotide. A set bit indicates the presence of a transition for that nucleotide. To simplify, as well as to help reduce branch divergence, we replace the letters ‘A’, ‘C’, ‘G’ and ‘T’ to ‘A’, ‘B’, ‘C’ and ‘D’. This replacement allows for direct transformation from a letter to a number in the set {0, 1, 2, 3}.
2. The *offset* indicates the position of the first child node in the vector. The offset and bitmap give us enough information to determine the position of all the child nodes.
3. The *failure transition* is the index in the array of nodes, to another node that represents the longest proper suffix of the current node’s path.
4. The *output link* is an index towards another node that represents a matched pattern, if exists.
5. The *pattern index* is an identifier for the specific pattern that ends at this node, if it exists.

Algorithm 1. Tree flattening algorithm

Input: trie.node pointer **node**, Node array **v[]**, integer **p** which stores the index of node on the vector, integer **nextIdx** which stores the next free space in the vector
Output: Flatten the tree into a vector **v**

```

procedure FLATTEN(node, v, p, nextIdx)
    v[p].patternEnd  $\leftarrow$  node.patternEnd
    if node.children =  $\emptyset$  then
        return
    end if
    v[p].offset  $\leftarrow$  nextIdx
    childIdx  $\leftarrow$  nextIdx
    for each c in node.children do
        v[p].bitmap  $\leftarrow$  v[p].bitmap | (1  $\ll$  (c.key – ‘A’))
        nextIdx  $\leftarrow$  nextIdx + 1
    end for
    for each c in node.children do
        FLATTEN(c.value, v, childIdx, nextIdx)
        childIdx  $\leftarrow$  childIdx + 1
    end for
end procedure
```

For compressing the tree into a vector, we use a recursive approach. For each node in the tree, starting from the root on index 0, we set the pattern which ends

in that position by copying it from the node and the offset as the next available space in the vector, then place all of its direct children in consecutive order starting from the offset position, and we set the corresponding bit in the bitmap for each of the existing children. Afterwards, we recursively call the function for each of the children, given we already know their index in the vector.

3.2 Pattern Searching on the GPU

After we transfer the vector of nodes in V-RAM, we divide the input text into sections of equal size and assign a thread to each of them. Additionally, we extend each of the sections by the maximum pattern length - 1 to avoid the boundary detection problem. Figure 3 depicts the distribution of work and the necessary section extensions for a maximum length pattern “AGC”.

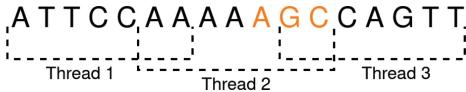
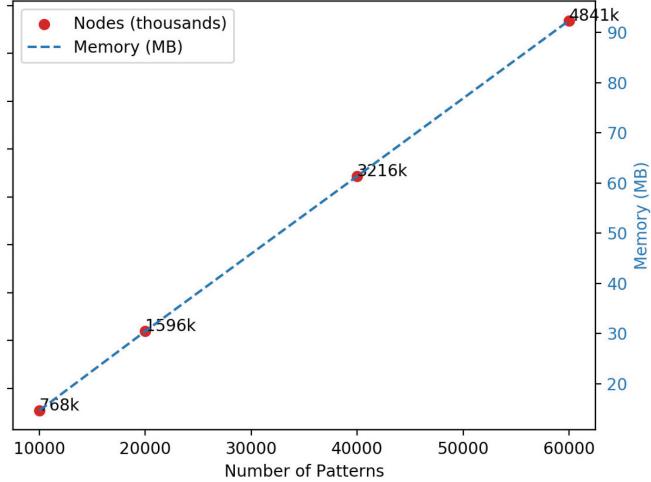
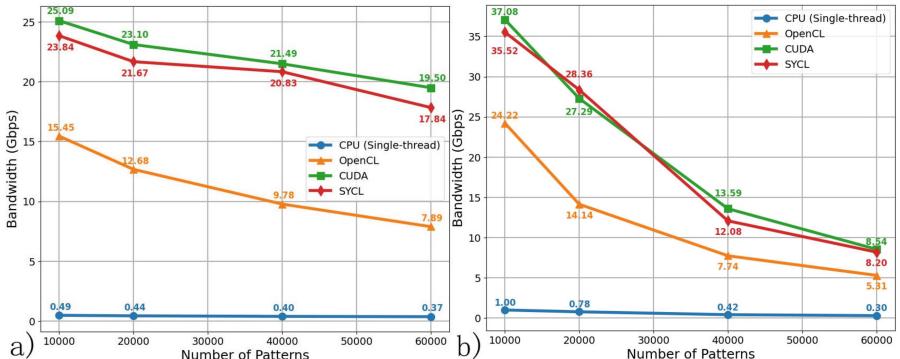


Fig. 3. Pattern search on the GPU. Example of work distribution between 3 threads in the input text “ATTCCAAAAAGCCAGTT” with the maximum length pattern to be searched - “AGC” of length 3.

The rest of the search follows the implementation of the classic Aho-Corasick algorithm, except for the way we parse the automaton. For moving from a parent node to a child node through edges, we follow the next steps: a) check whether an edge with the nucleotide from the input string is present in the automaton, the bit corresponding to the edge should be set to 1; b) create a bit-mask that has all bits beyond the bit corresponding to the nucleotide as 0, and the rest of them as 1; c) apply the *bit-wise and* operation between the bitmap of the parent and the previously computed bit mask, store this number in the bit-mask; d) compute the child index in the vector as $\text{parent.offset} + \text{popcount}(\text{bit - mask}) - 1$ (Fig. 4).

4 Experimental Results

For testing we performed two experiments: A) one using a dataset of 60,000 DNA pattern sequences with an average length of 100 characters, and B) another with 60,000 file header signatures with the same average length (for file carving in digital forensics). For experiment A, our target sequence had approximately 250M nucleotides, and for B our input dataset was an ISO image file about 650 MB in size. We implemented the approach described above with the three major frameworks discussed, alongside a single-threaded CPU implementation as baseline. We ran the experiments on machine with an NVIDIA RTX 2060 Super GPU, an AMD Ryzen 7 3700X CPU, and 32 GB of RAM memory. Our test CPU runs at a base clock speed of 3.6 GHz and boosts up to 4.4 GHz, while our GPU has

**Fig. 4.** Automaton complexity analysis.**Fig. 5.** Throughput of the Aho-Corasick algorithm optimized implementation with different frameworks, for a) experiment A and b) experiment B.

8GB of GDDR6 VRAM and runs at a base clock speed of 1470 MHz with boosts up to 1750 MHz. We measured the throughput of each implementation with 4 different datasets containing 10K, 20K, and 40K subsets of the patterns for the automaton, as well as the full patterns dataset. We optimized the parameters for each implementation separately to produce the achieve higher occupancy rate for our GPU cores. For our CUDA implementation, for example, we used 256 blocks with 128 threads per block in experiment A and 32 blocks with 256 threads for experiment B. For OpenCL, we used a global item size of 8192 and local item size of 128, and for SYCL we used a thread pool of 9600 (A) and 12,000 (B). For each setup we ran 5 separate experiments and observed the average of all the results obtained (Fig. 5).

In our experiments, we have noticed a constant throughput for the single-thread CPU implementation even when the number of patterns varies. This is somewhat to be expected however, as the CPU memory bandwidth thresholds are much easier reached due to its inherent limited scalability. For the GPU implementations, we've noticed a decrease in the throughput directly proportional with the number of patterns. Surprisingly however, the managed code handled by SYCL offers on-par performance with the native CUDA framework in both experiments ran, generating PTX code to a comparable degree. This outlines its potential for use in day-to-day applications, where cross-platform heterogeneous compatibility is key.

5 Conclusions

In this paper, we have explored the use of SYCL as a viable alternative to CUDA and OpenCL for the parallel implementation of the Aho-Corasick algorithm on GPUs for DNA sequence matching. The results suggest SYCL not only competes well in terms of performance, but also simplifies development by abstracting away much of the complexity associated with low level GPU programming. This is a promising output for computational biology as it means that programmers can concentrate more on the problems they are trying to solve and less on the mechanics of the hardware they are using because of its integration with modern C++.

Acknowledgements. This work has been partially supported by RoNaQCI, part of EuroQCI, DIGITAL-2021-QCI-01-DEPLOY-NATIONAL, 101091562.

References

1. Nishimura, T., Fukamachi, S., Shinohara, T.: Speed-up of Aho-Corasick pattern matching machines by rearranging states. In: Proceedings Eighth Symposium on String Processing and Information Retrieval, pp. 175–185 (2001)
2. Lin, C., Chang, S., Chen, Y., Sandnes, F.: Accelerating string matching using multi-threaded algorithm on GPU. In: 2010 IEEE Global Telecommunications Conference GLOBECOM (2010)
3. Lin, C., Liu, C., Chang, S., Huang, C.: Memory-efficient pattern matching architectures using perfect hashing on graphic processing units. In: 2012 Proceedings IEEE INFOCOM (2012)
4. Thambawita, D., Ragel, R., Elkaduwe, D.: An optimized parallel failure-less Aho-Corasick algorithm for DNA sequence matching. In: 2016 IEEE International Conference on Information and Automation for Sustainability (ICIAfS) (2016)
5. Aho, A., Corasick, M.: Efficient string matching: an aid to bibliographic search. Commun. ACM **18**, 333–340 (1975)
6. Unix Contributors Grep: A Unix utility that searches through text files for a string pattern (1975)
7. Haertel, M.: Gnugrep-2.0. (Usenet archive comp.sources.reviewed, 1993), vol. 3, July 1993
8. Memeti, S., Plllana, S.: Analyzing large-scale DNA sequences on multi-core architectures. In: The 18th IEEE International Conference on Computational Science and Engineering (CSE 2015) (2015)

9. Hasib, S., Motwani, M., Saxena, A.: Importance of Aho-Corasick string matching algorithm in real world applications. *Int. J. Comput. Sci. Inf. Technol.* **4**, 467–469 (2013)
10. Villa, O., Tumeo, A.: Accelerating DNA analysis applications on GPU clusters. In: IEEE 8th Symposium on Application Specific Processors (SASP), pp. 71–76 (2010)
11. Soroushnia, S., Daneshbalab, M., Plosila, J., Liljeberg, P.: Heterogeneous parallelization of Aho-Corasick algorithm. In: Saez-Rodriguez, J., Rocha, M.P., Fdez-Riverola, F., De Paz Santana, J.F. (eds.) 8th International Conference on Practical Applications of Computational Biology & Bioinformatics (PACBB 2014). AISC, vol. 294, pp. 153–160. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07581-5_19
12. Pungila, C., Galis, D., Negru, V.: Optimizing pattern-matching for memory-efficient heterogeneous DNA processing in bioinformatics. In: 2018 IEEE 12th International Symposium on Applied Computational Intelligence and Informatics (SACI), pp. 1–5 (2018)
13. Pungila, C., Negru, V.: FAST: a high-performance architecture for heterogeneous big data forensics. In: Pérez García, H., Alfonso-Cendón, J., Sánchez González, L., Quintián, H., Corchado, E. (eds.) International Joint Conference SOCO'17-CISIS'17-ICEUTE'17 León, Spain, 6–8 September 2017, Proceeding, vol. 12, pp. 618–627. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-67180-2_60
14. Pungila, C., Negru, V.: A highly-efficient memory-compression approach for GPU-accelerated virus signature matching. In: Gollmann, D., Freiling, F.C. (eds.) Information Security: 15th International Conference, ISC 2012, Passau, Germany, 19–21 September 2012, Proceedings, vol. 15, pp. 354–369. Springer, Cham (2012). https://doi.org/10.1007/978-3-642-33383-5_22
15. NVIDIA NVIDIA CUDA Compute unified device architecture programming guide (2007)
16. Reyes, R., Lomüller, V.: SYCL: single-source C++ accelerator programming. In: Parallel Computing: On the Road to Exascale, pp. 673–682 (2016)
17. Alpay, A., Soproni, B., Wünsche, H., Heuveline, V.: Exploring the possibility of a hipSYCL-based implementation of oneAPI. In: International Workshop on OpenCL, pp. 1–12 (2022)
18. Sanchez, G., González-Calderón, G.: A GPU-based evolution strategy for optic disk detection in retinal images. *Revista Ingenierías Universidad De Medellín* **15**, 173–190 (2016)
19. Suo, S., Zhu, R., Attridge, R., Wan, J.: GPU option pricing, November 2015
20. Fang, J., Varbanescu, A., Sips, H.: A comprehensive performance comparison of CUDA and OpenCL. In: 2011 International Conference on Parallel Processing, pp. 216–225 (2011)
21. Khemiri, R., Bouaafia, S., Bahba, A., Nasr, M., Sayadi, F.: Performance analysis of OpenCL and CUDA programming models for the high efficiency video coding. In: Digital Image Processing Applications (2021). <https://doi.org/10.5772/intechopen.99823>
22. The open standard for parallel programming of heterogeneous systems. <https://www.khronos.org/opencl>. Accessed 30 May 2024
23. CUDA C Programming Guide. http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA_C_Programming_Guide.pdf. Accessed 30 May 2024



Fuzzing Robotic Software Using HPC and LLM

Garnelo Del Río Francisco Borja^(✉), Francisco J. Rodríguez Lera, Camino Fernández Llamas, and Vicente Matellán Olivera

Universidad de León, Campus de Vegazana s/n, 24071 León, Spain
infbfgd01@estudiantes.unileon.es, {fjrodl,cferll,vmato}@unileon.es
<https://robotica.unileon.es/>

Abstract. This work investigates the application of fuzzing to enhance the cybersecurity of complex robotic software. Fuzzing requires substantial computational resources for automated simulation and testing. Employing generative AI methodologies in the fuzzing process, particularly in message generation and mutation, which enhances the efficacy of these tools. This paper describes a proof of concept by integrates a generalist generative Large Language Model (LLM), marcoroni, into the Robofuzz framework. This integration showcases the potential of AI in fuzzing and provides a ready-to-use deployable tool that can be refined with more targeted models. It is designed for modular utilization with Singularity container technology, facilitating large-scale deployment in High-Performance Computing (HPC) environments. All associated materials, including the proof of concept and code, are accessible on GitHub for review and utilization.

Keywords: Robotics · fuzzing · HPC · RoboFuzz · Software security · Singularity · llama_ros · ROS2 · Simulation · LLM · marcoroni · AI · GGUF

1 Introduction

Fuzz testing, also known as fuzzing, is a software testing technique that involves providing invalid, unexpected, or random data as inputs to a computer program. As probed in our previous research [9], this technique is particularly relevant in the context of ROS2 robotic systems due to their distributed and highly dynamic nature.¹

Robofuzz tool shows promising potential for improving the reliability and robustness of ROS2-based robotic systems. By subjecting software components to stress with randomly generated data, it identifies potential vulnerabilities and weaknesses in the system.[11].

Moreover, the viability of applying generative AI to create the fuzzer message corpus within a High-Performance Computing (HPC) environment has the

¹ <https://github.com/b0rh/HOUSE/>.

potential to significantly impact fuzz testing in robotics. The use of generative AI, particularly in the context of creating the fuzzer message corpus, can enhance the effectiveness and efficiency of fuzz testing techniques for ROS2 robotic systems. By leveraging generative AI, researchers and practitioners can automate the generation of diverse and complex input data to thoroughly test the robustness and resilience of ROS2-based robotic systems.

Furthermore, the integration of generative AI within an HPC environment can enable the parallelization and acceleration of the fuzz testing process, allowing for the rapid and comprehensive assessment of ROS2 robotic systems' behavior under various input scenarios. This can lead to more thorough and effective identification of potential vulnerabilities and edge cases, ultimately contributing to the overall reliability and security of ROS2-based robotic systems.

1.1 Contribution

This paper aims to present the enhancements developed for RoboFuzz, a fuzzer for robotic software, with the objective of incorporating the ability to utilize Large Language Models (LLM) in the seed generation and mutation process. The main contribution of this work is the integration of a new module, GenAI, into RoboFuzz. This fully operational module is activated via a switch in the fuzzer's execution parameters.

```
..
# Generates a random integer number
max_length = 6
max_tokens = int(max_length * 2)
description = "Generate a random integer compatible with the python3 int type"

value = int(run_prompt(f "length={max_length} description={description}", max_tokens,
→ max_length))
..
..
# Mutates the string <value> at position <pos> and save result in <new_value>
max_length = len(value)
max_tokens = int(max_length * 2)
description = f "Replace the character at position {pos} with another randomly
→ generated character in this string {value}"
new_value = str(run_prompt(f "length={max_length} description={description}",
→ max_tokens, max_length))
..
```

Listing 1. Code snippets from the genai.py module where the prompt is constructed.

Thanks to GenAI, the standard workflow can employ an Artificial Intelligence (AI) with a public model[8] or a proprietary model specifically trained

for this task. This eliminates the need to use the Mutator module for random content generation and modification. Instead, a generative AI model is used based on a set of easily modifiable prompts in natural language. Listing `reflis:codespspromptspsmutatespsgen` shows how the prompt is built in the GenAI module.

This innovative framework not only promotes the adoption of new research areas focused on AI models, but also maximizes all benefits derived from the use of artificial intelligence. This encompasses the ease of pattern definition via natural language, bypassing intricate implementations, or the semi-manual scrutiny of extensive data volumes. Moreover, for empirical validation, a general AI model and natural language descriptors were employed.

As part of the technical validation of the implementation, a proof-of-concept has been carried out with a public *Marcoroni-7B-v3* model in GGUF format for `llama.cpp`, integrated as a ROS2 node using `llama_ros`[10] package. The proof-of-concept, described in the Sect. 4, demonstrates the viability and potential of this proposal.

2 State of the Art

The *Fuzz testing*, also known as *fuzzing*, is a software testing technique that originated from a UNIX security study in the 1990s [13,14]. This technique has proven to be extremely useful in uncovering a variety of issues in software systems, including security vulnerabilities and programming errors. *Fuzzing* is performed by generating and feeding the target system a series of random, malformed, or unexpected input data. The purpose of this is to induce unanticipated responses, crashes, or unexpected behaviors in the system, thereby revealing weaknesses that can be exploited or that can lead to system instability.

Techniques of fuzzing have undergone significant advancements in recent years. Researchers have explored various approaches such as grammar-based fuzzing, fuzzing combined with symbolic execution and SMT solvers, taint-based fuzzing, and fuzzing with neural networks [4]. Current research in fuzzing has been primarily focused on generating inputs that are more likely to trigger vulnerabilities, improving code coverage, and devising new techniques, strategies, and algorithms to enhance fuzzing efficiency [5,12,18]. Most studies aim to find new input generation techniques to produce effective inputs and utilize coverage-based fuzzing for identifying vulnerabilities in real-world software applications [16,17,19].

Fuzzing techniques for ROS2 robotic systems have been an area of active research and development. Researchers have been focusing on devising new fuzzing strategies and algorithms to enhance the robustness and security of complex robotic software systems [7]. By applying *fuzz* testing to robotic software, errors related to coding, data handling, logic of robotic behavior, and architectural component initialization have been identified [7]. This approach has proven effective in uncovering vulnerabilities and weaknesses in fragile software interfaces by injecting unexpected and invalid inputs [6].

The field of research on fuzzing techniques for ROS2 robotic systems has made significant progress, focusing on enhancing the robustness and security of these systems. Notable work includes the development of “RoboFuzz: A Fuzz Testing Framework for ROS2” [11], which automates the generation and execution of fuzz tests on ROS2-based robotic systems to identify vulnerabilities and enhance system reliability.

Currently, fuzzing techniques for ROS2-based robotic systems, particularly utilizing the RoboFuzz tool, are characterized by a growing emphasis on addressing the challenges posed by the distributed and dynamic nature of ROS2 applications, as well as the increasing use of AI/ML techniques. The integration of generative AI within an HPC environment holds significant promise for enhancing the effectiveness and efficiency of fuzz testing techniques[15], ultimately contributing to the reliability and security of ROS2-based robotic systems.

3 Materials and Methods

This section describes the design of the proof-of-concept study conducted to investigate the integration of a generative AI into the fuzzer for robotic environments, which is prepared for high-performance computing (HPC) workloads. There are several components, the existing fuzzer, the Generative AI model used, etc. that are described in the following sections.

3.1 Fuzzer Tool

RoboFuzz is a specialized tool employing the fuzzing technique, explicitly designed for robotic software based on ROS2. The following Fig. 1, inspired by the diagram detailing the design and architecture in the RoboFuzz article[11].

This tool enables comprehensive automation of the testing process, including the autonomous generation and mutation of test inputs. Furthermore, RoboFuzz is distinguished by its ability to learn and continually improve through the feedback obtained from the results of the tests conducted. This allows for constant optimization of the testing process, enhancing the efficiency and effectiveness of error and vulnerability detection in robotic software.

3.2 Generative AI

The model used is *Marcoroni 7B V3-GGUF*. *Marcoroni 7B V3-GGUF* is an auto-regressive language model based on the *mistralai/Mistral-7B-v0.1* model and was trained by AIDC AI-Business. The model is specifically designed to operate with the English language. It is publicly accessible on the Hugging Face repository under the identifier *TheBloke/Marcoroni-7B-v3-GGUF*[2]. This model was developed by the Alibaba International AI Platform, also known as AIDC-ai-business[1].

The *llama_ros* package provides a suite of ROS 2 packages designed to integrate *llama.cpp* into ROS 2[10]. This suite of components allows for the seamless

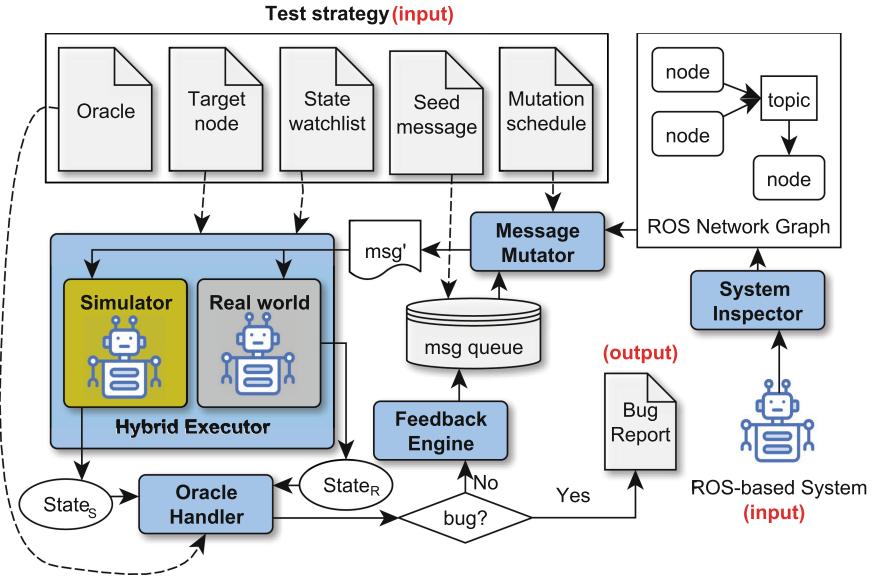


Fig. 1. Architecture and design of the operation flows of RoboFuzz.

incorporation of the powerful capabilities of *llama.cpp* into ROS2 software developments. By utilizing *llama_ros*, one can run Large Language Models (LLMs) and Very Large Language Models (VLMs) that are based on the GGUF format, such as the *Marcoroni 7B V3-GGUF* model. This integration effectively makes the capabilities of the *Marcoroni 7B V3-GGUF* model available within the ROS2 environment.

3.3 HPC Job Management

SLURM (Simple Linux Utility for Resource Management) is the job manager used in the proof of concept. It is a widely used open-source tool that serves as a workload manager and job scheduler for Linux and Unix-based clusters and supercomputers. Besides, it provides a versatile and scalable infrastructure for managing and scheduling tasks across a broad range of nodes.

3.4 Proof-of-Concept Design

RoboFuzz uses Gazebo, a 3D simulator, for the creation of virtual environments that replicate real-world scenarios. This feature enables RoboFuzz to assess the behavior of robots across a variety of situations without jeopardizing the robot's hardware.

Furthermore, RoboFuzz integrates with ROS to facilitate hybrid execution. This implies that it can acquire robotic states from both the real world and a simulator. This functionality is particularly useful for automating fuzzing, as

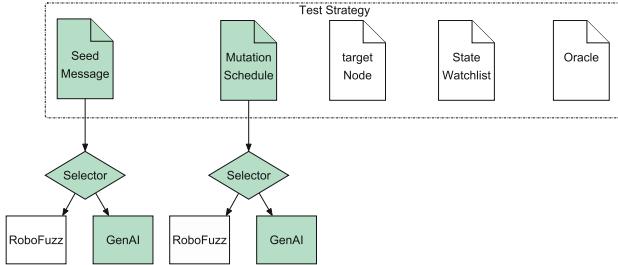


Fig. 2. The input test strategies of RobotFuzz, inclusive of the genai module features.

RoboFuzz can autonomously generate and mutate test inputs, and then observe how these inputs impact the robot’s behavior within the simulator.

For the proof of concept we have simulated the Turtlebot3[3] robot and conducted fuzzing on one of its designated topics. To enhance the support for generative Artificial Intelligence, the input management capabilities of RobotFuzz have been expanded. Now, RobotFuzz can utilize LLM or VLM models that are compatible with *llama.cpp*, through the use of *llama_ros*.

Figure 2 illustrates the enhanced components of RobotFuzz, highlighted in yellow. These improvements have been made through the GenAI module, which has been developed as part of this validation process. *llama_ros* deploys a node with the same name, using the configuration and parameterization of the model to be used.

```

You are a text string generator. Limit yourself to the requested length in characters
→ defined as an integer in the length field and to the description of the content to be
→ generated defined by the description field as text. Both fields are in the instruction
→ and their value is preceded by an = symbol.
### Instruction:
length=10 description=Fill in with the letter a
### Response:
aaaaaaaaaa
### Instruction:
length=7 description=Starts at 0 and completes with number incrementing from one to 1 until
→ the string is complete.
### Response:
0123456
### Instruction:
length=8 description=Replace the character at position 4 with another randomly generated
→ character in this string 'abcd'
### Response:
abce
### Instruction:
length=8 description=Fills with random characters
### Response:
azf2-+d6
  
```

Listing 2. alpaca.txt configuration.

The parameterization has been modified with the aim of functioning as a text generator, instead of its usual role as an assistant. This can be seen in the alpaca.txt file reflis:confpsalapaca.txt. In this document, a detailed description of the “personality” assigned to the node is provided, along with examples of valid instructions and responses that are within the framework of this configuration.

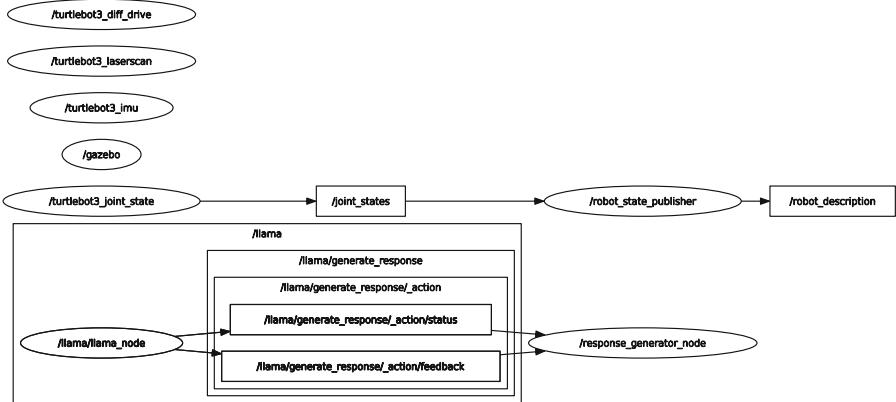


Fig. 3. rqt_graph associated all active nodes and topics during the validation process.

Figure 3 illustrates the state of the nodes and topics during the validation process. It is important to highlight the target topic, named /robot_description, as well as the operation of the *llama_ros* node.

4 Results

As detailed in Sect. 3.4, for the proof of concept, we have simulated the Turtlebot3 robot[3] and conducted a fuzzing process on one of its designated topics, /robot_description.

The operation of GenAI is detailed below, following the execution results of the generator and mutator functions.

The following figures illustrate the various stages of the flow in which the GenAI module has intervened. The detailed logs of GenAI, identified by [INFO] [TIMESTAMP] [COMPONENT]: [GENAI], are shown in the context of robofuzz execution. These logs, which include time statistics and the parameters used for input generation with GenAI, can be observed in Figs. 4 and 5.

In Fig. 4, it can be observed how GenAI is used to generate a seed randomly, represented by `5y@fB`.

Fig. 4. The operation of the GenAI module in seed generation.

In the second stage, as shown in Fig. 5, GenAI is used to mutate the previously generated seed three times, resulting in the following values: $5y@gB$, $5y#fB$ and $5y@fN$.

Finally, in Fig. 6, the monitoring of the messages from the `/robot_description` topic is shown. Here, it can be observed how the fuzzer has successfully used the generated values.

5 Discussion

The application of Artificial Intelligence (AI) is largely dependent on the quality of the data used and its orientation towards a more general use case, such as the model used, or a more specific one. This poses two opposing challenges: on the one hand, the use of general models to cover multiple purposes, which brings efficiency and simplification thanks to reuse, but at the expense of the quality of the result; on the other hand, the training of specific models for each use case, which implies a considerable effort. The main barrier here is the complexity of the task to describe using natural language. In the case of the experiment in question, the reference was a random generation system and a mutator that altered bit positions for text strings implemented in Robofuzz using Python. In more complex cases, such as network protocols or file formats where the structure is relevant to cover more test cases, it is essential to use a specific model for that task.

```

CYCLE: 0 ROUND: 1 EXEC: 1 1714129391.0297546
Current field: data (<U0)
STAGE: DETERM random
data before determ mutation: 5y@fB
bit_pos 0 ( 128 )
[INFO] [1714129391.0309567] [mutate_one]:[GENAI] Position 0 for the mutation is changed to 1.
[INFO] [1714129391.068932845] [response_generator_node]: [GENAI] Generation BEGIN prompt=|length=6 description=Replace the character at position 1 with another randomly generated character in this string ' 5y@fB' | max_token=12 max_chars=6
[INFO] [1714129391.068932845] [response_generator_node]: [GENAI] Generation END
[INFO] [1714129425.256263559] [response_generator_node]: [GENAI]Time to eval: 20.35485029228581 s
[INFO] [1714129425.256944687] [response_generator_node]: [GENAI] Prediction speed: 0.510284/2936437493 t/s
[INFO] [1714129425.261419908] [response_generator_node]: [GENAI] Generated AI result=| 5y@fB|
[executor] len(msg_list): 1
repeating 0-th sequence
(*) Starting TurtleBot3 STIL + TRD stack & Gazebo simulator
[tb3] started tb3_sitl stack 281785
[executor] started ros2 bag recording
[executor] start watching
[executor] stop watching
[executor] killing ros2 bag
[+] no error found

CYCLE: 0 ROUND: 1 EXEC: 2 1714129443.5679588
Current field: data (<U0)
STAGE: DETERM random
data before determ mutation: 5y@fB
bit_pos 8 ( 128 )
[WARNING] [1714129443.5693922] [mutate_one]:[GENAI] Out-of-length position. It is exchanged for 2 a random one within range 1 to 6.
[INFO] [1714129443.6200000] [response_generator_node]: [GENAI] Generation BEGIN prompt=|length=6 description=Replace the character at position 2 with another randomly generated character in this string ' 5y@fB' | max_token=12 max_chars=6
[INFO] [1714129443.6200000] [response_generator_node]: [GENAI] Generation END
[INFO] [1714129477.313229514] [response_generator_node]: [GENAI]Time to eval: 23.754883766174316 s
[INFO] [1714129477.317968619] [response_generator_node]: [GENAI] Prediction speed: 0.5049175452606389 t/s
[INFO] [1714129477.320124693] [response_generator_node]: [GENAI] Generated AI result=| 5y@fB|
[executor] len(msg_list): 1
repeating 0-th sequence
(*) Starting TurtleBot3 STIL + TRD stack & Gazebo simulator
[tb3] started tb3_sitl stack 281878
[executor] started ros2 bag recording
[executor] start watching
[executor] stop watching
[executor] killing ros2 bag
[+] no error found

CYCLE: 0 ROUND: 1 EXEC: 3 1714129495.1569345
Current field: data (<U0)
STAGE: DETERM random
data before determ mutation: 5y@fB
bit_pos 8 ( 128 )
[WARNING] [1714129495.1665177] [mutate_one]:[GENAI] Out-of-length position. It is exchanged for 5 a random one within range 1 to 6.
[INFO] [1714129495.202213039] [response_generator_node]: [GENAI] Generation BEGIN prompt=|length=6 description=Replace the character at position 5 with another randomly generated character in this string ' 5y@fB' | max_token=12 max_chars=6
[INFO] [1714129495.202213039] [response_generator_node]: [GENAI] Generation END
[INFO] [1714129524.347679937] [response_generator_node]: [GENAI]Time to eval: 24.44145369529724 s
[INFO] [1714129524.351964492] [response_generator_node]: [GENAI] Prediction speed: 1.0693786223403876 t/s
[INFO] [1714129524.353374497] [response_generator_node]: [GENAI] Generated AI result=| 5y@fB|
[executor] len(msg_list): 1
repeating 0-th sequence
(*) Starting TurtleBot3 STIL + TRD stack & Gazebo simulator
[tb3] started tb3_sitl stack 281976

```

Fig. 5. The operation of the GenAI module in mutation.

```

root@2f4347e537e8:/# ros2 topic echo /robot_description
INIT: 0x7f254d6b6348 0x7f254d6b925c
data: ' 5y@fB'
---
data: ' 5y@fB'
---
data: ' 5y@fN'
---

```

Fig. 6. It shows the monitoring of messages received by the /robot_description topic during the validation process.

Another aspect to discuss is that all LLM and VLM models require a series of features in the hardware for their use to be fluid. During validation, it was observed that when using relatively old hardware, such as processors from previous generations and without having graphic processors (GPU), the performance of each prompt is affected, each of them always taking times above 15 s for tasks that the default generator of Robofuzz would solve in less than a second.

Future work could explore the use of public general models^[8] like Llama or Mistral, or study performance using AI-optimized hardware.

6 Conclusions

The use of generalist generative AI represents a new and different paradigm. It offers significant advantages, such as the ability to “program” what is to be generated or how to modify it using natural language. However, this approach is not always optimal when faced with use cases that cannot be easily explained in natural language. Furthermore, all scenarios where AI is used present significant dependencies on the hardware characteristics where it runs. This means that it may not be as efficient on older systems or systems not optimized for AI use, which greatly penalizes its use due to low performance.

Acknowledgements. We gratefully acknowledge the financial support of Grant EDMAR PID2021-126592OB-C21 funded by MCIN/AEI/10.13039/501100011033 and by ERDF A way of making Europe.

References

1. Aidge — alibaba international ai platform. <https://aidc-ai.com/>. Accessed 28 April 2024
2. Thebloke/marcoroni-7b-v3-gguf. <https://huggingface.co/TheBloke/Marcoroni-7B-v3-GGUF>. Accessed 28 April 2024
3. Turtlebot3 - robotis e-manual. <https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/>. Accessed 28 April 2024
4. Bundt, J., Fasano, A., Dolan-Gavitt, B., Robertson, W., Leek, T.: Evaluating synthetic bugs (2021). <https://doi.org/10.1145/3433210.3453096>
5. Chen, Y., et al.: Savior: towards bug-driven hybrid testing (2020). <https://doi.org/10.1109/sp40000.2020.00002>
6. Cotroneo, D., Iannillo, A., Natella, R.: Evolutionary fuzzing of android os vendor system services. Empir. Softw. Eng. **24**(6), 3630–3658 (2019). <https://doi.org/10.1007/s10664-019-09725-6>
7. Delgado, R., Campusano, M., Bergel, A.: Fuzz testing in behavior-based robotics (2021). <https://doi.org/10.1109/icra48506.2021.9561259>
8. Face, H.: Hugging face models. <https://huggingface.co/models>
9. Garnelo Del Río, F.B., Rodríguez Lera, F.J., Fernández Llamas, C., Matellán Olivera, V.: Fuzzing robotic software using hpc. In: International Joint Conference CISIS 2015-ICEUTE 2015-SOCO 2015 Special Sessions, pp. 23–33. Springer (2023). https://doi.org/10.1007/978-3-031-42519-6_3
10. GitHub: llama_ros (2024). https://github.com/mgonzs13/llama_ros.git
11. Kim, S., Kim, T.: Robofuzz: Fuzzing robotic systems over robot operating system (ros) for finding correctness bugs. In: Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE 2022, pp. 447–458. Association for Computing Machinery, New York (2022). <https://doi.org/10.1145/3540250.3549164>
12. Klee, G., Ruef, A., Cooper, B., Wei, S., Hicks, M.: Evaluating fuzz testing **10**(1145/3243734), 3243804 (2018)
13. Miller, B.P., Fredriksen, L., So, B.: An empirical study of the reliability of UNIX utilities. Commun. ACM **33**(12), 32–44 (1990). <https://doi.org/10.1145/96267.96279>

14. Miller, B.P., et al.: Fuzz Revisited: A Re-examination of the Reliability of UNIX Utilities and Services, p. 23 (1995)
15. Garnelo del Río, F.B., Rodríguez Lera, F.J., Esteban Costales, G., Fernández Llamas, C., Matellán Olivera, V.: 39 house: Marco de trabajo modular de arquitectura escalable y desacoplada para el uso de técnicas de fuzzing en hpc (2021)
16. Salls, C., Machiry, A., Doupé, A., Shoshtaishvili, Y., Kruegel, C., Vigna, G.: Exploring abstraction functions in fuzzing (2020). <https://doi.org/10.1109/cns48642.2020.9162273>
17. Wang, Y., et al.: Not all coverage measurements are equal: fuzzing by coverage accounting for input prioritization (2020). <https://doi.org/10.14722/ndss.2020.24422>
18. Xu, W., Kashyap, S., Min, C., Kim, T.: Designing new operating primitives to improve fuzzing performance **10**(1145/3133956), 3134046 (2017)
19. Zhang, L., Zhao, B., Zhang, C.: Fuzzing: a survey. Cybersecurity **1**(1) (2018) <https://doi.org/10.1186/s42400-018-0002-y>



Performance Impact of Strengthening the Accountability and Explainability System in Autonomous Robots

Alejandro González-Cantón¹(✉), Miguel A. González-Santamarta¹, Enrique Soriano², Gorka Guardiola², and Francisco J. Rodriguez Lera¹

¹ Universidad de León, Campus de Vegazana, León, Spain
algonzc@unileon.es ,
fjrodl@unileon.es

² Universidad Rey Juan Carlos, Campus de Fuenlabrada, Madrid, Spain
<http://robotica.unileon.es>

Abstract. Enhancing data management and security in robotics is crucial for ensuring reliable and efficient robotic operations. Effective data management facilitates data auditing, which further improves system reliability and traceability. This paper explores the integration of SealFS, a secure file system, into autonomous robotic systems. The primary problem addressed is the potential impact of SealFS on system performance, particularly write latency, compared to traditional Ext4 file systems. The methodology involves conducting latency tests in audio, video, and navigation contexts, along with analyzing CPU usage, file size consistency, and power efficiency.

Keywords: Cybersecurity · ROS 2 · ROS integration · Secure file systems · System reliability · Traceability in robotics

1 Introduction

Autonomous robots function in dynamic and frequently uncontrolled environments, rendering them vulnerable to a range of cybersecurity threats [1]. Protecting the integrity of crucial operational logs is essential to ensure the reliability of forensic analysis [2]. Effective security measures must be implemented to mitigate these risks, safeguarding not only the operational data of the robot but also ensuring the overall reliability and safety of their functions.

The integration of off-the-self tools such as SealFS [3], a tamper-evident logging solution, becomes a valuable asset. SealFS allows the system to maintain an immutable record of its activities, detecting and preventing unauthorized alterations to crucial log files. Therefore, implementing SealFS on a robot running Robot Operating System 2 (ROS 2) [4,5] could have several positive implications:

- Security Enhancement: By ensuring the integrity of log files, the system can enhance the overall security of the robot’s operation. In a robotics context, where the system may interact with physical environments and potentially sensitive data, maintaining tamper-evident logs can be crucial for forensic analysis and identifying security breaches.
- Resource Considerations: ROS 2 systems typically run on resource-constrained devices, especially in embedded or mobile robot applications. While the described system aims for lightweight implementation, it’s essential to evaluate its resource requirements (CPU, memory, storage) to ensure compatibility with the robot’s hardware capabilities. The system’s impact on real-time performance, especially in time-sensitive robotic tasks, should also be considered.
- Integration Complexity: Integrating a Linux kernel module like SealFS into a robot running ROS 2 environment may introduce additional complexity to the system architecture. Some robot manufacturer use their own kernels for improving and guaranteeing robot performance. Besides, ROS 2 operates on top of middleware for communication between components, and integrating a file system-level security mechanism requires careful consideration of compatibility and potential conflicts with existing software stack components.

Implementing the described tamper-evident logging system on a robot with ROS 2 has the potential to enhance security and reliability, but it requires careful consideration of resource constraints, integration complexity, and physical security considerations inherent to robotic environments. Current alternatives such as the ones presented by Khasan [6] based on cryptographic file system can be also implemented as a middleware layer to encrypt individual files or directories using file system filter driver technology in the Windows kernel or a low-level file system layer, operating beneath the real file system, providing encryption for single- or multiple-disk partitions [7,8]. However, this research faces the unix-like approach [3], which is the most extended solution used by ROS developers. Thus, this paper performs an analysis and discusses the possibilities of integration given its impacts.

1.1 Contribution

Measuring the impact of SealFS on a robot involves assessing various factors related to its performance, security, and overall functionality. Given the context of using SealFS in the domain of autonomous robots, the contributions are:

- Integration and deployment in robot for the real world: these steps involve several critical steps such as adapting the system to the specific hardware and software configurations of the robots, ensuring seamless interoperability with existing components, and conducting extensive testing in real-world environments to validate performance and reliability.
- Performance evaluation: It helps determine how well the system or component performs under various conditions, ensuring that it operates efficiently and

meets the required standards [9]. The evaluation can reveal performance bottlenecks or limitations that may impede the system’s functionality, allowing for targeted improvements.

The remainder of this paper is organized as follows. Section 2 provides a detailed review of SealFS, highlighting the previous empirical findings that underpin our study and its integration in ROS middleware. Section 3 outlines the hardware and software methods employed, including the research design and data collection procedures. In Sect. 4, we present the results of our analysis, offering a comprehensive examination of the data and its implications. Section 5 discusses the findings in the context of the existing literature, addressing potential limitations and suggesting areas for future research. Finally, Sect. 6 concludes the paper by summarizing the main contributions and implications of our study.

2 Architecture

2.1 SealFS

The SealFS prototype [3] originates from the wraps framework, which is a stackable file system that allows multiple file systems to be “stacked” on top of each other, presenting a unified view to the user or other applications. SealFS has undergone progressive refinements to align with kernel version 5.19.0–43, following its initial deployment on versions 4.8.17 and 4.15.0 (referenced in the associated version tags, with substantial bug resolutions incorporated until version 5.4.0–65). Serving as a Linux kernel module, SealFS establishes itself as a stackable file system with a primary objective of authenticating written data, thereby ensuring the creation of log entries that resist tampering. Operating within the paradigm of a forward integrity model, SealFS robustly mitigates the potential for attackers to fabricate logs predating privilege elevation.

Upon SealFS deployment atop an extant file system, comprehensive protection is extended to all files positioned beneath the designated mount point. This system confines write operations exclusively to append-only, rigorously validating the integrity of data written to the underlying files managed by an alternative file system. The extant manifestation, SealFSv2, integrates the principles of ratcheting and storage-based log anti-tamper mechanisms. This innovative approach introduces a degree of adaptability, allowing users to choose between attaining a state of complete theoretical security akin to its predecessor, SealFSv1, or adopting a partially linear degradation reminiscent of classical ratchet schemes. This flexibility empowers users to finely calibrate security parameters, thereby effectively balancing security considerations with computational resource usage.

2.2 ROSSealFS

This component is a ROS 2 node dedicated to logging, designed to integrate with SealFS seamlessly. This node enables the logging of data generated during

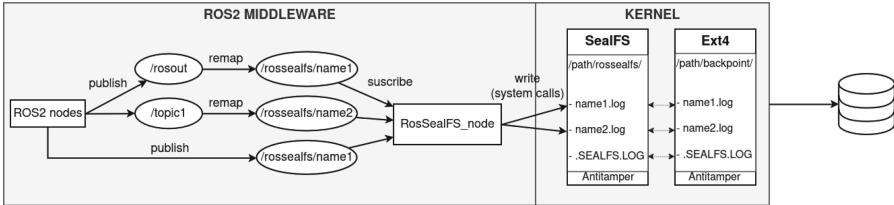


Fig. 1. Usage pipeline for SealFS and RosSealFS.

the robot's operations. When paired with SealFS configuration, it establishes tamper-evident log files, ensuring authentication for forward integrity.

In this stage, the component is supplied with data from other robot topics and general logs. The goal is to gather all information generated by regular communications in ROS 2, as well as to incorporate standard logs found in specific directories (such as *.ros*). Consequently, the files will be stored in the designated folder prepared for deploying SealFS. Figure 1 illustrates the general overview of the proposed approach.

3 Materials and Methods

While this research focuses on two specific robots used by the authors in recent years, the process can be adapted for any other robot.

3.1 Hardware Description

This research utilizes two commercially available robotic platforms, whose configurations are presented in Table 1: the Summit XL from Robotnik Automation and the TIAGo from PAL Robotics. These platforms are employed in both outdoor and indoor contexts to comprehensively evaluate their performance and versatility:

- Summit XL: It offers a versatile and customizable outdoor platform [10] for research, education and industrial applications in robotics, with its modular design, mobility, sensor suite, and compatibility with ROS. Regarding energy specifications, it features a nominal voltage of 56.5 V and a current ranging from 1.5 A at rest to 4.4 A in motion, resulting in a power consumption of 84.75 W to 248.6 W.
- TIAGo: It is a versatile and customizable indoor robot platform [11] designed to support research, education, and development in various domains of robotics. It has manipulation capabilities, mobility, perception sensors, HRI features, and ROS 2 compatibility. Concerning the battery, it boasts a nominal energy of 720 Wh, a maximum discharge of 20 A, and a nominal voltage of 19.2 V for the computer.

Table 1. Robots' configuration including the hardware and software setups.

	OS	Kernel	ROS Version	RAM	CPU	Storage
TIAGo	Ubuntu 20.04	5.10.0-rt	Noetic	16 GB	i7-7700	SATA 3.3 SSD
Summit	Ubuntu 20.04	5.15.0-generic	Noetic	32 GB	i7-10700	NVMe 1.3 SSD

3.2 Experimental Description

To conduct the experimental phase, we began by identifying the different data streams received from a robot. We focused on two message characteristics related to the ROS 2 framework: message size and frequency. Based on these observations, we proposed three data source scenarios:

- Text message streams with low and irregular frequencies and small content, such as relevant ROS 2 event logs.
- Data streams with high and regular frequency and small content, such as measurements from a laser sensor.
- Data streams with high and regular frequency and large content, such as RGB images from a camera.

The most common scenario for using SealFS in general purpose social robots would be the first scenario, with low and irregular frequencies, mainly associated to ROS 2 system logs. Thus, the remaining scenarios for using SealFS are feasible but impractical in a real-world setting due to the challenges surrounding storage size on some of the mobile robots.

The proposed methodology for practical experiments involves a systematic approach to integrating and evaluating SealFS within autonomous robots. Under these circumstances, it is possible to avoid the comparison between actuators characteristics, which would be analyzed in a future work. Three scenarios are proposed associated to three data flows that will be stored:

1. System logs will be stored during an autonomous navigation exercise using Nav2.
2. Artificial messages of intermediate size will be generated and sent at a high frequency through the middleware to simulate an auditory sensor, such as a microphone.
3. Frames captured by the robot's vision system will be stored independently.

These data flows will be stored in the file system of the robotic platform itself using SealFS file system, employing the default parameters of $nratchet = 5120$ and $size = 1000000$. For each scenario, five experiments will be conducted, each limited to a duration of 3 min, in order to obtain a sufficiently large data sample for analysis.

To measure the power consumption of the module, a different machine than the robotic platform will be utilized, as the battery does not provide a useful interface for employing this tool. To obtain a measurement as close to the actual one produced by the robot, adjustments have been made by modifying elements (such as screen, wheels, etc.) between measurements.

3.3 Metrics

To evaluate the integration of SealFS into the autonomous robots, we will analyze several key metrics.

- Throughput: Speed at which the robot can write log data to the file system with and without SealFS. This comparison will be necessary in order to ensure that SealFS does not significantly degrade performance. This value will be measured with iotop. This tool can produce the current write speeds for processes.
- Latency: Time it takes for the system to process write operations with and without SealFS. Lower latency is generally preferred. This value will be measured with ftrace [12]. With these tools, it is possible to extract the time it takes to make the right call.
- CPU Usage: Impact of SealFS on the robot’s CPU usage during normal operation and intensive logging scenarios. This value will be measured with perf. This tool outputs the CPU usage of every process and thread.
- Power Efficiency: Robot’s power consumption, especially in scenarios where the robot operates on battery power. This will be measured with Powertop [13]. This tool can output a power consumption estimation for every running process.

Table 2. Descriptive Statistics - Throughput in the proposed scenarios.

	Audio bandwidth (kB/s)				Video bandwidth (kB/s)				Navigation bandwidth (kB/s)			
	summit_ext4	summit_sealfs	TIAGo_ext4	TIAGo_sealfs	summit_ext4	summit_sealfs	TIAGo_ext4	TIAGo_sealfs	summit_ext4	summit_sealfs	TIAGo_ext4	TIAGo_sealfs
Valid	896	896	886	872	890	895	880	860	331	332	360	359
Missing	0	0	0	0	0	0	0	0	0	0	0	0
Mode	119.870	119.870	138.310	138.750	7662.150	7565.330	10670.080	10711.040	7.720	7.720	7.540	7.520
Mean	118.101	120.306	135.645	136.896	9719.360	9307.038	10691.641	10419.841	11.649	11.690	7.840	8.266
Std. Deviation	3.555	5.053	6.283	6.806	1017.225	1226.905	907.653	795.380	3.932	3.976	1.385	2.459
Minimum	93.730	91.740	122.100	120.700	7506.720	7274.770	7680.000	7772.160	3.840	3.830	3.690	3.730
Maximum	127.720	135.470	147.020	154.220	10334.540	10378.170	13137.920	13178.880	19.310	15.490	11.330	15.150
25th percentile	115.960	116.010	128.043	131.037	10102.090	7749.100	10434.561	10096.640	7.720	7.720	7.500	7.500
50th percentile	119.760	119.790	138.510	138.805	10210.390	10094.500	10670.080	10659.840	11.580	15.320	7.520	7.520
75th percentile	119.890	123.630	139.420	142.433	10274.325	10317.040	10792.960	10762.240	15.440	15.440	7.540	7.540
Sum	105818.320	107793.970	120181.610	119373.720	8.650×10^{10}	8.330×10^{10}	9.409×10^{10}	8.961×10^{10}	3855.690	3881.110	2822.520	2967.650

4 Results

Table 2 presents the analysis of throughput data across different bandwidth categories. It provides valuable insights into the performance of SealFS in comparison to Ext4 on two distinct robotic platforms: Summit and TIAGo. In the audio bandwidth category, SealFS demonstrates a slight improvement in mean throughput for both robots compared to Ext4. However, this enhancement is accompanied by an increase in variability, suggesting potential trade-offs between performance and consistency. For video bandwidth, SealFS exhibits mixed results, with lower mean throughput observed for Summit but higher

for TIAGo compared to Ext4. Notably, SealFS introduces greater variability in performance, particularly evident in Summit. In the navigation bandwidth category, SealFS generally presents marginal improvements in mean throughput for both robots, with comparable variability to Ext4. These findings underscore the nuanced impact of SealFS integration on data throughput across diverse operational contexts, highlighting the need for careful consideration of trade-offs between performance enhancements and stability in autonomous robotic systems.

Table 3 shows the latency data across the three tests, namely Audio, Video, and Navigation, provides a comprehensive insight into the performance of SealFS and Ext4 file systems in various operational contexts within autonomous robotics. In the Audio test, both Summit and TIAGo exhibit a notable increase in write latency with SealFS integration compared to Ext4. The mean write latency for Summit is $17.589\ \mu s$ with Ext4 and $71.513\ \mu s$ with SealFS, indicating a substantial increase in latency with SealFS integration. This trend is consistent across the mode, median, and 75th percentile measurements. SealFS also exhibits higher variability in write latency, with a larger standard deviation ($32.827\ \mu s$) compared to Ext4 ($8.008\ \mu s$). This increase in latency is consistent across all measured parameters, indicating a significant impact on data processing speed.

Table 3. Descriptive Statistics - Latency in Audio test

	time (μs)			
	summit_ext4_write	summit_sealfs_write	TIAGO_ext4_write	TIAGO_sealfs_write
Valid	19450	18627	17040	16496
Mode	18.166	66.664	11.115	36.643
Mean	17.589	71.513	15.277	71.506
Std. Deviation	3.982	5.430	8.008	32.827
Minimum	9.001	58.409	8.003	26.227
Maximum	45.337	149.267	277.732	293.878
25th percentile	15.418	68.346	11.135	41.079
50th percentile	17.562	70.803	13.479	74.326
75th percentile	19.543	73.463	17.543	89.801
Sum	342100.189	1.332×10^6	260311.746	1.180×10^6

Table 4 presents the Video test. SealFS integration results in a substantial rise in write latency for both robots. The variability in write latency is particularly pronounced, suggesting potential challenges in maintaining consistent performance levels, especially in video-intensive tasks.

However, the Navigation test, as shown in Table 5, presents a more nuanced picture, with SealFS demonstrating summit higher write latency compared to TIAGo. This disparity underscores the importance of evaluating the performance

Table 4. Descriptive Statistics - Latency in Video Test

	time (μ s)			
	summit_ext4_write	summit_sealfs_write	TIAGo_ext4_write	TIAGo_sealfs_write
Valid	2733	3272	3902	4240
Mode	1387.584	14853.500	1035.439	10518.330
Mean	1513.815	15520.996	1318.313	11412.904
Std. Deviation	547.953	723.789	1235.535	1106.017
Minimum	1363.485	14603.020	625.503	9701.663
Maximum	6531.803	22898.950	16741.160	20321.590
25th percentile	1404.792	15059.908	1081.766	10805.368
50th percentile	1421.612	15325.030	1146.722	11169.010
75th percentile	1451.222	15488.157	1232.809	11692.485
Sum	$4.137 \times 10^{+6}$	$5.078 \times 10^{+7}$	$5.144 \times 10^{+6}$	$4.839 \times 10^{+7}$

impact of SealFS integration across different robotic tasks, as the implications may vary depending on the specific operational requirements and hardware configurations.

Table 5. Descriptive Statistics - Latency in Navigation Test

	time (μ s)	
	summit_sealfs_write	TIAGo_sealfs_write
Valid	327	346
Missing	0	0
Mode	94.353	75.937
Mean	93.356	90.867
Std. Deviation	11.862	18.511
Minimum	72.875	67.443
Maximum	148.174	159.787
25th percentile	86.987	77.046
50th percentile	90.028	82.279
75th percentile	96.704	105.529
Sum	30527.356	31439.834

Besides it also analyzed various parameters associated with SealFS integration and impact within autonomous robotic systems. On the one hand, CPU Usage Through *perfmonitoring* showing a consistent CPU usage of 0.6% across all different scenarios, indicating stability in CPU usage. On the other hand, power efficiency (Table 6) obtaining the usage consumption of 4.025 μ s under

scenarios with high message transmission frequencies and the PC power consumption at 364.375 μW , which indicates energy efficient operation.

Table 6. Descriptive Statistics for Power Consumption

	Usage (μs) [sealfs]	Pw Estimate (μW) [sealfs]
Valid	8	8
Missing	0	0
Mode	2.400	171.000
Mean	4.025	364.375
Std. Deviation	0.870	125.518
Minimum	2.400	171.000
Maximum	5.200	501.000
25th percentile	3.700	285.000
50th percentile	4.050	396.000
75th percentile	4.575	456.250
Sum	32.200	2915.000

Upon analyzing the performance between the evaluated robots, it becomes evident that the performance of the TIAGo robot surpasses that of the Summit model. Furthermore, a significant similarity in the behaviors and trends of the file system evaluated on both platforms is observed.

5 Discussion

Based on the results obtained, several key conclusions have been identified. In environments where disk stress is low, the tool’s impact is negligible. The integration of SealFS results in a notable increase in write latency for both Summit robot and TIAGo robot compared to Ext4. SealFS introduces higher variability in write latency, potentially impacting the predictability and responsiveness of robotic operations. Thus, SealFS’s impact on latency warrants further consideration in real-time robotic applications. However, in high-load scenarios, although there is an increase in latency, this added time is measured in microseconds and does not present any significant issues for real-world deployment.

The tool does not result in any significant increase in CPU usage, suggesting that it operates efficiently without imposing additional processing burdens on the system. Additionally, there is no detectable increase in file size when using the tool, indicating that it does not add any overhead in terms of storage require-

ments. Furthermore, the tool incurs minimal additional energy consumption in percentage and nominal terms. This slight increase in energy usage is minor and unlikely to significantly impact overall system performance or efficiency.

6 Conclusions

In summary, the tool integration onboard of a robot demonstrates minimal impact on system performance and resource usage under various conditions, with some minor considerations for functionality and energy consumption. Further work could focus on addressing the limitation regarding folder creation, as the current SealsFS filesystem does not allow users to create new folders as soon it is activated. Additionally, the authors will explore optimization opportunities to reduce the minor latency and energy consumption observed, which could further enhance the tool's efficiency and user experience.

Acknowledgements. We gratefully acknowledge the financial support of Grant DMARCE Project: EDMAR PID2021-126592OB-C21 – CASCAR PID2021-126592OB-C22 funded by MCIN/AEI/10.13039/501100011033 and by ERDF A way of making Europe. Moreover, this research is part of the project TESCAC, financed by “European Union NextGeneration-EU, the Recovery Plan, Transformation and Resilience, through INCIBE”.

References

1. Lera, F.J.R., Llamas, C.F., Guerrero, Á.M., Olivera, V.M.: Cybersecurity of robotics and autonomous systems: privacy and safety. In: *Robotics-Legal, Ethical and Socioeconomic Impacts* (2017)
2. Botta, A., Rotbei, S., Zinno, S., Ventre, G.: Cyber security of robots: a comprehensive survey. *Intell. Syst. Appl.* **18**, 200237 (2023)
3. Soriano-Salvador, E., Guardiola-Múzquiz, G.: SealFS: storage-based tamper-evident logging. *Comput. Secur.* **108**, 102325 (2021)
4. Macenski, S., Foote, T., Gerkey, B., Lalancette, C., Woodall, W.: Robot operating system 2: design, architecture, and uses in the wild. *Sci. Robot.* **7**(66), eabm6074 (2022)
5. Quigley, M., et al.: ROS: an open-source robot operating system. In: *ICRA Workshop on Open Source Software*, vol. 3, p. 5, Japan, Kobe (2009)
6. Khashan, O.A., et al.: Dynamic multimedia encryption using a parallel file system based on multi-core processors. *Cryptography* **7**(1), 12 (2023)
7. Cho, C., Seong, Y., Won, Y.: Mandatory access control method for windows embedded OS security. *Electronics* **10**(20), 2478 (2021)
8. Seong, Y.S., Cho, C., Jun, Y.P., Won, Y.: Security improvement of file system filter driver in windows embedded OS. *J. Inf. Process. Syst.* **17**(4), 834–850 (2021)
9. Rodríguez-Lera, F.J., Matellán-Olivera, V., Balsa-Comerón, J., Guerrero-Higueras, Á.M., Fernández-Llamas, C.: Message encryption in robot operating system: collateral effects of hardening mobile robots. *Front. ICT* **5**, 2 (2018)
10. Arregi, M.O., Secco, E.L.: Operative guide for 4-wheel summit XL mobile robot set-up. *Acta Sci. Comput. Sci.* **5**(4), 39–47 (2023)

11. Pages, J., Marchionni, L., Ferro, F.: TIAGo: the modular robot that adapts to different research needs. In: International Workshop on Robot Modularity, IROS, vol. 290 (2016)
12. Rostedt, S.: Finding origins of latencies using Ftrace. In: Proceedings of RT Linux WS (2009)
13. Benedict, S.: Energy-aware performance analysis methodologies for HPC architectures-an exploratory study. *J. Netw. Comput. Appl.* **35**(6), 1709–1719 (2012)



Generation of Industrial Protocol-Traffic via Enhanced Wasserstein GAN

Mikel Moreno^{1,2(✉)}, Lander Segurola^{1,3}, Francesco Zola¹, Arantza del Pozo¹, and Iker Pastor²

¹ Fundación Vicomtech, Basque Research and Technology Alliance, Mikeletegi 57, 20009 San Sebastian, Spain

{mmoreno,lsegurola,fzola,adelpozo}@vicomtech.org

² Universidad de Deusto, Avda. Universidades 24, 48007 Bilbao, Spain
iker.pastor@deusto.es

³ Department of Computer Science and Artificial Intelligence, University of the Basque Country (UPV/EHU), Manuel Lardizabal pasealekua, 1, 20018 Donostia-San Sebastián, Spain

Abstract. In recent years, due to the increase in the number of smart devices connected to the internet, IoT system security has quickly become an issue of societal concern. Because of the large volume of data generated by the IoT platform, traditional protection mechanisms may be ineffective. Thus, new machine learning (ML) solutions have been introduced. However, in the cybersecurity context, both the lack of available data and the privacy issues are recurrent problems when tackling network anomaly and attack detection tasks. Recent works have shown that synthetic data obtained from generative models can help mitigate this issue for network anomaly detectors. Based on this, the following work applies the power of generative techniques in an industrial context to achieve high-quality synthetic data. The obtained results show that the synthetic and real data distributions are similar enough to be used in model training or injected as background noise in real networks.

Keywords: Synthetic Data Generation · Generative Adversarial Network · Network Traffic · Internet of Things · Unsupervised Learning

1 Introduction

Machine learning (ML) has become a useful technology in cybersecurity for anomaly and attack detection tasks [5]. To get the best performance of those ML models, a large amount of training data with specific features is required. However, training models using real user data leads to privacy exposure and ethics problems. The option of real data anonymization has been proven unsuccessful in providing satisfactory privacy protection or preventing data quality from degrading too much for machine learning model training. The only option left is the generation of new realistic synthetic data that can ensure private data can be substituted when training machine learning models [1].

Compared to generating images, learning distributions on multi-variate time-series data poses a different set of challenges. Multi-variate data is more diverse in the real world, and such data usually has more complex dependencies (temporal and spatial) as well as heterogeneous attribute types (continuous and discrete) [24].

Synthetic data generation models often treat each column as a random variable to model joint multivariate probability distributions. The modelled distribution is then used for sampling. Traditional modelling algorithms [4, 15, 22] have the restraint of distribution data types and due to computational issues, the dependability of synthetic data generated by these models is extremely limited.

GANs (Generative Adversarial Networks) have proven effective in generating network data [12, 14, 23]. However, they are either restricted to a static dependency without considering the temporal dependence usually prevalent in real-world data [17, 23], or only partially build temporal dependence inside GAN blocks [11, 12].

From a network traffic generation perspective, most of the work has been focused on IT protocols. The reality is that in the world of IoT, most devices work in an Operational Technology (OT) environment using industrial protocols, which have been less studied than their IT counterparts. This lack of research has made it challenging for security professionals to detect and prevent attacks on these systems, leaving them vulnerable to cyber threats. As the world becomes increasingly interconnected, the security of industrial systems is becoming more critical than ever before. Researchers need to continue studying and developing methods for generating synthetic network traffic for industrial protocols so that security professionals can better detect and prevent attacks on these systems. Modbus is a widely used communication protocol in industrial automation and control systems. Its applications extend to industrial automation, SCADA systems, and other contexts requiring efficient device communication. The increase in the number of available IoT devices reinforces the need for new and robust Intrusion Detection Systems (IDS). However, building these IDS requires the availability of datasets to process, train and evaluate AI models.

This work aims to prove if GAN architecture can learn the diverse types of industrial traffic instigated by users and machines on a network with multiple industrial protocols. For that, the following research questions are tackled: 1) Can GANs overcome their usual limitations and generalize and generate realistic industrial network traffic? 2) Can the generated synthetic data be injected and fool a genuine industrial service? A methodology comprised of two steps is proposed as an answer. The first is a generator module that can create synthetic network packets. The second is a simulator module that injects the synthetic data into the network to simulate a genuine Modbus client process.

2 Background

GANs have emerged as a powerful and versatile architecture in the field of machine learning, particularly for generating realistic and high-quality synthetic

data [6,8]. They employ a unique framework involving two competing neural networks, the generator and the discriminator, that are trained iteratively to optimize the generation process. Traditional GANs face several challenges during training, including mode collapse (the generator fails to capture the full diversity of the training data and instead produces limited variations), vanishing gradients, and instability [9,25].

To address these challenges, the Wasserstein GAN (WGAN) [3] was introduced as an alternative formulation of GANs. It employs a different loss function, known as the Wasserstein distance or Earth Mover's distance, which provides a more meaningful and stable gradient for training. The key idea behind this technology is to encourage the generator to produce samples that are not only realistic but also lie along the data manifold. This is achieved by measuring the Wasserstein distance between the distributions of real and generated samples. Unlike traditional GANs, WGANs do not rely on a discriminator to provide a binary classification. Instead, the discriminator is replaced by a critic, which provides a continuous score representing the quality of the generated samples.

Optimizing the Wasserstein distance offers several benefits. First, they alleviate the issue of mode collapse, allowing the generator to capture a more diverse range of data samples. Second, provides a smooth and informative gradient signal, addressing the problem of vanishing gradients encountered in traditional GANs. Finally, WGANs exhibit improved stability, making the training process more reliable and less prone to failure.

Several solutions based on GAN technology have been presented to solve the lack of availability of high-quality network datasets and the imbalance problem in network traffic classification. However, the quality of the generated traffic samples negatively affects the performance of classification models [25]. This detriment is due to a poor generalization of the trained traffic generator, which cannot correctly represent the data that is given as an input. To correct this, Shadid et al. [20] proposed adding a vanilla autoencoder pre-step to a WGAN. The autoencoder transforms the categorical data into a vector in a continuous space, after that the GAN is trained with data represented in that space and the decoder is used to reverse data back into its original form.

Some other researchers have also worked on using GANs to generate synthetic network data [16,17]. However, they focus on generating the network flow metadata (such as the number of bytes, number of packets, and flow duration) and don't generate the network packets. The work of Nukavarapu et al. [13] extends the proposed method by implementing a packet generation model to create a complete packet generation framework. It uses a sequence-based approach to model the bytes of a network packet, making it more scalable for training packets with large byte streams. Even if this approach overcomes the previous state of the art, it is limited to DNS packets and several future lines of research are pointed out. For example, the improvement of packet generation using LTSM-based models for capturing the temporal dependencies between packets in a flow and understanding the protocol syntax is marked as a future line of research. This improvement is what Hui et al. [10] present in their work, a large-scale IoT

traffic generator via a knowledge-enhanced GAN framework which introduces both the semantic knowledge (e.g., location and environment information) and the network structure knowledge for various IoT devices via a knowledge graph. Their innovation consists on a conditioning mechanism used to incorporate the knowledge and device category for IoT traffic generation and the use of LSTMs and self-attention mechanisms to capture the temporal correlation in the traffic series.

3 Proposal

In this work, a methodology based on 2 steps is proposed. The first is a generator to obtain new synthetic information from the input data. The second is a simulator to introduce the synthetic data into the network. In the next subsection, an in-depth description of the architecture is made.

On the one hand, the generator module, composed of a GAN and a data loader (a vectorizer and an encoder), is in charge of feeding the model with network traffic in pcap format.

On the other hand, a simulator module has been developed for validation purposes. In this simulation, the output content of the GAN is passed onto a Decoder and devectorizer to be injected into a network of docker containers using a packet crafting tool [18]. At the same time, a Wireshark container has been added to evaluate the number of malformed packets generated during the simulation (Fig. 1).

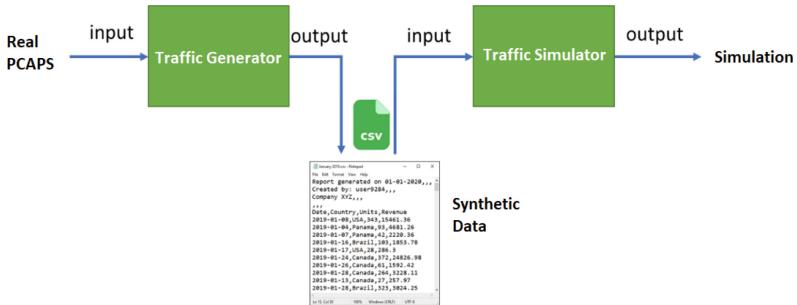


Fig. 1. Pipeline of the described solution

For each implemented protocol a separate training has been performed, so that the training process is completed with one model per protocol. Each model will be able to replicate the characteristics of the synthetic traffic to be reproduced by the injection module. It has been proved that the output of GAN models suffers from mode-collapse, a decrease in the variability by continuously generating instances of the same data. To avoid this a Wasserstein GAN (also known as WGAN) is implemented.

When training the traffic generator, the pipeline is comprised of three steps, the data acquisition, the data transformation (via an autoencoder) and the model fitting. First, the network traffic is extracted from the desired environment and the categorical features of the packets (for example source, destination or protocol fields) are parsed into vectors of continuous space via an encoder.

After that, the parsed data values are ingested in the WGAN architecture for training. Once the stop conditions are fulfilled, the model can represent new packets as continuous space vectors, to get them back to the original state, the decoder is used. The generated packets are then passed to the simulator to read them.

Once the synthetic data is generated, the packet simulator reads and introduces the values in a network of docker containers that interact with each other sending the synthetic data generated by the GAN using the packet crafting tool Scapy [18]. To verify the simulation matches the given input, the packet sniffing tool *Wireshark* is used.

In this study the open-source dataset TON_IoT dataset [2] is used; an Industry 4.0, Internet of Things (IoT) and Industrial IoT (IIoT) dataset for validating and testing various AI-based Cybersecurity applications. Specifically, a processed version of the Modbus dataset where the most important features are extracted is used. The features containing timing data will be discarded as the packet crafting tool will add its timestamp while performing the simulation. The final feature selection can be seen in Table 1.

Table 1. Features used in model training proposed by Moustafa et al. [2]

Feature	Type	Description
FC1_Read_Input_Register	Number	Modbus function code that is responsible for reading an input register
FC2_Read_Discrete_Value	Number	Modbus function code that is in charge of reading a discrete value
FC3_Read_Holding_Register	Number	Modbus function code that is responsible for reading a holding register
FC4_Read_Coil	Number	Modbus function code that is responsible for reading a coil

4 Validation Metrics

The generator, due to being trained by a second model (discriminator) instead of being trained directly, possesses no objective function that can help measure its performance or samples [19]. Still, for the quantitative validation of the GAN model, a series of tests have been performed using the metrics provided by Datacebo.Inc [7] to assure the quality of synthetic data. The performed quality

report evaluates synthetic data fidelity (how well the synthetic data captures mathematical properties from real data) by capturing the Column Shapes (CS) and Column-Pair Trends (CPT), as well as performing a data likelihood test and testing it through a classifier.

The Column Shape metric [7] indicates the similarity between real and synthetic data columns in terms of shapes (the marginal distribution or 1D histogram). This is calculated using the Kolmogorov-Smirnov (*KS*) statistic [21] and returns a value between 0 and 1. In this case, the statistic is inverted ($1 - KS$) so the higher the score the higher the quality and the more similar the distributions of real and synthetic data.

Gaussian Mixture Likelihood (GML) [7] is a data likelihood metric that calculates the likelihood of the synthetic data belonging to the real data. This metric fits multiple Gaussian mixture models to learn the distribution of the real data and produce a likelihood estimate ranging from $-\infty$ to $+\infty$, where a positive score means the synthetic data is likely part of the original set and a negative one that is not.

Tabular Detection (TD) [7] describes a set of metrics that calculate how difficult it is to tell apart the real data from the synthetic data. In this work, a Logistic Regression Classifier is assigned the task of telling apart synthetic values in a test set, which later on is evaluated using Cross Validation and returns a value between 0 and 1.

Aside from the quality report, a diagnostic has been made on the obtained synthetic data to give a general sense of the strengths and weaknesses of the model. This part involved capturing the Synthesis (is the synthetic data unique or does it copy the real rows), Coverage (does the synthetic data cover the range of possible values) and Boundaries (does the synthetic data respect the boundaries set by the real data). All of these metrics range between 0 for the minimum value and 1 for the maximum [7].

The quantitative validation of the simulation is measured using the correctness, as a measure of how many of the synthetically generated packets are legitimate Modbus network packets [13]. This is calculated by dividing the number of well-formed packets by the total number of generated packets. A well-formed packet travels from origin to destination without a problem and raises no alarms in the packet sniffers (for example Wireshark). To do that, it is necessary to build a module capable of constructing traffic from this data to verify that the synthetic traffic not only respects the distribution of the original traffic but also maintains consistency with the protocol itself (no null or contradictory values are generated that could result in malformed packets).

In conclusion, for a synthetically generated packet to be well-formed, 3 conditions have to be fulfilled. First, the generator must reflect the distribution of the original space (it has to generate something similar to the source). Second, the translation between values in the latent space and the real space is correct (the values created by the generator represent real values in the context of the simulated protocol). Third, the injection of the synthetic packets in a realistic

network does not raise alarms (the simulator successfully converts the output of the GAN into real packets).

5 Experimental Results

Table 2 shows the results obtained from the metrics described in Sect. 4. The different lines show the synthetic data generated by the same model but trained in different epochs. In the following section, an analysis of the obtained results is performed.

Table 2. Results of the proposed model

Epochs	GML	loss	TD	CS	Boundary	Synthesis
1000	-45.0088	0.5218	0.8008	0.8435	0.9537	0.9995
5000	-45.7497	0.4982	0.9389	0.9305	0.9117	0.9993
15000	-45.2039	0.4982	0.8916	0.9063	0.9387	0.9996

On the one hand, in the case of the performed quality report (involving columns 2 to 5), the 5000 epoch model outperforms the rest in Tabular Detection and Columns Shapes, indicating the data generated by this configuration is more difficult to tell apart from the original one because it has a better understanding of the shape of the columns. The 1000 epoch model in contrast has the lowest GML, this is because the data points generated by this configuration respect more the boundaries of the original distribution than the rest. Lastly, the 15000 epoch model obtains no noticeable improvement compared to the rest. It is then possible to assume that for the proposed solution, an increase in the number of epochs leads to a decrease in the general loss but little improvement in the quality of the generated synthetic data.

On the other hand, the outcomes obtained from the diagnostics report (Coverage, Boundary and Synthesis) conclude that the synthetic data covers all of the numerical ranges present in the real data (as all of them have a value of 1) and follow over 91% of the min/max boundaries set by the real data, although it is notorious a slight decrease in boundary values as the number of epochs is increased. At the same time, the synthesis value is maintained from the start (over 99%), indicating that the generated sample of the synthetic rows are not copies of the real data.

From the simulation perspective, the number of malformed packets detected by the Wireshark container has been about 10.8% from the total 8000 sent, meaning the presented solution has a correctness of 89.2%. In Fig. 2 a two dimensional plot of the real and synthetic data values visually represents how the generated distribution follows the original distribution of the data.

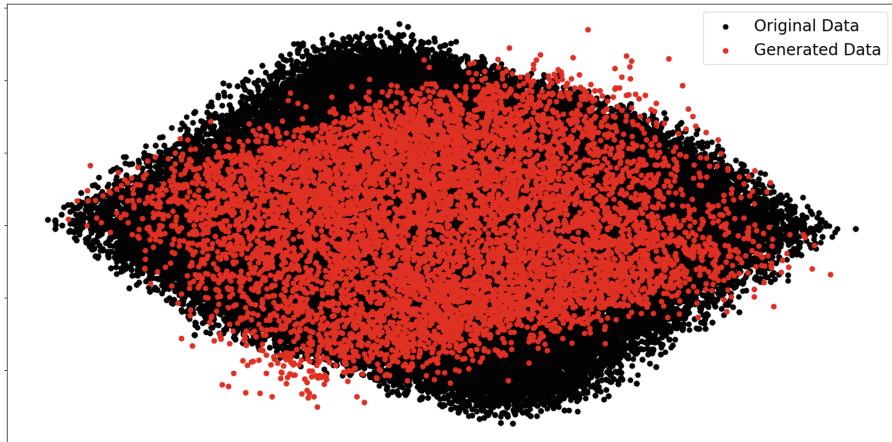


Fig. 2. 2D-Graphic representation of the original and synthetic data generated by the 1000 epoch model

6 Conclusions and Future Lines

To improve the amount and quality of open data for IoT network anomaly detection, this paper explores a novel autoencoder-enhanced Wasserstein GAN model for IoT network synthetic data generation. The proposed solution extends previously explored generative technology into novel industrial scenarios, creating a realistic representation of an OT network. The efficacy of the generator module is demonstrated through synthetic data validation and likelihood tests. Based on the results obtained it is feasible to conclude that the proposed solution can create well-formed Modbus packet content based on the given input for the simulation task. To this end, the outputs and inputs of the tools have been homogenized so that they use the same data format. Still, the generative model suffers from the problems associated with GAN models such as being prone to mode collapse, the lack of objective validation methods to compare the performance of different methods [19] and the limited use of the generated data. Future works will be focused on the use of more recent generative technology and the number of included features, this will further improve the quality and diversity of the generated content to be used in other tasks (like anomaly detection model training) which is the main limitation of the actual implementation. At the same time, the creation of a data flow for continuous learning is contemplated, to be able to capture and reproduce real-time changes in a network.

Acknowledgement. This work has been partially supported by the European Union's Horizon Program under the project KINAITICS (Grant Agreement No. 101070176) and BEACON (KK-2023/00085).

References

1. Al-Rubaie, M., Chang, J.M.: Privacy-preserving machine learning: threats and solutions. *IEEE Secur. Privacy* **17**(2), 49–58 (2019). <https://doi.org/10.1109/MSEC.2018.2888775>
2. Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., Anwar, A.: TON_IoT telemetry dataset: a new generation dataset of IoT and IIoT for data-driven intrusion detection systems. *IEEE Access* **8**, 165,130–165,150 (2020). <https://doi.org/10.1109/ACCESS.2020.3022862>
3. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: Precup, D., Teh, Y.W. (eds.) *Proceedings of the 34th International Conference on Machine Learning, Proceedings of Machine Learning Research*, vol. 70, pp. 214–223. PMLR (2017). <https://proceedings.mlr.press/v70/arjovsky17a.html>
4. Aviñó, L., Ruffini, M., Gavaldà, R.: Generating synthetic but plausible healthcare record datasets. arXiv (2018). <https://doi.org/10.48550/arXiv.1807.01514>
5. Buczak, A.L., Guven, E.: A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Commun. Surv. Tutorials* **18**(2), 1153–1176 (2016). <https://doi.org/10.1109/COMST.2015.2494502>
6. Cheng, A.: PAC-GAN: packet generation of network traffic using generative adversarial networks. In: *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference, IEMCON 2019*, pp. 728–734 (2019). <https://doi.org/10.1109/IEMCON.2019.8936224>
7. DataCebo, Inc.: Synthetic Data Metrics (2023). <https://docs.sdv.dev/sdmetrics/>. Version 0.13.0
8. Dowoo, B., Jung, Y., Choi, C.: PcapGAN: packet capture file generator by style-based generative adversarial networks. In: *Proceedings - 18th IEEE International Conference on Machine Learning and Applications, ICMLA 2019*, pp. 1149–1154 (2019). <https://doi.org/10.1109/ICMLA.2019.00191>
9. Guo, Y., Xiong, G., Li, Z., Shi, J., Cui, M., Gou, G.: TA-GAN: GAN based traffic augmentation for imbalanced network traffic classification, pp. 1–8. IEEE (2021). <https://doi.org/10.1109/IJCNN52387.2021.9533942>, <https://ieeexplore.ieee.org/document/9533942/d>
10. Hui, S., et al.: Knowledge enhanced GAN for IoT traffic generation. In: *WWW 2022 - Proceedings of the ACM Web Conference 2022*, pp. 3336–3346 (2022). <https://doi.org/10.1145/3485447.3511976>
11. Jan, S.T., et al.: Throwing darts in the dark? Detecting bots with limited data using neural data augmentation. In: *2020 IEEE Symposium on Security and Privacy (SP)*, pp. 1190–1206. IEEE (2020). <https://doi.org/10.1109/SP40000.2020.00079>
12. Lin, Z., Jain, A., Wang, C., Fanti, G., Sekar, V.: Using GANs for sharing networked time series data: challenges, initial promise, and open questions. In: *Proceedings of the ACM Internet Measurement Conference, IMC 2020*, pp. 464–483. Association for Computing Machinery, New York, NY, USA (2020). <https://doi.org/10.1145/3419394.3423643>
13. Nukavarapu, S.K., Ayyat, M., Nadeem, T.: MirageNet - towards a GAN-based framework for synthetic network traffic generation, pp. 3089–3095. IEEE (2022). <https://doi.org/10.1109/GLOBECOM48099.2022.10001494>
14. Park, N., Mohammadi, M., Gorde, K., Jajodia, S., Park, H., Kim, Y.: Data synthesis based on generative adversarial networks. *Proc. VLDB Endow.* **11**(10), 1071–1083 (2018). <https://doi.org/10.14778/3231751.3231757>

15. Procopiuc, C., Yu, T., Shen, E., Srivastava, D., Cormode, G.: Differentially private spatial decompositions, pp. 20–31. IEEE Computer Society (2012). <https://doi.org/10.1109/ICDE.2012.16>
16. Rigaki, M., García, S.: Bringing a GAN to a knife-fight: adapting malware communication to avoid detection (2018). <https://doi.org/10.1109/SPW.2018.00019>
17. Ring, M., Schröder, D., Landes, D., Hotho, A.: Flow-based network traffic generation using generative adversarial networks. Comput. Secur. **82**, 156–172 (2019). <https://doi.org/10.1016/j.cose.2018.12.012>
18. Rohith Raj, S., Rohith, R., Moharir, M., Shobha, G.: SCAPY-a powerful interactive packet manipulation program. In: 2018 International Conference on Networking, Embedded and Wireless Systems (ICNEWS), pp. 1–5 (2018). <https://doi.org/10.1109/ICNEWS.2018.8903954>
19. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training GANs. In: Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS 2016, pp. 2234–2242. Curran Associates Inc., Red Hook, NY, USA (2016)
20. Shahid, M.R., Blanc, G., Jmila, H., Zhang, Z., Debar, H.: Generative deep learning for Internet of Things network traffic generation, pp. 70–79. IEEE (2020). <https://doi.org/10.1109/PRDC50213.2020.00018>
21. Smirnov, N.: Table for estimating the goodness of fit of empirical distributions. Ann. Math. Stat. **19**(2), 279–281 (1948). <https://doi.org/10.1214/aoms/1177730256>
22. Sun, Y., Cuesta-Infante, A., Veeramachaneni, K.: Learning vine copula models for synthetic data generation. Proc. AAAI Conf. Artif. Intell. **33**(01), 5049–5057 (2019). <https://doi.org/10.1609/aaai.v33i01.33015049>
23. Xu, L., Skoulikidou, M., Cuesta-Infante, A., Veeramachaneni, K.: Modeling Tabular Data Using Conditional GAN. Curran Associates Inc., Red Hook, NY, USA (2019)
24. Xu, S., Marwah, M., Arlitt, M., Ramakrishnan, N.: STAN: synthetic network traffic generation with generative neural models. In: Wang, G., Ciptadi, A., Ahmadzadeh, A. (eds.) MLHAT 2021. CCIS, vol. 1482, pp. 3–29. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-87839-9_1
25. Zingo, P., Novocin, A.: Can GAN-generated network traffic be used to train traffic anomaly classifiers? In: 11th Annual IEEE Information Technology, Electronics and Mobile Communication Conference, IEMCON 2020, pp. 540–545 (2020). <https://doi.org/10.1109/IEMCON51383.2020.9284901>



Cyber Security in Hospitals: A Network-Oriented Model for Behavioural Learning of Employees During Phishing Simulations

Debby Bouma^{1,2}, Charlotte Hoffmans^{1,2}, Niek Jan van den Hout^{1(✉)},
Natalia Zwarts¹, Jan Treur^{1,2}, and Peter Roelofsma^{1,3}

¹ Center of Expertise Cyber Security, The Hague University of Applied Sciences,
The Hague, The Netherlands
n.j.vandenhouw@hhs.nl

² Social AI Group, Department of Computer Science, Vrije Universiteit Amsterdam,
Amsterdam, The Netherlands

³ Department of Neonatal and Pediatric Intensive Care, Division Neonatology,
Erasmus University Medical Center, Rotterdam, The Netherlands

Abstract. This paper presents a multi-level adaptive network modelling approach to analyse and predict the behaviour of employees during phishing training simulations. The presented adaptive network model is based on elements of the COM-B framework for behavioural change which takes into account factors related to the capability, opportunity and motivation to determine employees reactions to phishing emails. In addition to this the cyber security risks within hospitals associated with phishing are highlighted. The study demonstrates that through feedback learning processes the behaviour of employees changes over time leading to more secure behaviour.

Keywords: Cyber security · phishing · COM-B model · adaptive network model

1 Introduction

Due to a variety of socio-demographic factors the worldwide demand for medical care is increasing rapidly [1, 2]. Today, healthcare systems in many countries are already experiencing significant pressure [3, 4]. To manage the increased workload, healthcare organizations, including hospitals, are adopting a wide variety of information technology (IT) solutions [5, 6]. This increased dependence on information technology brings forth a wide variety of novel cyber security risks [7–9], including the unavailability of critical systems and data, the unintended modification of data and the publication or theft of confidential data.

One striking example of the materialization of these risks is the 2017 WannaCry ransomware attack on the UK National Health Service (NHS). As a

result of the attack large amounts of patient data were encrypted and made inaccessible to healthcare professionals [10]. This resulted in the cancellation of patient appointments, including emergency appointments, which consequently negatively affected the health of patients [11].

A common cyber security attack vector associated with ransomware within healthcare is the use of phishing. This can include the use of phishing emails, texts, instant messages or even regular phone calls to obtain sensitive information, including credentials.

The credentials obtained via successful phishing attempts are then used to gain access to the organizations computer network by taking over the employee user account [12].

One of the methods an organization can employ to improve employees reactions to phishing emails and initiate a learning process is through the use of phishing training simulations [13, 14].

A new way of modelling organizational learning processes and the effect of interventions such as phishing training simulations, is the application of computational network modelling [15]. By modelling the learning processes and the influence of interventions on these processes, more clarity can be achieved on the effectiveness of interventions and their effect on the overall cyber security posture of hospitals. In this paper the focus lies on presenting a specific scenario and a corresponding model that includes the influence of COM-B elements on the reporting of phishing emails by hospital employees during phishing training simulations.

To present the model first some background information is provided in Sect. 2. Then the scenario and the COM-B framework are elaborated upon in Sect. 3. Section 4 describes the basics of the self-modelling network approach in the context of organizational learning and describes the conceptual model and table which represent the scenario. This is then followed by the resulting graph in Sect. 6. Finally a discussion and conclusion with limitations is provided in Sect. 8.

2 Background Knowledge

In recent years, healthcare has changed significantly. Trends such as the increased demand for healthcare [16], increased spending [2], shortage of workers [17], advancement of personalized treatment [18] and increased provision of care outside of the hospital [19] all heavily rely on technological advancements and the increased acquisition, analysis and application of information. It is therefore not surprising that in 2021, digital investments in the healthcare sector nearly doubled to \$57 billion [19]. This increase in digital dependency also highlights the relevance of tackling cyber security risks that come with this increased dependence on information technology.

Phishing as an Attack Vector

One type of risk that is especially relevant is that of ransomware. As earlier illustrated, this type of attack can have severe negative impact on healthcare organizations. A cohort study shows 364 of these incidents at US healthcare organizations between January 2016 and December 2021 [20]. Out of these 364,

82 were reported at hospitals. Of these 82, almost 80% disrupted healthcare activities. The authors mention several operational disruptions, such as ambulance diversion, cancelled appointments and surgeries, unavailable equipment such as X-ray and CT scanners and disrupted systems, including EHR (Electronic Health Record) systems. Other studies also report increased waiting times and increased patient mortality [21].

Generally, before ransomware is activated and starts encrypting files, the attacker first has to gain access to the victim's computer network [22]. While a large variety of technological and organizational controls can be deployed to protect the network against unauthorized access, most organizations, if not all, are still vulnerable to a network breach. One of the most frequently used techniques to gain access to a computer network and bypass security controls is the use of phishing [12,23]. As earlier described phishing can be defined as "scam emails or text messages that contain links to websites which may contain malware, or trick users into revealing sensitive information (such as passwords) or transferring money" [24]. Often this sensitive information, such as credentials, is used to gain access to the victim's organizational network. A practical example of this, is an employee that receives a phishing email and that clicks on a link redirecting the employee to a malicious website. On this website the user is tricked to login with their personal username and password, sending these credentials to a server under the attacker's control. As a consequence, the attacker can now utilize these stolen credentials to access sensitive information and privileged network systems. The desired behavior in this case would be that the employee notices the phishing attempt and disregards or, even better, reports the email to the IT security department. Recent analysis proposes a requirement for a supervisory guardian role within the organization, to intervene before the phishing attempts occur [25].

In order to support and enhance employees' secure behavior, organizations can adopt a wide range of strategies and interventions, ranging from dedicated authentication hardware (such as tokens) [26] and software-based phishing detection mechanisms [27] to a holistic organizational awareness program [28]. Recent developments within the field of Machine Learning (ML) have greatly enhanced the true-positive detection rates of software-based phishing detection mechanisms [28–30]. However, due to the constantly changing characteristics of phishing attacks it is necessary to also review and enhance employee security awareness as an additional layer of defense against phishing [28,31].

An essential part of any organization's anti-phishing strategy is to take a proactive stance by educating employees to identify phishing emails and report them [28]. One intervention that educates employees is the execution of phishing training simulations during which the organization's IT security department recreates phishing scenarios by sending out artificial phishing emails to employees [32]. Instead of visiting a malicious website the employee that clicks on the link in the email is redirected to a notification page with a message informing the employee that he or she took part in a phishing simulation. The desired situation in such a scenario would be that the employee reports the phishing

email to the IT security department. Often this is facilitated by adding an extra ‘report’ button to the employee’s email client [33].

Recent experiments have shown that phishing simulations in health care institutions improve employee’s behaviour, especially when the intervention is regularly repeated [34]. The notion of required repetition is supported by established literature, with different recommendations regarding re-training intervals required to sustain the knowledge [35]. The estimation of needed and repeated sessions ranges from four times per year, up to a simulation occurring every 16 days [36]. In order to fully understand the effects of phishing training simulations on employee behavior, several behavioral analysis approaches could be used.

COM-B Framework for Behavioral Change

One of these approaches is the COM-B framework for behavioral change. This widely used framework to examine employee (security) behavior was developed in 2011 by Michie et al. and focuses on how capability (C), opportunity (O) and motivation (M) influence behavior (B) [37]. We chose this framework because it has been used frequently in the field of cyber security to understand employee (security) behavior and design effective behavioral interventions [38–41], including in the area of phishing prevention [42,43]. Because the framework takes into account all relevant internal and external elements it can serve as a foundation to build upon and create a more tailored network-oriented model. The different elements present in the COM-B framework that influence the final behavior are described in Table 1.

Table 1. The main elements in the COM-B Framework for Behavioral Change.

COM-B element	Description
Capability	The individual’s psychological and physical capacity to engage in the activity. It includes having the necessary knowledge and skills
Opportunity	All the factors that lie outside the individual that make the behavior possible or prompt it
Motivation	All those brain processes that energize and direct behavior. It includes habitual processes, emotional responding as well as analytical decision-making

The elements described in Table 1 can be divided further into several sub-elements. For the presented network-oriented model it chosen not to do so in order to facilitate the creation of a high-level network-oriented model. The final behaviour of an employee is determined by capability, opportunity and motivational factors. Consequently increasing the capability, opportunity or motivation through a specialized intervention, can influence the final behavior of the employee.

Another option to describe behaviour is the ISM model, looking into (1) the Individual, (2) the Social, and (3) the Material [44]. While both models

are influential, in this paper the COM-B model was picked because it focuses more on individual behaviour and has been used widely within the field of cyber security.

3 Scenario

As described earlier, phishing simulation training is utilized by many organizations to enhance employee behavior and prevent them from clicking on malicious links. The training works as follows: when an employee clicks on the link in the simulated phishing email, the employee is notified that they have fallen for a phishing attempt. Alternatively, the employee can report the simulated phishing email and receive a success notification. In either case, the employees undergo a learning process, which has a certain effect on the capability, motivation and opportunity factors.

To illustrate the impact of this learning process and demonstrate the effectiveness of interventions on overall cyber security posture, a scenario within a hospital environment is proposed and utilized as an example.

Consider an individual starting to work as a secretary at a hospital, receiving numerous emails daily. In an effort to improve the hospitals organizational cyber security, phishing simulation training is implemented immediately upon the individual's employment. The employee is characterized by two individual traits: capability and motivation, and by external conditions (opportunity) provided by the company, based on the COM-B model for behavioral analysis as discussed in Sects. 2 and 5.

In this scenario the individual is motivated to learn, but has limited capability and opportunity to recognize and respond properly to phishing emails. Throughout their employment, the individual encounters simulated phishing emails, as part of a training intervention, offering the options to either click on the link or report it. Feedback is provided for both actions. By doing this the hospital aims to improve the employees cyber security behavior.

4 The Self-modelling Network Approach

The novel technique to simulate effects of behavioral interventions presented in this paper is dynamic computational modelling through self-modelling networks described by Treur in [45] and [15]. The self-modelling network approach is an adaptive model network used to model scenarios and observe the effects states can have on each other within a MATLAB environment. In this paper, an adaptive model network refers to a model which consists of several states. These states can be influenced by other, and coevolve. These states are influenced by several characteristics. This approach has demonstrated its use in various examples, such as modeling shared mental models in teamwork within hospitals [46], and modeling synchronization and social bonding based on adaptive homophily [47]. The program is highly accessible due to the comprehensive descriptions provided in the referenced book [15], and the MATLAB files are available for free

download. But the use of this adaptive modeling is not limited to these areas, but can also be used to show the influence of epigenetics on mental illnesses, or the influence of leadership on an organisation. Due to the versatility in modeling diverse scenarios and its accessibility, this method was chosen to model the scenario described in this paper.

In these self-modelling temporal-causal networks, nodes (or states) X and Y exhibit activation levels over time (t) ranging between 0 and 1. Nodes function either as base-level nodes for actors or organizations executing actions, or as higher-order self-model states that describe learning processes and their control. Network characteristics defining the network model include:

- **Connectivity characteristics:** Network characteristics describing which nodes X connect to which nodes Y , and with what connection weight $\omega_{X,Y}$ indicating how strongly nodes are connected.
- **Aggregation characteristics:** Network characteristics which describe the way in which causal impacts have combined effects on node Y by assigning a combination function c_Y to Y , see Table 2 for the combination functions used in this paper.
- **Timing characteristics:** Network characteristics specifying speed factors η_Y for all nodes Y for which certain changes of node activations take place, for example used for timing of enabling or disabling certain mechanisms in the model; moreover, initial values are needed for all of the nodes.

Table 2. Combination functions

Combination function	Explanation
Alogistic	$\text{alogistic}_{\sigma,\tau}(V_1, \dots, V_k) = \left(\frac{1}{1+e^{-\sigma(V_1+\dots+V_k-\tau)}} - \frac{1}{1+e^{\sigma\tau}} \right) (1 + e^{-\sigma\tau})$
Compalogistic	$\text{compalogistic}_{\sigma,\tau}(V_1, \dots, V_k) = -\left(\left(\frac{1}{1+e^{-\sigma(V_1+\dots+V_k-\tau)}} - \frac{1}{1+e^{\sigma\tau}} \right) (1 + e^{-\sigma\tau}) \right)$
Stepmod	This function is used to control time intervals.
Stepmodopp	This function is used to control time intervals, but opposite of stepmod.

Standard Difference Equation: Based on these network characteristics, the following difference equation is used for simulation and analysis. Here the X_i are all states from which state Y has incoming connections:

$$Y(t + \Delta t) = Y(t) + \eta_Y [c_{\pi_Y, Y}(\omega_{X_1, Y} X_1(t), \dots, \omega_{X_k, Y} X_k(t)) - Y(t)] \Delta t$$

The network characteristics are specified in a standard table format called role matrices. These are input for the dedicated software environment, after which the above difference equation is automatically generated and simulated within this software environment [15].

Self-model Nodes for Adaptation: Usually, the states can be adaptive, meaning that the states and their characteristics can change over time. To model adaptivity, self-model nodes are added. For example, if a connection weight $\omega_{X,Y}$ changes over time due to learning or forgetting, then a self-model node $W_{X,Y}$ is introduced to represent the dynamic value of $\omega_{X,Y}$. During processing, the activation value of $W_{X,Y}$ is utilized for $\omega_{X,Y}$. This approach enriches causal modeling, allowing not only node activations but also the underlying causal relations to evolve over time. Self-model nodes are depicted at a higher level with a conceptual 3D structure, as shown in Fig. 1. They are integrated into the network structure by specifying connections to and from them, along with other network characteristics. The majority of first-order network adaptation principles known in the literature consider the adaptation of connection weights over time, with additional extensions for changes in structure or processing speed [15].

In the context of a computationally modelling phishing simulations, it becomes possible to show that the elements of the COM-B framework are significantly influenced by the learning process by having an adaptive layer to adjust the weights of the COM-B elements over time. This could provide insight into the effectiveness of interventions over time.

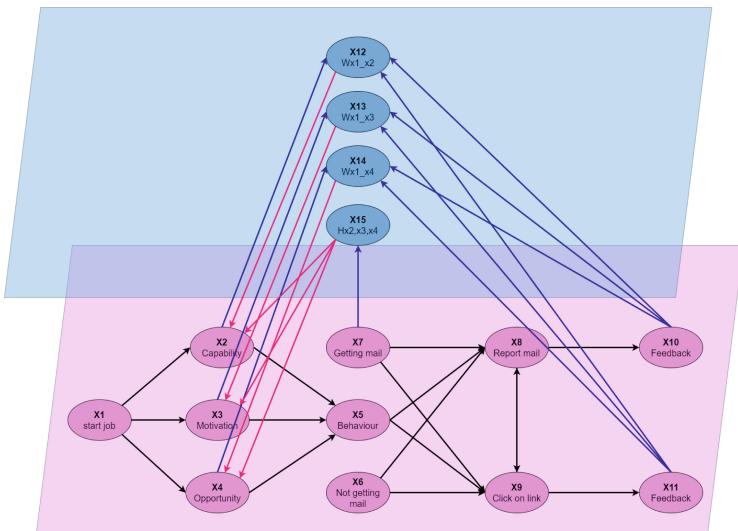


Fig. 1. Graphical representation of the phishing simulation model.

Table 3. Explanation states

State nr	State name	Explanation	Level
X1	Start job	Start of simulation, someone starts a new job	
X2	Capability	Level of employee's capability	
X3	Motivation	Level of employee's motivation	
X4	Opportunity	Level of employee's opportunity to click a link in a phishing email	
X5	Behaviour	Likeliness of the employee to behave safely	
X6	Not getting mail	Period where there is no email sent	
X7	Getting mail	Period where there is an email sent	
X8	Report mail	Action of not opening the link and sending a report to the IT department	
X9	Click on link	Action of clicking on the phishing	Base level
X10	Feedback	When reporting the phishing email, employee receives a praising email due to their safe behaviour	
X11	Feedback	In case of an employee clicking the phishing link, they receive feedback on their unsafe behaviour	
X12	\mathbf{W}_{x1_x2}	Self-model state for the weight of the connection between start job and capability, learned by feedback of X10 and X11 and the initial values of capability	
X13	\mathbf{W}_{x1_x3}	Self-model state for the weight of the connection between start job and motivation, learned by feedback of X10 and X11 and the initial values of motivation	
X14	\mathbf{W}_{x1_x4}	Self-model state for the weight of the connection between start job and opportunity, learned by feedback of X10 and X11 and the initial values of opportunity	First-order
X15	$\mathbf{H}_{x2,x3,x4}$	Self-model state for the speed of the state's capability, motivation and opportunity, when there is no email there is no learning.	self-model level

5 Adaptive Network Model

A simplified version of the COM-B framework is used to show the effects of feedback on responding to phishing emails. All fifteen states which are included in order to simulate the employee learning processes are described in Table 3.

5.1 Assumptions

To represent the previously described scenario as a model, certain simplifications and assumptions are made.

All states, except for the two email states, exist on a spectrum. This means that they range from 0 (completely off) to 1 (completely on). This can be adjusted for each state to enhance realism and versatility. The emails arrive at regular intervals, and the employee only learns when an email arrives and receives feedback.

It is assumed that the elements outlined in the COM-B framework, as applied in the context of this paper, are defined as:

- **Capability:** Does the employee have the physical and psychological capacity (such as competences, skills, and knowledge) to notice and report the simulated phishing email?
- **Opportunity:** Does the working environment (including policies, procedures, and social environment) allow and support the employee to report the simulated phishing email?
- **Motivation:** Is the employee motivated (e.g., are the beliefs, habits, and emotions supportive) to report the simulated phishing email?
- **Behaviour:** The behavior of the employee is either clicking on the simulated malicious link or reporting the phishing email by pressing the “report phishing” button in the email client.

In addition, assumptions are made about the employee in the scenario regarding motivation, capability, and opportunity. Since the employee is motivated to learn, the initial value of X13, representing the weight between the start state and motivation, starts the highest of the three **W**-states (X12, X13, and X14). The person struggles with capability; therefore, X12 has a lower initial value than X13. Lastly, X14 (learning associated with opportunity) has the lowest initial value since training within the organization has not yet begun.

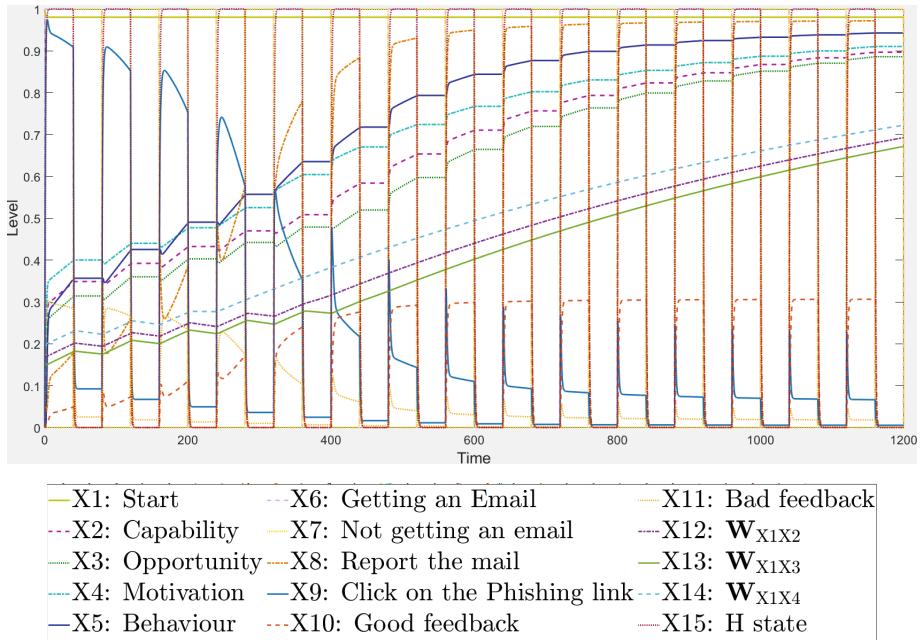
For all states, a combination function is utilized. It is assumed that the chosen function represents the states accurately.

- **Alogistic:** Used in states X1–X5, X8, and X10–X15.
- **Compalogistic:** Used in X9, since it happens opposite to when X8 occurs.
- **Stepmod:** Used in X6 to represent when an email is sent.
- **Stepmodopp:** Used in X7 to represent when there is no email sent.

There are 4 states with an initial value, the first being X1, representing the start of the simulation. Because it is nothing that can be learned, but rather something that happens or not, it has value 1. For the other states X12 (0.17), X13 (0.15), and X14 (0.20), represent the learning process. These values were relatively low, but non-zero, to show the learning of an employee over time. The values were non-zero to represent that the employee had some knowledge, but had to learn.

The evolution of states are calculated by the alogistic, and compalogistic formulas represented in Table 2. For every new time point, the value is determined by the characteristics described in Sect. 4.

6 Results



The simulation begins with X1, a constant indicating that the employee started a new job at a hospital. As this value remains constant, it will always remain 1. X2 capability, X3 opportunity, and X4 motivation are based on the COM-B framework. They all increase as a consequence of learning with the weights associated with X12, X13, and X14. Specifically, X3 exhibits the least increase due to X13 having the lowest initial value. X2 demonstrates a slightly higher increase because the initial value of X12 is higher than that of X13. Finally, X14 experiences the most significant increase, given its highest initial value compared to X12 and X13. X5, the behaviour, is influenced by X2, X3, and X4. Therefore, as these factors increase, X5 will also increase and exhibit a similar trend to that of X2, X3, and X4.

X6 indicates the periods when an email is sent, enabling learning during these occurrences. This is evident in the states of X2, X3, and X4, where an email is present (X6 equals 1), as they demonstrate an increase. Furthermore, when X6 is 1, X8 and X9 are also observed to be active. X7 is the opposite of X6, so when there is no email send, here X2, X3 and X4 stay constant, and X8 and X9 are not present.

X8, the frequency of reporting emails, increases as a result of learning. Conversely, the frequency of clicking on links X9 decreases due to learning. The graph illustrates their intersection, indicating that users have learned enough to report suspicious links more frequently than clicking on them. As the learning

progresses, the states of X12, X13, and X14 increase, signifying the learning process. X15, the H-state is 1 when an email is present and 0 when there is none, ensuring that X2, X3, and X4 remain constant in the absence of learning.

7 Conclusions

In this article the effects of a phishing training intervention was demonstrated, using a multi-level adaptive network-oriented model. The results indicate that initially, an employee struggles to recognize phishing emails and frequently clicks on links. However, consistent feedback leads to the gradual improvement of individuals' ability to identify and report suspicious emails. As the three elements-capability, motivation, and opportunities- evolve, the effectiveness of the training in promoting secure behavior through reporting simulated phishing emails becomes evident.

8 Discussion

For the purpose of modelling the described scenario several simplifications were made. For example, the influence of the employees colleagues and existing organizational security policies and guidelines on the opportunity element have not been taken into account. In addition to this, additional motivational factors, such as the use of incentives, have not been taking into account. Previous studies on the influence of the workload employees experience [48] and the nature of the phishing email itself [49] (generic vs tailor-made) could also provide insights to enhance the model and change the values of the capability, opportunity, motivation elements. A more detailed version of the COM-B framework could also be used to enhance the model. The model presented in this paper is highly suitable to be expanded upon. Furthermore, an experimental study testing the model in practice could validate the model and potentially enhance its accuracy.

References

1. Cruz-Gomes, S., Amorim-Lopes, M., Almada-Lobo, B.: The demand for health-care services and resources: patterns, trends and challenges in healthcare delivery. In: Alves, M.J., Almeida, J.P., Oliveira, J.F., Pinto, A.A. (eds.) IO 2018. SPMS, vol. 278, pp. 91–106. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-10731-4_7
2. W. E. F., World health day: 8 trends shaping global healthcare, Online Article (2023). <https://www.weforum.org/agenda/2023/04/world-health-day-healthcare-trends/>
3. W. H. O., Health and care workforce in Europe: time to act. World Health Organization. Regional Office for Europe (2022)

4. W. H. O., et al.: Future of digital health systems: report on the who symposium on the future of digital health systems in the European region (2019)
5. Alotaibi, Y.K., Federico, F.: The impact of health information technology on patient safety. *Saudi Med. J.* **38**(12), 1173–1180 (2017)
6. Kraus, S., Schiavone, F., Pluzhnikova, A., Invernizzi, A.C.: Digital transformation in healthcare: analyzing the current state-of-research. *J. Bus. Res.* **123**, 557–567 (2021). <https://www.sciencedirect.com/science/article/pii/S0148296320306913>
7. Coventry, L., Branley, D.: Cybersecurity in healthcare: a narrative review of trends, threats and ways forward. *Maturitas* **113**, 48–52 (2018). <https://www.sciencedirect.com/science/article/pii/S0378512218301658>
8. Bhuyan, S.S., et al.: Transforming healthcare cybersecurity from reactive to proactive: current status and future recommendations. *J. Med. Syst.* **44**(5), 98 (2020)
9. Zwarts, N., van den Hout, N.J.: Innovation with microgames for hospital management: using serious games to generate response plans against cyberterrorism. *Technol. Innov.* **25** (2024). <https://www.openscience.fr/Innovation-with-microgames-for-hospital-management-using-serious-games-to>
10. Martin, G., Ghafur, S., Kinross, J., Hankin, C., Darzi, A.: WannaCry—a year on. *BMJ* **361** (2018). <https://www.bmjjournals.org/content/361/bmj.k2381>
11. Ghafur, S., Kristensen, S., Honeyford, K., Martin, G., Darzi, A., Aylin, P.: A retrospective impact analysis of the WannaCry cyberattack on the NHS. *NPJ Digit. Med.* **2**(1), 98 (2019)
12. Almomani, A., Gupta, B.B., Atawneh, S., Meulenberg, A., Almomani, E.: A survey of phishing email filtering techniques. *IEEE Commun. Surv. Tutorials* **15**(4), 2070–2090 (2013)
13. McElwee, S., Murphy, G., Shelton, P.: Influencing outcomes and behaviors in simulated phishing exercises. *SoutheastCon* **2018**, 1–6 (2018)
14. Yeoh, W., Huang, H., Lee, W.-S., Jafari, F., Mansson, R.: Simulated phishing attack and embedded training campaign. *J. Comput. Inf. Syst.* **62**(4), 802–821 (2022). <https://doi.org/10.1080/08874417.2021.1919941>
15. Treur, J.: Network-Oriented Modeling for Adaptive Networks: Designing Higher-Order Adaptive Biological, Mental and Social Network Models. Springer Nature, Cham (2020). <https://doi.org/10.1007/978-3-030-31445-3>
16. W. H. Organization, Who methods and data sources for life tables 1990–2019, PDF file (2020). https://cdn.who.int/media/docs/default-source/gho-documents/global-health-estimates/ghe2019_life-table-methods.pdf?sfvrsn=c433c229_5
17. Poon, Y.-S.R., Lin, Y.P., Griffiths, P., Yong, K.K., Seah, B., Liaw, S.Y.: A global overview of healthcare workers' turnover intention amid Covid-19 pandemic: a systematic review with future directions. *Hum. Resour. Health* **20**(1), 70 (2022)
18. Hamburg, M.A., Collins, F.S.: The path to personalized medicine. *New Engl. J. Med.* **363**(4), 301–304 (2010). <https://www.nejm.org/doi/full/10.1056/NEJMOp1006304>
19. W. E. Forum, Global health and healthcare strategic outlook: shaping the future of health and healthcare, Online Article (2023). <https://www.weforum.org/publications/global-health-and-healthcare-strategic-outlook-shaping-the-future-of-health-and-healthcare/>
20. Neprash, H.T., et al.: Trends in ransomware attacks on us hospitals, clinics, and other health care delivery organizations, 2016–2021. *JAMA Health Forum* **3**(12), e224873 (2022)
21. McGlave, C.C., Neprash, H., Nikpay, S.: Hacked to pieces? The effects of ransomware attacks on hospitals and patients. The Effects of Ransomware Attacks on Hospitals and Patients, 4 October 2023

22. Brewer, R.: Ransomware attacks: detection, prevention and cure. *Netw. Secur.* **2016**(9), 5–9 (2016)
23. Alkhaili, Z., Hewage, C., Nawaf, L., Khan, I.: Phishing attacks: a recent comprehensive study and a new anatomy. *Front. Comput. Sci.* **3**, 563060 (2021)
24. NCSC, Glossary. https://www.ncsc.gov.uk/section/advice-guidance/glossary/#section_16
25. Leukfeldt, E.R.: Phishing for suitable targets in the Netherlands: routine activity theory and phishing victimization. *Cyberpsychol. Behav. Soc. Netw.* **17**(8), 551–555 (2014)
26. Alhothaily, A., Hu, C., Alrawais, A., Song, T., Cheng, X., Chen, D.: A secure and practical authentication scheme using personal devices. *IEEE Access* **5**, 11 677–11 687 (2017)
27. Dou, Z., Khalil, I., Khreichah, A., Al-Fuqaha, A., Guizani, M.: Systematization of Knowledge (SoK): a systematic review of software-based web phishing detection. *IEEE Commun. Surv. Tutorials* **19**(4), 2797–2819 (2017)
28. Jampen, D., Gür, G., Sutter, T., Tellenbach, B.: Don't click: towards an effective anti-phishing training. A comparative literature review. *Human-Centric Comput. Inf. Sci.* **10**(1), 33 (2020)
29. Atlam, H.F., Oluwatimilehin, O.: Business email compromise phishing detection based on machine learning: a systematic literature review. *Electronics* **12**(1), 42 (2023). <https://www.mdpi.com/2079-9292/12/1/42>
30. Atawneh, S., Aljehani, H.: Phishing email detection model using deep learning. *Electronics* **12**(20), 4261 (2023). <https://www.mdpi.com/2079-9292/12/20/4261>
31. Hillman, D., Harel, Y., Toch, E.: Evaluating organizational phishing awareness training on an enterprise scale. *Comput. Secur.* **132**, 103364 (2023). <https://www.sciencedirect.com/science/article/pii/S0167404823002742>
32. Teitsma, M., Sandberg, J., Maris, M., Wielinga, B.: Using an ontology to automatically generate questions for the determination of situations. In: Hameurlain, A., Liddle, S.W., Schewe, K.D., Zhou, X. (eds.) *Database and Expert Systems Applications: 22nd International Conference, DEXA: Toulouse, France, 29 August–2 September 2011, Proceedings, Part II* 22, pp. 456–463. Springer, Cham (2011). https://doi.org/10.1007/978-3-642-23091-2_39
33. Cooper, M., Levy, Y., Wang, L., Dringus, L.: Heads-up! an alert and warning system for phishing emails. *Organ. Cybersecur. J. Pract. Process People* (2021)
34. Gordon, W.J., et al.: Assessment of employee susceptibility to phishing attacks at us health care institutions. *JAMA Netw. Open* **2**(3), e190 393–e190 393 (2019)
35. Kumaraguru, P., et al.: Getting users to pay attention to anti-phishing education: evaluation of retention and transfer. In: *Proceedings of the Anti-phishing Working Groups 2nd Annual eCrime Researchers Summit*, pp. 70–81 (2007)
36. Jackson, C., Simon, D.R., Tan, D.S., Barth, A.: An evaluation of extended validation and picture-in-picture phishing attacks. In: Dietrich, S., Dhamija, R. (eds.) *Financial Cryptography and Data Security: 11th International Conference, FC: and 1st International Workshop on Usable Security, USEC 2007, Scarborough, Trinidad and Tobago, 12–16 February 2007, Revised Selected Papers* 11, pp. 281–293. Springer, Cham (2007). https://doi.org/10.1007/978-3-540-77366-5_27
37. Michie, S., Van Stralen, M.M., West, R.: The behaviour change wheel: a new method for characterising and designing behaviour change interventions. *Implement. Sci.* **6**, 1–12 (2011)
38. Joinson, A., van Steen, T.: Human aspects of cyber security: behaviour or culture change? *Cyber Secur. Peer-Review. J.* **1**(4), 351–360 (2018)

39. van der Kleij, R., Wijn, R., Hof, T.: An application and empirical test of the capability opportunity motivation-behaviour model to data leakage prevention in financial organizations. *Comput. Secur.* **97**, 101970 (2020)
40. van der Kleij, R., van't Hoff-De Goede, S., van de Weijer, S., Leukfeldt, R.: How safely do we behave online? An explanatory study into the cybersecurity behaviors of Dutch citizens. In: Schmorow, D.D., Fidopiastis, C.M. (eds.) International Conference on Applied Human Factors and Ergonomics, pp. 238–246. Springer, Cham (2021). https://doi.org/10.1007/978-3-031-35017-7_29
41. van Steen, T.: Measuring behavioural cybersecurity: an overview of options. In: Schmorow, D.D., Fidopiastis, C.M. (eds.) International Conference on Human-Computer Interaction, pp. 460–471. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-35017-7_29
42. Alshaikh, M., Naseer, H., Ahmad, A., Maynard, S.B.: Toward sustainable behaviour change: an approach for cyber security education training and awareness (2019)
43. Thamae, R., Abdullah, H., Mujinga, M.: Toward a framework to improve employees' compliance with cybersecurity policy in organizations. In: Yang, X.-S., Sherratt, R.S., Dey, N., Joshi, A. (eds.) Proceedings of Eighth International Congress on Information and Communication Technology, pp. 359–369. Springer, Cham (2024). https://doi.org/10.1007/978-981-99-3043-2_28
44. Darnton, A., Horne, J.: Influencing behaviours-moving beyond the individual: ISM user guide. Scottish Government (2013). <https://www.gov.scot/publications/influencing-behaviours-moving-beyond-individual-user-guide-ism-tool/>. Accessed 1 March 2022
45. Treur, J.: Network-Oriented Modeling. Springer, Cham (2016). <https://doi.org/10.1007/978-3-319-45213-5>
46. van Ments, L., Treur, J., Klein, J., Roelofsma, P.: A second-order adaptive network model for shared mental models in hospital teamwork. In: Nguyen, N.T., Iliadis, L., Maglogiannis, I., Trawiński, B. (eds.) Computational Collective Intelligence: 13th International Conference, ICCCI: Proceedings 13, pp. 126–140. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88081-1_10
47. Mukeria, Y., Treur, J., Hendrikse, S.: A multi-adaptive network model for human Hebbian learning, synchronization and social bonding based on adaptive homophily. *Cogn. Syst. Res.* **84**, 101187 (2024)
48. Jalali, M.S., Bruckes, M., Westmattelmann, D., Schewe, G.: Why employees (still) click on phishing links: an investigation in hospitals. *J. Med. Internet Res.* **22**(1), e16775 (2020)
49. Rizzoni, F., Magalini, S., Casaroli, A., Mari, P., Dixon, M., Coventry, L.: Phishing simulation exercise in a large hospital: a case study. *Digit. Health* **8**, 20552076221081716 (2022). pMID: 35321019. <https://doi.org/10.1177/20552076221081716>



Augmenting API Security Testing with Automated LLM-Driven Test Generation

Emil Marian Pasca^(✉), Rudolf Erdei, Daniela Delinschi, and Oliviu Matei

Technical University of Cluj Napoca, North University Centre of Baia Mare,
Cluj-Napoca, Romania
`emil.pasca@campus.utcluj.ro`

Abstract. In this paper, we propose a novel approach that leverages Large Language Models (LLMs) to automatically generate Karate DSL test cases from OpenAPI Specification documents. Our approach aims to exploit the natural language understanding and generation capabilities of LLMs to produce high-quality and comprehensive test cases that cover various security aspects. We present a proof-of-concept framework that integrates LLMs with Karate and demonstrates its feasibility and effectiveness on a sample API. We also discuss the limitations and future directions of our approach, which opens up new possibilities for applying LLMs to API security testing and other complex domains.

Keywords: Machine Learning in Software Testing · Large Language Models (LLMs) · Cybersecurity · Natural Language Understanding and Generation

1 Introduction

Application Programming Interfaces (APIs) have become fundamental components in modern software architectures, facilitating communication and data exchange between various systems. However, their ubiquity also introduces significant security risks, making API Security Testing a critical aspect of software development and maintenance. Traditional security testing methods often struggle to keep pace with the rapid evolution of APIs and the increasing sophistication of cyber threats.

This paper introduces a novel approach that leverages Large Language Models (LLMs) to automatically generate Karate DSL test cases from OpenAPI Specification documents. Our method aims to address several key challenges in API security testing:

- The time-consuming nature of manual test case creation;
- The complexity of covering a wide range of potential security vulnerabilities;
- The need for continual updates to test suites as APIs evolve.

By harnessing the natural language understanding and generation capabilities of LLMs, our approach seeks to produce high-quality, comprehensive test cases that cover various security aspects efficiently. We present **KRTLLM** a proof-of-concept framework that integrates LLMs with Karate, a popular open-source tool for API testing, demonstrating its feasibility and effectiveness.

The choice of Karate DSL as our testing framework is motivated by its intuitive syntax (see Listing 1.1), which closely resembles natural language, making it an ideal candidate for LLM-generated test cases. Additionally, Karate's built-in support for API testing and its ability to handle complex scenarios make it well-suited for security-focused test generation.

1.1 Research Questions

Our research is guided by three questions focusing on the effectiveness, limitations, and future applications of LLMs in API security testing:

- **RQ1:** *What is the effectiveness of LLMs in generating Karate DSL test scenarios for API security vulnerabilities?* We assess LLMs' ability to interpret API specifications and generate relevant test cases across security vulnerabilities;
- **RQ2:** *What are the main limitations of using LLMs for automated test generation, and how can they be addressed?* This examines the challenges and potential enhancements for the LLM methodology in test case generation;
- **RQ3:** *How can LLM integration in API security testing be extended to broader cybersecurity domains?* We explore the potential for LLMs to innovate beyond API security testing into other cybersecurity areas.

2 Literature Review

API security has evolved significantly, moving from static approaches to dynamic, adaptive strategies. Sun et al. [15] introduced comprehensive asset discovery and security audits, while Maheswari et al. [8] and Hussain et al. [7] developed personalized security APIs and intelligent service mesh frameworks, respectively. These innovations reflect the trend of combining traditional and novel approaches for enhanced API security management.

As APIs grew more complex, the need for sophisticated automated test generation became apparent. Early attempts, however, faced challenges in addressing Inter-Parameter Dependency (IPD), as highlighted by Ed-doubi et al. [5] and Banias et al. [3]. Responding to these challenges, Martin-Lopez et al. developed the RESTest framework [9], employing fuzzing and constraint-based testing techniques.

The integration of machine learning marked a significant advancement in API security testing. Bahrami and Chen [2] demonstrated progress in automatic API specification generation, while Mirabella et al. [10] applied deep learning for request validation, enhancing testing accuracy and efficiency.

```

Feature: "Test API for SQL Injection vulnerability on User Login"
  Scenario: "Attempt an SQL injection attack on the user login API"

    Given path '/user/login'
    And param username = 'admin'
    And param password = '1\\' OR \\ '1\\'=\\'1'
    When method get
    Then status 200
    And match response != 'Login successful'

```

Listing 1.1. Karate DSL - SQL Injection Test Scenario

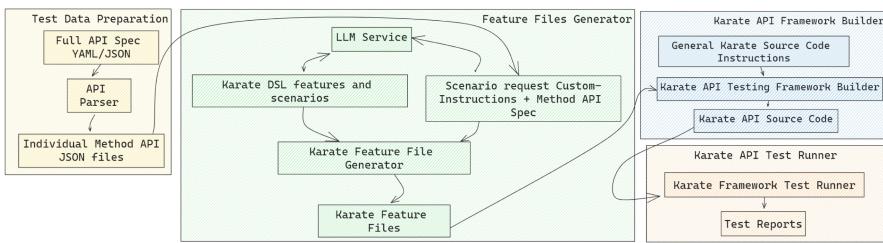


Fig. 1. Karate DSL LLM Assisted Test Case Generator (KRTLLM) architecture.

LLMs have revolutionized software engineering, including testing. Hou et al. [6] reveal that LLMs are predominantly applied to software development and maintenance tasks, with code generation being a primary focus. The evolution of LLM architectures from encoder-only to decoder-only models has dramatically increased their applicability in software engineering research.

In software testing, Schäfer et al. [13] introduced TESTPILOT, an approach for automated unit test generation using LLMs without additional training or manual effort. TESTPILOT achieved impressive results.

Recent developments have focused on refining prompt engineering techniques, with few-shot and zero-shot prompting being most common. Chat-based technologies are also emerging as valuable tools in software testing, aiding in code quality, and facilitating education [4,11,12,16].

The increasing use of open-source datasets in LLM research for software engineering supports reproducibility and generalizability. As the field evolves, we anticipate further innovations in LLM-driven software testing, potentially transforming software quality assurance by addressing current limitations and expanding the range of automated testing tasks.

3 Methodology

3.1 KRTLLM

The framework's architecture, depicted in Fig. 1, is a structured pipeline comprising four integral components.

- **Test Data Preparation:** The initial step involves an ‘API Parser’ script, designed to deconstruct the API’s JSON/YAML file into manageable segments. This granular breakdown facilitates the subsequent processing by the LLM service, accommodating its payload limitations;
- **Feature File Generator:** This module processes each API segment, combining it with predefined instruction prompts to formulate requests for the LLM Service. The response from the service, encompassing the proposed test scenarios, is then formatted into Karate DSL feature files. GPT4ALL¹, was chosen as a LLM Wrapper for running a quantised version of Mistral 7B Model [1], as it proved to be more adept at handling our requests, in comparison with LLM Studio, where we had various issues when escaping special symbols found in API Specifications;
- **Karate API Framework Builder:** Following the feature file generation, this component assembles the necessary Java files, POM.xml, and other pertinent elements, positioning the generated feature files for execution;
- **Karate API Test Runner:** Although not a standalone component, this stage involves the execution of the ‘mvn test’ command by the user or an automated scheduler, culminating in the generation of test reports.

3.2 Evaluation Metrics

Definition 1 (Accuracy). *The Accuracy evaluates how a generated test case reflects the given API specification across multiple dimensions. Unlike a simple True Positive measure, this metric assesses the structural and semantic alignment of the test with the API’s defined characteristics. It is calculated on a scale of 1 to 5 using the following equation:*

$$A = 1 + 4 \cdot \sum_{i=1}^5 w_i \cdot f_i \quad (1)$$

where:

- A is the Accuracy Score;
- f_1 : Endpoint Accuracy - Correctness of the API endpoint URL in the test case;
- f_2 : Method Conformity - Alignment of the HTTP method with the API specification;
- f_3 : Parameter Fidelity - Inclusion and correctness of all specified parameters;

¹ <https://gpt4all.io/index.html>.

- f_4 : Payload Structure Adherence - For methods with a body, conformity to the specified payload structure;
- f_5 : Response Expectation Alignment - Proper representation of expected response codes and data structures as per the API specification;
- w_i : Weight assigned to each factor (where $\sum_{i=1}^5 w_i = 1$);

Each factor f_i is scored on a scale from 0 to 1, where 0 indicates complete misalignment and 1 indicates perfect alignment with the API specification for that aspect.

Note: This score does not measure the test's execution results or its ability to detect actual API behavior. Instead, it quantifies how well the generated test case structurally and semantically matches the provided API specification document as well as in the context of security testing.

Definition 2 (Interoperability). The Interoperability Score quantifies the degree of manual intervention required to make a generated test case compile and run correctly in the target testing framework/environment. It is calculated on a scale of 1 to 5, where 1 indicates extensive manual intervention and 5 indicates no intervention needed.

$$I = 5 - \sum_{i=1}^4 p_i \cdot w_i \quad (2)$$

where:

- I is the Interoperability Score;
- p_i are penalty factors for different types of manual interventions:
 - p_1 : Syntax corrections (0–1);
 - p_2 : Framework-specific adjustments (0–1);
 - p_3 : Data formatting changes (0–1);
 - p_4 : Assertion modifications (0–1);
- w_i are weights assigned to each penalty factor, where $\sum_{i=1}^4 w_i = 1$.

The process to compute the Interoperability Score involves:

1. Attempt to compile and run the generated test case in the target environment;
2. For each type of manual intervention required, assign a penalty factor p_i between 0 and 1, where 0 means no intervention needed and 1 means significant intervention needed;
3. Calculate the weighted sum of penalties and subtract from 5 to get the final score.

A higher Interoperability Score indicates better compatibility with the target testing framework and less need for manual adjustments.

Definition 3 (Complexity). *The Test Complexity Score quantifies the intricacy and comprehensiveness of a generated test case. It is calculated on a scale of 1 to 5, where 1 indicates a very simple test and 5 indicates a highly complex and comprehensive test.*

$$C = 1 + \sum_{i=1}^4 c_i \cdot w_i \quad (3)$$

where:

- C is the Test Complexity Score;
- c_i are complexity factors for different aspects of the test:
 - c_1 : HTTP Method Complexity (0–1);
 - c_2 : Parameter and Payload Complexity (0–1);
 - c_3 : Assertion Complexity (0–1);
 - c_4 : Scenario Coverage (0–1);
- w_i are weights assigned to each complexity factor, where $\sum_{i=1}^4 w_i = 1$.

The process to compute the Test Complexity Score involves:

1. Evaluate each complexity factor c_i ;
2. Calculate the weighted sum of complexity factors;
3. Add 1 to the result to ensure the score falls between 1 and 5.

A higher Test Complexity Score indicates a more sophisticated and comprehensive test case.

3.3 Comparative Analysis of Test Generation Across Various APIs

Our experimentation includes case studies of the proposed test generation framework in a controlled local environment across popular APIs such as VAmPI², PetStore³, RestCountries⁴, and Vapi⁵. The objective was to gauge the performance of our framework across a spectrum of API types, encompassing diverse business contexts and technical challenges. These included variations in payloads, HTTP methods, and response types. The results of these experiments are presented in Table 1, where we have also presented an overall “Total score” for each API, as for the evaluation metrics, they are defined in Sect. 3.2.

VAmPi: For our initial study, we selected VAmPI, a widely recognized vulnerable web application designed for training purposes. VAmPI serves as a **Self-Hostable API** that emulates various scenarios from the OWASP API Top 10, offering a diverse array of exercises for security testing. Our framework successfully generated 26 features and 36 test cases within 43 min. Upon execution, only 3 scenarios were initially successful. This highlighted a current limitation

² <https://github.com/erev0s/VAmPI>.

³ <https://petstore.swagger.io/>.

⁴ <https://gitlab.com/restcountries/restcountries>.

⁵ <https://github.com/roottusk/vapi>.

of our tool: it does not yet generate feature files and scenarios ready for execution without manual intervention. We encountered various compilation issues, including incorrect syntax usage and inaccurate JSON payload structures. After rectifying these issues and enhancing readability, we re-executed the tests. While the number of successful scenarios increased, it underscored the need for further refinement in test case generation, particularly in terms of parameterization and assertions.

PetStore: The test suite for PetStore was completed in about 73 min, generating 40 test cases. It demonstrated extensive path, method, and endpoint coverage. Similar to the VAmPi API, the interoperability was average, and accuracy was below average, requiring several adjustments in path and body parameters. The complexity metric showed promising results, with the framework generating comprehensive payloads incorporating a mix of body and path parameters.

RestCountries: The test suite for RestCountries was developed in 20 min, comprising 23 test cases. The performance on this API specification was moderate, with satisfactory interoperability and minimal requirement for test modifications. Despite its average accuracy, the complexity evaluation could not exceed a score of 3 due to the exclusive use of GET methods across endpoints. This API may not have been the most suitable for a comprehensive evaluation of our testing framework.

Vapi: Completed in 18 min, the Vapi test suite consisted of 19 cases. This API proved to be an adequate test subject, with the framework displaying average performance in interoperability. However, accuracy and complexity scores were lower, primarily due to the simplistic nature of the generated tests, which often resembled boilerplate scenarios.

4 Results and Discussion

Our analysis across multiple APIs revealed both significant achievements and key limitations of our current framework. The KRTLLM framework demonstrated comprehensive path coverage and efficiently generated diverse test cases, showcasing its potential to enhance traditional software testing metrics [14].

4.1 Achievements and Potential

By incorporating non-functional testing aspects, particularly security evaluations, our approach offers a more holistic assessment of API security beyond mere functional tests. The integration of LLMs, specifically using GPT4ALL with a quantized version of the Mistral 7B Model, proved effective for our initial study.

4.2 Challenges and Limitations

Despite these achievements, we encountered several challenges:

- **Interoperability Issues:** We faced various interoperability challenges requiring manual intervention, ranging from syntax errors to inaccurate JSON payload structures;
- **Manual Adjustments:** The need for manual adjustments varied across APIs, reflected in our interoperability scores ranging from average to below average;
- **Accuracy Limitations:** While capable of generating relevant test cases, our framework often fell short in capturing all nuances of complex API specifications;
- **Varying Complexity:** The complexity of generated tests varied, with simpler APIs often resulting in more basic, boilerplate-like scenarios.

4.3 Future Enhancements

These findings underscore the need for future enhancements:

- Refining prompts for the LLM service;
- Implementing additional post-processing steps to improve test case quality and usability;
- Optimizing test case generation time through more powerful computational resources or specialized LLMs;
- Conducting comprehensive user studies to refine the framework’s usability;
- Exploring various LLM architectures and training datasets to improve test case quality and effectiveness (Table 1).

Table 1. Reviewed API Test Generated Results.

API Name	TestCases	Acc.	Inter.	Co.	Total
VAmPI	36	3	2	3	71.79%
PetStore	40	3	2	4	60.00%
RestCountries	23	2	3	3	46.66%
Vapi	19	2	3	1	40.00%

TestCases = Total Number of generated test cases, Acc. = Accuracy, Inter. = Interoperability, Com. = Complexity, Total = Overall Total Score.

5 Conclusions

This work introduces a groundbreaking approach by integrating LLMs with Karate to advance API security testing. Our framework demonstrates significant improvements in test coverage and efficiency, evaluating its performance across various APIs like VAmPI, PetStore, Vapi and RestCountries.

Key Findings and Limitations: Despite its successes, challenges in interoperability and security testing depth were identified. Future work will focus on enhancing the LLM training dataset with a broader array of API descriptions and vulnerabilities to address these limitations.

Future Directions: We aim to refine the framework for higher autonomy and specialized focus on critical API vulnerabilities, such as Broken Object-level Authorization (BOLA). Additionally, exploring the use of communicative AI agents suggests a promising avenue for automating security testing further. These efforts align with the intention to create a more efficient, effective, and adaptive API security testing landscape.

5.1 Answers to Research Questions

- **RQ1:** Demonstrated the LLM’s capability to generate comprehensive Karate DSL test scenarios, significantly influenced by the diversity and specificity of the training data;
- **RQ2:** Identified interoperability and coverage depth as primary limitations, suggesting dataset enrichment and methodological improvements;
- **RQ3:** Highlighted the potential for expanding LLM application to broader cybersecurity areas, indicating a transformative impact on cybersecurity practices.

Overall, our research underscores the potential of LLMs in enhancing API security testing and opens up prospects for their application in wider cybersecurity contexts, setting the stage for future innovations in the field.

Acknowledgment. This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CCCDI - UEFISCDI, project number EUROSTARS-4-E!4691-GenDeg 1/2024, within PNCDI IV.

This work is also supported by the project “Collaborative Framework for Smart Agriculture” - COSA that received funding from Romania’s National Recovery and Resilience Plan PNRR-III-C9-2022-I8, under grant agreement 760070.

Data Availability. All our data, source code and documentation to reproduce our results are available at <https://github.com/EmilPasca/krtllm.git>.

References

1. TheBloke/Mistral-7B-OpenOrca-GGUF · Hugging Face, July 2024. Accessed 17 July 2024
2. Bahrami, M., Chen, W.-P.: Automated web service specification generation through a transformation-based learning. In: IEEE International Conference on Services Computing (2020)
3. Banias, O., Florea, D., Gyalai, R., Curiac, D.-I.: Automated specification-based testing of REST APIs. *Sensors* **21**(16), 5375 (2021)
4. Cao, J., Li, M., Wen, M., Cheung, S.C.: A study on prompt design, advantages and limitations of ChatGPT for deep learning program repair. [arXiv: abs/2304.08191](https://arxiv.org/abs/2304.08191) (2023)
5. Ed-douibi, H., Luis Cánovas Izquierdo, J., Cabot, J.: Automatic generation of test cases for rest APIs: a specification-based approach. In: 2018 IEEE 22nd International Enterprise Distributed Object Computing Conference (EDOC), pp. 181–190 (2018)
6. Hou, X., et al.: Large language models for software engineering. A systematic literature review (2024)
7. Hussain, F., Li, W.O., Noye, B., Sharieh, S., Ferworn, A.: Intelligent service mesh framework for API security and management. In: 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), pp. 0735–0742 (2019)
8. Maheswari, R., Rani, S.S., Sharmila, P., Rajarao, S.: Personalized secured API for application developer. In: Smart Innovations in Communication and Computational Sciences (2018)
9. Martin-Lopez, A., Segura, S., Ruiz-Cortés, A.: RESTest: automated black-box testing of RESTful web APIs. In: Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis, ISSTA 2021. Association for Computing Machinery (2021)
10. Mirabella, A.G., Martin-Lopez, A., Segura, S., Valencia-Cabrera, L., Ruiz-Cortés, A.: Deep learning-based prediction of test input validity for restful APIs. In: 2021 IEEE/ACM Third International Workshop on Deep Learning for Testing and Testing for Deep Learning (DeepTest), pp. 9–16 (2021)
11. Okanovic, D., et al.: Can a chatbot support software engineers with load testing? Approach and experiences. In: Proceedings of the ACM/SPEC International Conference on Performance Engineering (2020)
12. Paschoal, L.N., Melo, S.M., de Oliveira Neves, V., Conte, T.U., do Rocio Senger de Souza, S.: An experimental study on a conversational agent in software testing lessons. *Inform. Educ.* **22**, 99–120 (2022)
13. Schäfer, M., Nadi, S., Eghbali, A., Tip, F.: An empirical evaluation of using large language models for automated unit test generation (2023)
14. Sinaga, A.M., Hutajulu, O.D., Hutahaean, R.T., Hutagaol, I.C.: Path coverage information for adaptive random testing. In: Proceedings of the 2017 International Conference on Information Technology, ICIT 2017, pp. 248–252, New York, NY, USA. Association for Computing Machinery (2017)
15. Sun, R., Wang, Q., Guo, L.: Research towards key issues of API security. In: CNCERT (2021)
16. White, J., Hays, S., Fu, Q., Spencer-Smith, J., Schmidt, D.C.: ChatGPT prompt patterns for improving code quality, refactoring, requirements elicitation, and software design. [arXiv: abs/2303.07839](https://arxiv.org/abs/2303.07839) (2023)



Exploring the Use of LLMs to Understand Network Traces

Rubén De la Torre Vico, Roberto Magán-Carrión^(✉),
and Rafael Alejandro Rodríguez-Gómez

University of Granada, 18071 Granada, Spain
rubendltv@correo.ugr.es, {rmagan,rodgom}@ugr.es

Abstract. The growing threat of cyberattacks, combined with the increasingly complex digital landscape, has created a pressing need for cutting-edge cybersecurity solutions. The recent surge in large language models (LLMs) has brought both opportunities and risks, as their advanced language processing abilities can be harnessed for both good and evil purposes. In most cases for an organization, the network is the main gateway where security issues appear. The main problem with this issue is that the traffic of a big network is impossible to analyze by a human, that is why some of the tools used in networks analysis uses deep learning techniques to predict and solve problems. With this in mind and the recent success of LLMs for several types of tasks is the perfect motivation to check how good are the main models solving this hard job. For this purpose, in this paper we design a methodology to evaluate the ability of LLMs models to understand network traces using different prompt engineering techniques.

Keywords: LLM · prompting · network traces · cybersecurity

1 Introduction

The number of cyberattacks has increased significantly from 2010 to 2021, more than three times in this period [16]. Experts predict an even worse scenario in the years to come mainly because of the variety of devices, communications and the heterogeneous information they share.

In recent years, large language models (LLMs) have become increasingly popular, not only due to their impressive natural language processing capabilities, but also because they excel in other tasks. For instance, they have been utilized to fix bugs in applications [6], as a code programming assistants [11] and to assist in healthcare tasks [13]. However, LLMs could be used for malicious purposes as well. For instance, there was a significant increase in the number of unique social engineering attacks from January to February 2023, with a 135% of increment compared to the previous year [10]. Such increment coincides with the release of the now-famous ChatGPT in October 2022 and could be motivated by this fact. Since that time, the research community has been seeking to enhance

the capabilities of these models through the development of more sophisticated architectural designs, improved and optimized training strategies, expansion of context lengths, on demand fine-tuning, or to improve the multimodality characteristic of LLMs, among others advances [11].

In the context of cybersecurity, it could be argued that these tools could also be helpful to experts in the field, for example, in performing specific time-consuming tasks in the security incident response workflow. Due to their inherent characteristics, LLMs perform extremely well as human assistants. For this reason, they could be of great interest in their applicability in the analysis and mitigation of security events within a security operation center (SOC) [1].

To successfully carry out the analysis of a security incident, security analysts must perform a deep inspection of raw data sources, such as network traffic traces, system logs, NIDS alerts, etc. This task is time-consuming and relies on the analyst's background, experience and skills. Consequently, the total time required to resolve the security event is extended, which has a direct impact on the compromised system or network [5].

In this work, we propose a novel approach to facilitate the understanding and analysis of network traces based on the use of LLMs as assistants. For that, prompt engineering augmentation and validation techniques have been developed and tested. The use of LLMs to make sense of the vast amount of data being transmitted across the network could significantly aid cybersecurity experts in identifying potential threats. As a result, the total time it takes to resolve a security incident would be significantly reduced.

The remainder of the paper is organized as follows. Section 2 provides an overview of the progress made in the use of LLMs applied to cybersecurity. Next, Sect. 3 presents the general architecture of the proposed solution and the methodology used to evaluate it. Finally, the results are presented in Sect. 4 and some conclusions and future work are outlined in Sect. 5.

2 Related Work

In the book *From ChatGPT to HackGPT* [14], Karen Renaud et al. explore the idea of building smarter and more powerful security systems to detect or generate attacks using ChatGPT [12]. As they demonstrated, ChatGPT, and other LLMs models, have the potential to be utilized as a means of detecting phishing attacks. This can be achieved through the integration of GPTZero and ZeroGPT with mail servers.

Similarly, Repede et al. [15] employed the BERT model to identify fake news. Their findings indicate that their approach exhibited superior performance, outperforming other legacy and deep learning techniques.

Fayyazi et al. [4] highlight the inherent challenges in security event analysis and diagnosis, along with the ambiguity in interpretation by security analysts. They propose the use of NLP techniques to map natural language descriptions of incidents to MITRE ATT&CK's TTPs (Tactics, Techniques, and Procedures), reducing ambiguity and improving diagnosis accuracy. The authors compare the performance of LLM models to their base model (BERT) concluding that

SecureBERT performs better. However, while they mention classical ML and DL-based NLP techniques, they do not directly compare them with LLM models.

Chen et al. [2] introduce BERT-Log, a novel LLM model based on pre-trained BERT for anomaly detection of system log traces. The authors propose a new method to parse log information, which aims to capture the order and semantics of log traces, thus making them more amenable to natural language processing. BERT-Log outperforms other LLM and classical ML/DL models in anomaly detection. Similarly, Jüttner et al. propose ChatIDS [7], a security alert interpretation system that utilizes ChatGPT. Although the concept has considerable potential, the preliminary nature of the work and the lack of comprehensive validation make it premature for further consideration.

Wu et al. [19] explore in NetLLM the use of LLMs to address networking challenges. In this case the investigation focused on 3 specific tasks: (i) viewport prediction, to forecasts the next user viewport frames to load to reduce bandwidth consumption, (ii) adaptive bitrate streaming, to adjusts the user quality experience in the most optimal way, and (iii) cluster job scheduling that trains a reinforcement learning model to schedule and allocate computing resources for jobs within a distributed computing cluster. The authors have conducted a fine-tuning process that has enabled the resulting LLM to outperform state-of-the-art algorithms. Finally, they discuss the advantages, drawbacks and limitations of the use of LLM in the networking field. For instance, they conclude that the demonstration of increased performance and generalisation by NetLLM represents a preliminary step towards considering LLMs for these tasks in the future.

As it has been stated, LLMs have potential applications in a variety of areas beyond the NLP context. However, few works focus on understanding network traces, especially to solve complex open problems in security event analysis. Our work-in-progress attempts to advance the state of the art by the evaluation the potential application of LLMs in providing relevant insights, of interest to a security analyst, obtained from network traces by prompting them.

3 Methodology

The proposed methodology for evaluating the ability of LLM models to understand network traces is illustrated in Fig. 1 and is divided in three main stages:

1. **Data Collection:** Here we collect a set of network traces from open repositories on the Internet. A data filtering process is also performed at this stage.
2. **Dataset Generation:** A set of questions has been prepared to assess the ability to understand network traces of several LLM models. Once the network traces are collected, the correct answers to these questions are automatically calculated. For the next stage, we provide a dataset containing the correct answers to the questions and the network traces converted to text.
3. **Evaluation:** In this final stage, the necessary prompts are first generated using various prompt engineering techniques. These prompts are queried to a set of LLMs models and their responses are collected. Based on the responses

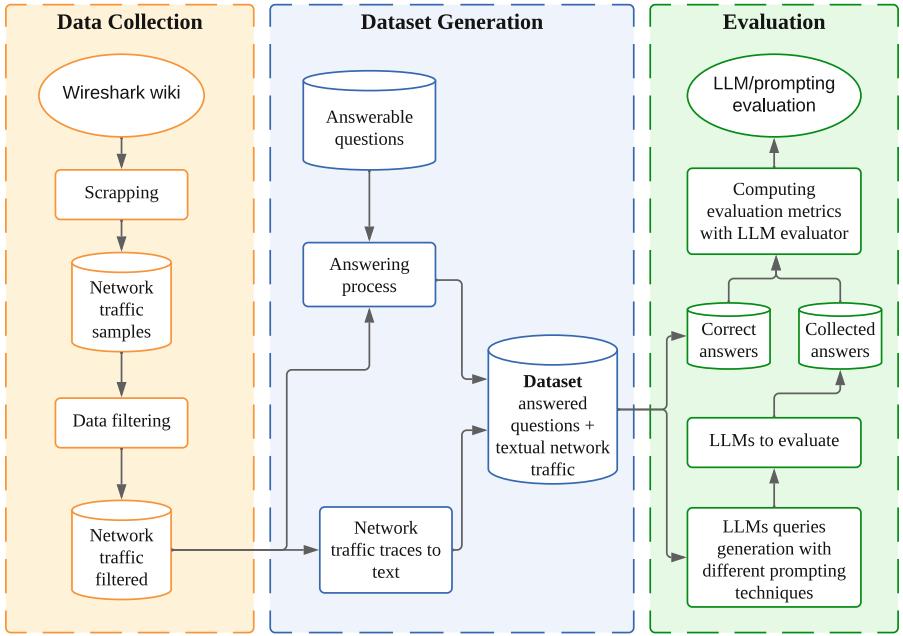


Fig. 1. General methodology for evaluating the ability of LLM models to understand network traces

obtained, another LLM model (LLM evaluator) is queried to evaluate whether the response obtained matches the responses generated in the previous stage, which is considered as the ground truth. Based on this, results are generated to compare the different prompting strategies and LLMs used.

3.1 Data Collection

To the best of our knowledge, there is no network dataset that can be used directly to feed an LLM. Consequently, we collect and preprocess network traffic traces following the next steps.

Network Traffic Traces Collected from the Internet. The initial step to obtain a substantial quantity of raw network traffic data has been to download all available sample captures from the Wireshark Wiki [3], which contains many network traffic samples in PCAP format. Once this scrapper has been developed similar sites with network traces samples such us Cloudshark can be used.

Data Filtering. Due to the limitation of the context window of LLM models, a maximum number of packets per trace must be selected. Any collected trace that exceeds this *filtering threshold* will be deleted.

As we will explain in the next stage, the evaluated LLM will use a textual version of the collected network traces. Empirically, it has been calculated that a network packet occupies about 30 tokens in its text version and the longest question considered contains ~ 50 tokens. Since the context window of the most restrictive model to be used in the evaluation allows 4000 tokens, we choose 128 packets as the filtering threshold. Using the worst case scenario we will have a network trace with 128 packets and a question with 50 tokens, so the number of tokens in the context window will be $128 \times 30 + 50 = 3890$ which is less than 4000, the upper limit of the context window of the smaller LLM considered.

3.2 Dataset Generation

The generation of the dataset comprising all the network traces from the previous stage is done in this step. Since the model is intended to assist network analysts, a Q/A (question and answer)-based dataset is generated. Specifically, the dataset should comprise network traces converted to text, along with a question about each network trace and the corresponding correct answer.

Generating a large Q/A dataset manually, which means that one person is asking questions and another is providing answers related to the network traces, is a time-consuming and hard process. To overcome it, we propose to automate the entire process. Consequently, the objective is to develop an approach to automatically derive the correct answer to several questions about network traces.

Designing Answerable Questions. The definition of the questions that can be answered automatically is a crucial step in creating the dataset. Therefore, a total of nine fundamental questions have been meticulously crafted to focus on the most basic aspects of each network trace e.g., the total number of packets of a network trace. These questions are not designed to generate a specialized model, but rather to assess its capability to comprehend the language and terminology found in network traces. These questions are:

- *What is the total number of packets in the trace?*
- *How many unique communicators are present in the trace?*
- *What is the IP that participates the most in communications in the trace?*
- *What is the total size of transmitted bytes?*
- *What is the average size of packets in bytes?*
- *What predominates in the capture the use of ICMP, TCP or UDP?*
- *How long in seconds does the communication last?*
- *What is the average of packets sent per second?*
- *What is the average bytes/s sent in the communication?*

All the questions are designed in such a way that they should be answered by numbers. In those questions where this is not true is because the logic behind the question is related to numbers like counting the number of appearances e.g., the IP that participates the most or what protocol predominates in the capture.

Automating the Answering Process. The pyshark [8] library was used to automatically answer each question. This approach involves traversing all packets and collecting the most relevant data. The wording of all type of answers is the same, the only thing that changes is the result according to the trace. For instance, considering the first question previously introduced, every answer will be: *In the capture there are a total of X packets*, being X the output variable whose value can vary depending on the used trace.

The generation of the dataset can be summarized in the pseudo-code described in Algorithm 1.

Algorithm 1. Dataset Generation Procedure

```

1: procedure GENERATEDATASET
2:   traces  $\leftarrow$  Load all network traces
3:   questions  $\leftarrow$  Load all generic questions
4:   for trace in traces do
5:     packets  $\leftarrow$  Extract packets from trace
6:     for packet in packets do
7:       info  $\leftarrow$  Extract relevant info (num. packets, IP, bytes, time)
8:     end for
9:     for question in questions do
10:      context  $\leftarrow$  "You are a network analyst ..."
11:      prompt  $\leftarrow$  Create prompt with trace and question
12:      answer  $\leftarrow$  Based on question use a template text and introduce the
        correct value extracted from info
13:      Append (context, prompt, answer) as row to dataset
14:    end for
15:  end for
16: end procedure

```

Network Traffic Traces to Text. Since LLMs were originally devised to understand natural language text, the downloaded network traces in PCAP format cannot be directly used. To do this a multi-modal LLM should be used. Multi-modality is the ability of an LLM has to understand data sources other than text, such as audio, video or PCAP files. However, although multi-modal models are increasing in quantity and quality, they are still far behind in development compared to models that only receive text as input. For this reason, in this work, we have chosen to transform network traces in PCAP format to text, which will subsequently be used to feed LLMs.

In order to transform PCAP binary network files to text, the tshark tool is used. By default, tshark outcomes and overview of each packet in the trace in text. Additionally, a “|” character between fields has been added to make easier the understanding of fields separations to the LLM. An example of the extracted information of a network packet can be observed in Fig. 2.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	145.254.160.237	65.208.228.223	TCP	62	3372 → 80
↪		[SYN]	Seq=0 Win=8760 Len=0 MSS=1460 SACK_PERM			

Fig. 2. Example of the extracted textual information from a network binary packet.

On the one hand, the pcap-to-text approach, which summarized the contents of a network packet, avoids the limitation of the maximum length of the context windows of the LLMs. The context windows contain the maximum number of tokens and prompts that can be managed in a single request. On the other hand, network traces are huge and difficult to manage in real time. Therefore, this type of summarizing or aggregation of relevant information is recommended. More information about the meaning of each extracted field can be seen in Table 1.

Table 1. Extracted field description of a network packet.

Feature	General Information
No.	Indicates the sequence number of the packet in the network trace.
Time	Timestamp of when the packet was captured, essential for analyzing packet sequence and timing.
Source	Identifier address of the device that sent the packet, crucial for identifying the origin of the traffic.
Destination	Identifier address of the device that received the packet, vital for understanding the packet's intended recipient.
Protocol	The highest layer protocol used to transmit the packet, essential for understanding the type of communication (e.g., HTTP, FTP, SSH, TCP, UDP, ARP).
Length	The size of the packet in bytes, important for analyzing packet overhead and potential performance issues.
Info.	Detailed information about the packet, including the source and destination ports if the protocol is using TCP or UDP, etc. This information is critical to understand what is the packet doing in the context of the communication.

3.3 Evaluation

This final evaluation stage takes as input the previously generated dataset consisting of the network traces in text format and the questions together with the automatically calculated correct answers. As output, this stage generates the success/fail rate of these questions for each of the LLMs evaluated and for each of the prompt engineering techniques used.

The steps of this stage are described below:

LLM Queries Generation. In this step, each LLM to be evaluated is prompted to answer all the questions in the dataset.

First, a set of questions to be asked to each LLM is generated. These questions are constructed by taking as input one of the questions from the answerable questions of the previous stage, reformulating it to follow the prompt engineering strategy to be evaluated, and adding the trace in text format on which this question is asked. Prompt engineering refers to the way in which questions are posed to an LLM model; depending on the strategy chosen, the output of an LLM query could be significantly improved [18].

Next, each LLM is queried to answer the previous set of questions. This is done for each of the selected prompt engineering strategies and for each LLM to be evaluated. As a result, all the answers provided by the models are collected.

Computing the Evaluation Metrics. Finally, to evaluate if the collected answers are correct, we use another LLM as an evaluator. To perform this evaluation, a question is constructed for the LLM evaluator, combining *i*) the answer collected in the previous step as a result of querying the LLMs to be evaluated, with *ii*) the correct answer generated by tshark in the dataset generation stage. The LLM evaluator assesses whether the collected answer is correct or incorrect, and returns the success/fail rate per LLM model and prompt strategy.

4 Results

The previous methodology was used to evaluate how LLMs perform in understanding network traces. In the data collection stage, we downloaded 433 network traces and after the data filtering step, this number was reduced to 297 traces. In the dataset generation stage, we transformed these 297 traces into text and combined them with the questions and their correct answers to generate our dataset. Finally, in the evaluation stage, LLama2 and Mistral LLMs were selected and in terms of prompting techniques, the following three were selected:

1. *Zero-Shot prompting* [9]: questions are done without relying on previous knowledge or experience.
2. *One-Shot prompting* [20]: prior of giving of the current question and network trace, a little example with tiny network trace, question and answer is given.
3. *Chain Of Thought* [17]: prior of giving of the current question and network trace, the model should be fed with an illustrative example that demonstrates the reasoning required to solve the question.

Figure 3 shows the success rate of each assessed model depending on the prompting technique used. In this first experiment, the success rate results to be low whatever the prompting technique used, with Mistral model outperforming Llama2. Regarding the prompting technique, the zero-shot approach outcomes better results than others more elaborated. This is a preliminary result that requires a larger number of experiments to fairly compare the different prompt engineering techniques.

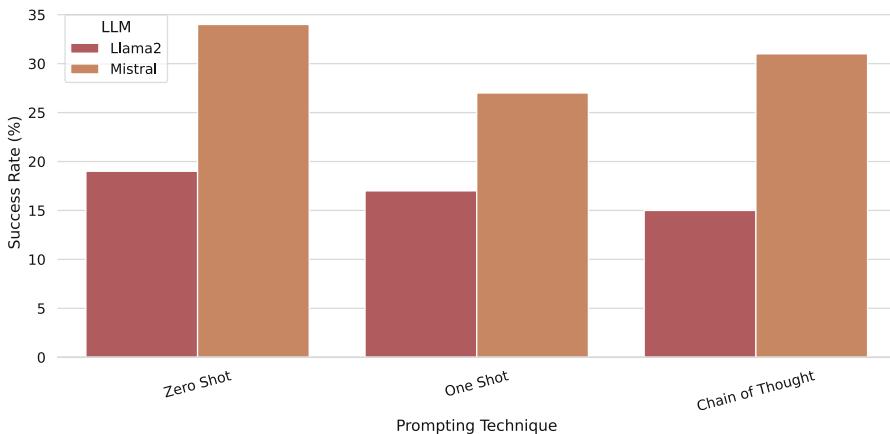


Fig. 3. Success rate comparison between LLama2 and Mistral by prompting technique.

5 Conclusions and Future Work

In conclusion, it can be stated that LLM systems could be a suitable tool to understand network traces, even though they are not intended for that purpose. They still have problems when dealing with logical and mathematical tasks, such as adding values to obtain a result. This paper explores the possibility not only to check if they can perform this task, but also to assess their performance and compare it with that of different models, such as LLama2 and Mistral, using different prompting techniques. Although further study is needed, these preliminary results seem to indicate that fine-tuning to interpret network traces in text format using these models.

Once a base methodology has been identified, subsequent research could be directed towards the fine-tuning of the models and the comparison of the results obtained with them. Furthermore, it would be beneficial to enhance the dataset by incorporating examples pertinent to the field of cybersecurity, as this would be the ultimate objective of the model.

Acknowledgments. This work has been supported by NetSEA-GPT (C-ING-300-UGR23) funded by Consejería de Universidad, Investigación e Innovación and the European Union through the ERDF Andalusia Program 2021–2027 and SICRAC (PID2020-114495RB-I00) of the Ministerio de Ciencia, Innovación y Universidades.

References

1. 10 Essential Capabilities of a Modern Security Operations Center. <https://splk.it/3PqfITR>
2. Chen, S., Liao, H.: BERT-log: anomaly detection for system logs based on pre-trained language model. *Appl. Artif. Intell.* **36**(1), 2145,642 (2022)
3. Combs, G.: Wireshark wiki. <https://wiki.wireshark.org/>

4. Fayyazi, R., Taghdimi, R., Yang, S.J.: Advancing TTP analysis: harnessing the power of encoder-only and decoder-only language models with retrieval augmented generation. arXiv preprint [arXiv:2401.00280](https://arxiv.org/abs/2401.00280) (2023)
5. IBM: Cost of a data breach 2023 | IBM (2023). <https://ibm.co/3RCTOjU>
6. Jin, M., et al.: InferFix: end-to-end program repair with LLMs. In: Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp. 1646–1656 (2023)
7. Jüttner, V., Grimmer, M., Buchmann, E.: ChatIDS: explainable cybersecurity using generative AI. arXiv preprint [arXiv:2306.14504](https://arxiv.org/abs/2306.14504) (2023)
8. KomiNewt: Pyshark. <https://pypi.org/project/pyshark/>
9. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large language models are zero-shot reasoners. *Adv. Neural. Inf. Process. Syst.* **35**, 22199–22213 (2022)
10. Nageab, W.M., Alrasheed, R., Khalifa, M.: Cybersecurity in the era of artificial intelligence: risks and solutions. In: 2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS), pp. 240–245. IEEE (2024)
11. Naveed, H., et al.: A comprehensive overview of large language models. [arXiv:2307.06435](https://arxiv.org/abs/2307.06435) (2023)
12. OpenAI: ChatGPT: a large-scale generative model for open-domain chat. <https://bit.ly/3zSSX6W> (2021)
13. Reddy, S.: Evaluating large language models for use in healthcare: a framework for translational value assessment. *Inform. Med. Unlocked* **41**, 101304 (2023)
14. Renaud, K., Warkentin, M., Westerman, G.: From ChatGPT to HackGPT: meeting the cybersecurity threat of generative AI. *MIT Sloan Management Review* (2023)
15. Repede, S.E., Remus, B.: A comparison of artificial intelligence models used for fake news detection. *Bull. Carol I Natl. Defence Univ.* **12**(1), 114–131 (2023)
16. Sharif, M.H.U., Mohammed, M.A.: A literature review of financial losses statistics for cyber security and future trend. *World J. Adv. Res. Rev.* **15**(1), 138–156 (2022)
17. Wei, J., et al.: Chain-of-thought prompting elicits reasoning in large language models. *Adv. Neural. Inf. Process. Syst.* **35**, 24824–24837 (2022)
18. White, J., et al.: A prompt pattern catalog to enhance prompt engineering with ChatGPT. arXiv preprint [arXiv:2302.11382](https://arxiv.org/abs/2302.11382) (2023)
19. Wu, D., et al.: NetLLM: adapting large language models for networking (2024)
20. Ye, X., Durrett, G.: The unreliability of explanations in few-shot prompting for textual reasoning. *Adv. Neural. Inf. Process. Syst.* **35**, 30378–30392 (2022)

CISIS: General Track - Cyber Attacks, Blockchain and Quantum Computation



Securing V2X Communication: A Hybrid Deep Learning Approach for Misbehavior Detection

Tiago Costa¹(✉), Cesar Analide¹, Antonio Costa¹, and Jose Cunha²

¹ ALGORITMI Research Center/LASI, University of Minho, Braga, Portugal
unchart6565@gmail.com

² Optare Solutions, Parque Tecnologico de Vigo, 35315 Vigo, Spain
jcunha@optaresolutions.com

Abstract. The emerging vehicle-to-everything communications lead to a future of Intelligent Transportation Systems. However, this also calls for set of novel mechanisms to address new vulnerabilities and security problems. In this context, misbehavior detection approaches aim to detect malicious behavior of rogue vehicle-to-everything entities and possible attacks that may originate from them. This paper proposes a hybrid deep learning approach for misbehavior detection in 6G vehicle-to-everything communication. The approach utilizes a two-stage architecture. The evaluation results show that the proposed approach achieves high accuracy in identifying misbehaving vehicles, including unknown ones. Additionally, the best performing model outperforms existing frameworks in terms of key evaluation metrics while also being lightweight for resource-constrained environments and having real-time requirements in mind.

Keywords: Misbehavior Detection · V2X Communications · Machine Learning · Artificial Intelligence · Deep Learning

1 Introduction

The growing advancements of Vehicle-to-Everything (V2X) communications unlocks a future of increasing connectivity and driving autonomy levels. The vast adoption of V2X connectivity leads to the actualization of the Internet of Vehicles (IoV) paradigm, a concept that has recently evolved from the Internet of Things (IoT). However, these technological advancements also introduce a larger attack surface that exposes vulnerabilities and allows evolving security threats.

This paper is motivated by these security threats and offers a solution for the problem of misbehaving vehicles in a V2X network. Misbehaving V2X entities intentionally transmit incorrect or misleading data. Identifying these nodes is crucial for guaranteeing the security and reliability of information exchanged within the V2X entities. Recent advancements in Artificial Intelligence (AI) offer promising solutions for misbehavior detection [1]. Supervised learning techniques

have been explored for identifying specific attacks, but they rely on labeled datasets, which can be impractical in real-world scenarios.

To address these limitations, this paper proposes a hybrid deep learning approach for V2X misbehavior detection. Our approach combines the strengths of unsupervised learning and supervised learning. The results demonstrate the effectiveness of the approach in identifying anomalies, including unknown ones while also taking in account the real time requirements of this dynamic environment.

This paper is organized as follows. Section 2 introduces some concepts of Deep Learning. Section 3 presents the network scenario. Section 4 details the proposed approach. Section 5 evaluates the proposed work and compares with previous works. Finally, Sect. 6 concludes the paper.

2 Background

This section presents a brief background explanation of the different types of Deep Learning (DL) Models explored in this work. By employing different DL models within our approach, we aim to compare their performance and identify the most effective model for V2X anomaly detection. DL is an alternative to traditional learning methods due to their ability to automatically learn intricate patterns from vast amounts of data and, usually, obtaining better performance. Generically, DL models are a great choice in the real-time context of V2X applications, since they automatically extract features from data, which normally eliminates the need for manual feature engineering.

Here, we provide a brief overview of the deep learning architectures studied in this work:

- **Autoencoders:** Autoencoders are a type of DL architecture designed to learn efficient representations of data. They are composed by two parts: an encoder that takes the input data and compresses it into a lower dimensional representation, and a decoder that attempts to reconstruct this representation into the original input.
- **Long Short-Term Memory (LSTM):** LSTMs are a type of Recurrent Neural Network (RNN) that were explicitly designed to address the vanishing gradient problem that RNNs suffer. They use a memory cell and gates to control the flow of information, which allows them the ability to retain information. This makes them great in the task of sequence and time series processing since they learn the relationships between past values and future values.
- **Convolutional Neural Network (CNN):** CNN is a class of neural networks that specializes in processing data that has a grid-like topology, such as images and time series sequences. They extract spatial features from data by applying filters to the input which leads to the capture of hidden patterns. A CNN typically has three layers: a convolutional layer, a pooling layer, and a fully connected layer.

- **CNN-Bidirectional LSTM (CNN-BiLSTM):** Bidirectional LSTMs have the same purpose as LSTMs, however they allow the processing of data in both forward and backward directions. This permits the network to capture dependencies from past elements in the sequence and also from future elements. This leads to better understanding of the sequence. Combined with CNN, these architectures can benefit from the strengths of both.
- **Stacked LSTMs:** Stacked LSTMs involve building multiple LSTM layers on top of each other. Each layer processes the output of the previous layer, allowing the network to learn increasingly complex hierarchical representations of the input data. This can be beneficial for tasks that require the extraction of intricate features from complex sequences.

3 Network Scenario

In this article we follow the V2X network scenario that is represented in Fig. 1. This network is composed by vehicles that periodically send Basic Safety Messages (BSM) with data like position, speed and acceleration. These messages are received by the roadside unit (RSU) and then the edge/cloud servers aggregate the messages of various RSUs. The approach followed converts the data received in the edge servers into sequences, that are then passed to a Deep Learning model that will classify as an intrusion or not. The assumption made is that the models are trained on the cloud servers and that the inference prediction service will run in the edge servers. This offers 2 advantages: the *cloud servers* are cost and computation-intensive which makes them useful for model training, and *edge servers* bring the computation closer to the nodes reducing the latency. In this scenario, a vehicle will be considered an attacker if their message contains falsified information.

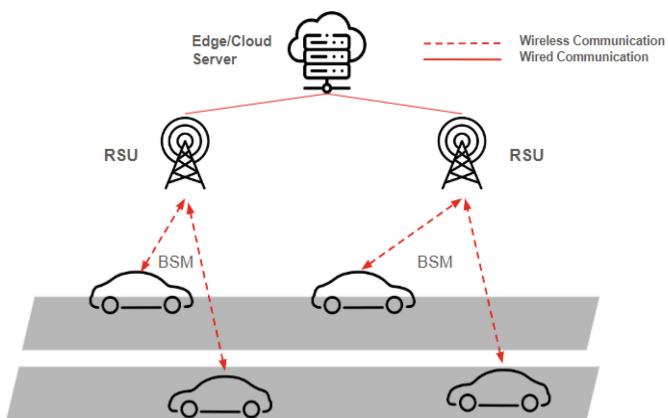


Fig. 1. V2X network scenario

4 Proposed Misbehavior Detection Approach

In this section we discuss the proposed approach that can effectively detect misbehaving vehicles in the considered network scenario.

The hybrid methodology followed can be defined into 2 stages that will be explained below. By using this two-staged approach we can take advantage of the unsupervised anomaly detection and supervised learning to achieve an efficient and adaptable misbehavior detection system.

- **Stage 1: Unsupervised Anomaly Detection**

- i) Raw V2X network data is processed to extract Basic Safety Messages (BSMs) and converted to sequences.
- ii) An unsupervised deep learning model reconstructs the sequences and calculates a reconstruction error.
- iii) Based on this error, a potential threshold is calculated via a thresholding algorithm.
- iv) Using this potential threshold, the sequence is classified as anomalous or genuine and generates a labeled dataset for the next stage.

- **Stage 2: Supervised Learning for Classification**

- i) Leveraging the labeled dataset from the stage 1, we train a supervised deep learning model. This model learns the relationship between the BSM features and labels (genuine/anomalous).
- ii) During deployment, the trained supervised model receives incoming BSM sequences as input. It analyzes the BSM features and classifies the sequences as normal or attack in real-time.

4.1 Data Preprocessing

To evaluate the proposed approach we used the VeReMi Extension dataset [2]. This is a publicly available dataset with the purpose of being a reference in the area of misbehavior detection in V2X networks. It was generated by using VEINS, an open-source simulation tool [3]. Even though the dataset is simulated, the implemented scenario is based in real traffic data provided by the Luxembourg SUMO Traffic (LuST) scenario.

The dataset comprises 19 types of attacks and the data fields present are the send time, sender ID, sender pseudo ID, message ID, and position-based information - position, speed, acceleration, and heading. Due to limited computational power, just the position and speed vectors from the position-based information were used to the reconstruction of the sequences. The reason to choose the sender pseudo ID and not the actual ID of the vehicles is because of a V2X privacy requirement, since these systems must maintain private the identity of their participants. Since this is a time-series data, it can be segmented into windows. For windows creation we opted for a window size of 20 and step size of 10. These values come from related state of art works that also used the VeReMi Extension dataset [4,5]. Before passing the window to the models, the data is

normalized with a standard scalar using the following equation $z = (x - u)/s$, where z is the normalized data, x is the input data, u is mean and s is the standard deviation.

4.2 Unsupervised Anomaly Detection

To the unsupervised anomaly detection we employed 3 different Deep Neural Networks (DNN) - Model 1: LSTM Autoencoder (M1), Model 2: CNN-LSTM (M2) and Model 3: CNN-BiLSTM (M3). The configurations of their architectures are explained as follows. In M1 we used 2 encoder layers and two decoder layers, each with 1024 and 512 LSTM units respectively. The M2 is composed by 2 CNN layers of 1024 and 512 layers respectively and 4 stacked layers of 512 LSTM units. To the M3 we opted for 2 CNN layers of 1024 and 512 units, and 4 stacked layers of BiLSTM units. The units for the models were chosen based on several experiments performed to arrive at the best results.

In this stage of the approach each model takes as input, sequences of 20 messages and 6 features (X and Y of position and speed vectors, the send time and sender pseudo ID). The idea in this stage is to detect anomalies, by reconstructing the sequences passed as input and checking if the reconstruction error is superior to a threshold. To do so, each model was only trained in genuine sequences to learn the representation of a genuine behavior. So, when the model is passed a sequence, it will reconstruct that sequence producing a sequence whose pattern and values should be identical to the genuine data sequence. We then calculate the reconstruction error and pass this values to the thresholding algorithm. To calculate the reconstruction error we used the mean squared error between the reconstructed sequence and the original sequence.

4.3 Thresholding Algorithm

Also belonging to the first stage of the approach, the thresholding algorithm takes the reconstruction errors from the model and calculates the threshold that dictates if the sequence is anomalous or not. If the reconstruction error is higher than the threshold then it is a misbehaving sequence, and it is classified as genuine if the error is smaller than the threshold. To identify the most effective threshold for anomaly classification, we define a list of possible thresholds and at each iteration we generate the F1-Score of the generated dataset to get the most effective threshold. We analyze the performance metrics across different thresholds since the goal is to find the threshold that maximizes the F1-Score. This metric was chosen because it balances precision and recall, ensuring that the model effectively identifies anomalies while minimizing false positives which is important in a V2X real-time scenario. This algorithm also produces a visual report as it can be seen in Fig. 2, where it explains the variation of the performance metrics versus threshold values.

The output of the thresholding algorithm is a labeled dataset that will be passed to the supervised algorithm, reducing the need of a manual labeling which is impractical in a real-time V2X scenario.

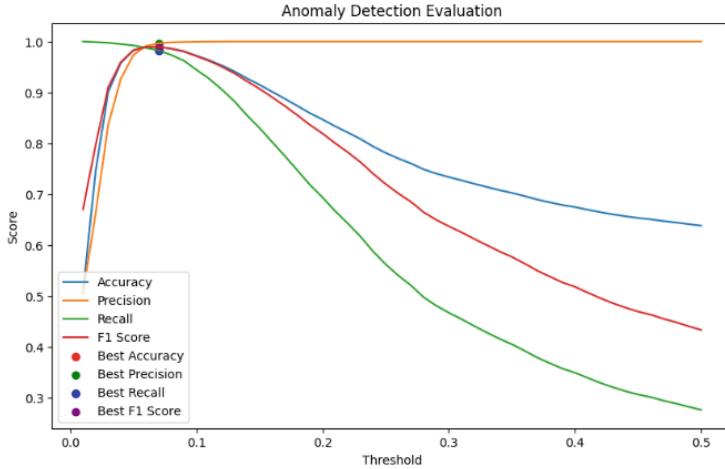


Fig. 2. Variation of Threshold Values vs Performance Metrics for the best model (M1)

4.4 Supervised Learning for Misbehavior Detection

The second stage of the proposed approach is the supervised learning for misbehavior detection. This stage is the one that is deployed and the one that detects the misbehaviors in the proposed V2X scenario. To this task we employed 2 different DNN : Model 1: CNN (S1) and Model 2: Stacked LSTM (S2). These models were chosen since the authors in [6,7], have already assessed that these are the best performing models to the given task. To define the number of units in each model several experiments were performed to arrive at the ones that held the best results.

In [6], the results show that converting the sequence into an image and then passing to a CNN is the best approach to detect misbehavior. So, taking these results into account we convert our input sequence of 20×6 into an image and then we pass it to the S1 model. The S1 is composed by 2 2D CNN layers of 256 and 128 units respectively, being followed by 2 dense layers of 100 and 10 units. The output is also a dense layer, however with just one unit and activated by the *sigmoid* function since we want to classify the sequence as attack or genuine.

The S2 model is a result of the conclusions that the work in [7] made, where they discovered that the 4 stacked LSTM yield the best performance in the task of sequence classification. So, our model follows this architecture and it is composed by 4 layers of 256 LSTM units. The output layer is also a dense layer with just one unit activated by the *sigmoid* function.

5 Simulation Results

In this section we discuss the obtained results for all the models, as well as the dataset splits used for the training, testing and validation.

5.1 Training and Testing

Tensorflow [8] was used to the development of all the models, while the training and testing of these was carried out on Google Colaboratory environment [9]. The splits done for the model training are discussed below.

1. *Dataset for the Unsupervised Learning* - The train dataset for this task has a total of 66500 sequences. It is composed only by genuine sequences from each of the attack scenarios, meaning that it has 3500 genuine sequences from each attack scenario. This was made to expose the models to every type of genuine behavior. The test dataset for this stage has a total of 15200 sequences. It was split having 7600 sequences of attacks and 7600 sequences of genuine behavior. Another important step taken was to grant that the number of sequences of each type of attack are equal.
2. *Dataset for the Supervised Learning* - The train dataset for the second stage is the dataset generated by the previous stage and has a total of 15200 sequences. To the testing and validation, this dataset is split in 30% and 10% respectively, while the remaining 60% are used for training.

To the unsupervised learning training we used the Adam optimizer [10] and the Mean Absolute Error as the loss function while using a batch size of 20. In stage 2, the training was also performed using the Adam optimizer, however the loss function used was the binary crossentropy. The supervised models were trained using an Early Stopper to ensure no overfitting was occurring. The unsupervised models were trained by 100 epochs each and the method used to test the models was using k-fold cross-validation with k=3. The results from all folds were then averaged to obtain a more robust estimate of the models' performance.

5.2 Unsupervised Learning Evaluation

In this subsection we evaluate the first stage of the hybrid approach in terms of four metrics. The metrics chosen were Accuracy, Recall, Precision and F1-Score. Normally, one wouldn't be able to assess these metrics in the unsupervised learning phase since they require labels, however in the case of the VeReMi dataset we are provided labels and so we can perform this step to evaluate the performance of this stage.

Table 1 shows the overall precision, recall and F1-score of Genuine and Anomaly classes along with their accuracy of prediction. From these results it is evident that M1 is the best model with an accuracy of 0.9895. It also is possible to see the precision and recall of 1.0 and 0.98, respectively, in the attack class, which shows the efficacy of the model in distinguishing what an attack is. From the 3 presented models, M1 is the one that has the highest F1-Score, which ensures the model effectively identifies anomalies while minimizing false positives. The other 2 models, while being inferior in terms of performance comparing to the first one also show great capability in detecting attacks, with M2 showing an accuracy of 0.9719 and M3 with an accuracy of 0.9820.

Table 1. Evaluation Metrics for Anomaly Detection

Model	Class	Precision	Recall	F1-Score	Accuracy
M1	Attack	1.00	0.98	0.99	0.99
	Genuine	0.98	1.00	0.99	
M2	Attack	0.98	0.96	0.97	0.97
	Genuine	0.96	0.98	0.97	
M3	Attack	0.99	0.98	0.98	0.98
	Genuine	0.98	0.99	0.98	

5.3 Hybrid Approach Evaluation

Despite the performance of the first stage being already excellent, the hybrid approach that we developed offers some advantages when compared to the deployment of the first stage alone. One of the first advantages that we were waiting was the increase of performance in the evaluation metrics, which happened as we can see in Table 2. The other is related to the real time requirements of the V2X scenarios. Even though there are still no official requirements related to intrusion detection in the 3GPP Releases, the inference time of a model must be taken into account since it has a great impact in the safety of the vehicles. This is where the combination of the 2 stages comes into place.

Table 2. Hybrid Approach Evaluation

Model	M1	S1	S2	Optimization
Precision	0.9964	0.9865	0.9830	-0.99%
Recall	0.9825	0.9956	0.9944	+1.33%
F1-Score	0.9894	0.9910	0.9887	+0.17%
Accuracy	0.9895	0.9911	0.9887	+0.16%
Parameters	15 Million	1.3 Million	2 Million	+91.33%
Prediction Time	0.1666 s	0.0661 s	0.0759 s	+60.24%

The unsupervised models developed have a lot of complexity, with the best model (M1) having 15 million parameters which reflects in the inference time. In Table 2 we can see the optimization we gained by deploying the supervised models when comparing to the best stage one model (M1). Another salient information is that the S1 model is the one that outperforms model M1, while model S2 maintains the same performance achieved by M1. With S1 we achieved better performance results in three of the evaluation metrics, such as accuracy, recall and F1-score. In terms of number of parameters, we obtained an optimization of 91.33%, by reducing from 15 million parameters to 1.3 million. This is reflected in the inference time that goes from 0.1666 s to 0.0661 s. The best performing

approach is the use of a LSTM autoencoder (M1) in stage one with a CNN (S1) in stage two, which grants an accuracy of 0.9911 and a F1-Score of 0.9910.

5.4 Comparison with Prior Works

Several different works proposed schemes for effective anomaly detection [11, 12]. Most of these works rely in the labeling of the dataset so they use only supervised approaches. Our proposed approach combines the unsupervised model to automatically generate labels (normal/anomaly) significantly reducing the labeling effort, with a supervised model that leverages these labels to improve the detection ability. Another advantage of our hybrid architecture is the fact that it is trained only on the genuine sequence data, which makes it useful in identifying any other type of anomaly based on its reconstruction error and the previously calculated threshold. This makes our approach identify all 19 types of attacks present in the VeReMi effectively. In Table 3 we compare the results of our proposed approach with the works proposed in [4, 7]. When compared to [4], that uses an unsupervised approach, we can see that our work achieves better performance in metrics like Recall, Accuracy and F1-Score. Additionally, when comparing to a supervised approach like [7], we outperform all the performance metrics. This results can be explained by the performance of the developed unsupervised model, that is able to capture the anomalies with great accuracy. The addition of the second stage, refines the classification and thus we obtain a better performance when compared to other approaches.

Table 3. Comparison with other works

Model	Precision	Recall	F1-Score	Accuracy
Hybrid Approach	0.987	0.996	0.991	0.991
DeepADV [4]	0.996	0.956	0.976	0.980
Stacked LSTM [7]	0.979	0.979	0.979	0.986

6 Conclusion

The recent advancements in V2X communications have improved existing transportation systems by increasing the connectivity and driving autonomy levels. However, the benefits of V2X come with challenges like security vulnerabilities and breaches. This article addresses these challenges by proposing a Hybrid Deep Learning Approach for Misbehavior Detection in 6G V2X communication.

Our proposal takes advantage of the strengths of a 2 stage architecture. The first stage utilizes a deep learning model, such as an LSTM Autoencoder or CNN-LSTM, to reconstruct BSM sequences and calculate reconstruction errors. These errors serve as anomaly scores that are passed to a threshold algorithm.

The supervised stage employs a deep learning model trained on the labeled data generated by the previous stage to refine the classification of genuine and anomalous BSM sequences.

This paper evaluated the performance of the hybrid architecture using three different unsupervised DNN models for the first stage and two supervised DNN for the second stage. The results demonstrate the effectiveness of the framework in identifying anomalies, including unknown ones, based on reconstruction errors. Additionally, the best-performing model surpassed the performance of existing frameworks in terms of key evaluation metrics while also being lightweight for resource constrained environments.

References

1. Sedar, R., Kalalas, C., Vázquez-Gallego, F., Alonso, L., Alonso-Zarate, J.: A comprehensive survey of V2X cybersecurity mechanisms and future research paths. *IEEE Open J. Commun. Soc.* **4**, 325–391 (2023)
2. Kamel, J., Wolf, M., van der Hei, R.W., Kaiser, A., Urien, P., Kargl, F.: VeReMi extension: a dataset for comparable evaluation of misbehavior detection in VANETs. In: ICC 2020 - 2020 IEEE International Conference on Communications (ICC), pp. 1–6 (2020)
3. Sommer, C., German, R., Dressler, F.: Bidirectionally coupled network and road traffic simulation for improved IVC analysis. *IEEE Trans. Mob. Comput.* **10**(1), 3–15 (2011)
4. Alladi, T., Gera, B., Agrawal, A., Chamola, V., Yu, F.R.: DeepADV: a deep neural network framework for anomaly detection in VANETs. *IEEE Trans. Veh. Technol.* **70**(11), 12013–12023 (2021)
5. Hossain, S., Boualouache, A., Brik, B., Senouci, S.-M.: A lightweight 5G-V2X intra-slice intrusion detection system using knowledge distillation. In: ICC 2023 - IEEE International Conference on Communications, pp. 1112–1117 (2023)
6. Alladi, T., Kohli, V., Chamola, V., Yu, F.R., Guizani, M.: Artificial intelligence (AI)-empowered intrusion detection architecture for the internet of vehicles. *IEEE Wirel. Commun.* **28**(3), 144–149 (2021)
7. Alladi, T., Kohli, V., Chamola, V., Richard Yu, F.: Securing the internet of vehicles: a deep learning-based classification framework. *IEEE Netw. Lett.* **3**(2), 94–97 (2021)
8. Abadi, M., et al.: TensorFlow: large-scale machine learning on heterogeneous systems (2015). Software available from tensorflow.org
9. Bisong, E.: Google Colaboratory, pp. 59–64. Apress, Berkeley (2019)
10. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization (2017)
11. Gyawali, S., Qian, Y., Hu, R.Q.: Machine learning and reputation based misbehavior detection in vehicular communication networks. *IEEE Trans. Veh. Technol.* **69**(8), 8871–8885 (2020)
12. Sharma, A., Jaekel, A.: Machine learning based misbehaviour detection in VANET using consecutive BSM approach. *IEEE Open J. Veh. Technol.* **3**, 1–14 (2022)



Use of GAN Models and CNN Heatmaps in Malware

Xabier Ceballos Diez, Iker González Requejo, Adrián Martínez Navarro,
Eduardo Jorge Sanjurjo Aramburu, José Gaviria de la Puerta^(✉),
and Iker Pastor-López

University of Deusto, Avenida de las Universidades 24, 48007 Bilbao, Spain
jgaviria@deusto.es

Abstract. Malware has become an increasingly critical issue in the field of cybersecurity, as the latest malware techniques are designed to evade most of the malware detectors nowadays, resulting in the traditional security techniques becoming obsolete. Some of the current techniques are developed through the use of neural networks. These include malware generation using Generative Adversarial Network (GAN) models, and the use of Convolutional Neural Network (CNN) models. This survey will present the current state of both CNN and GAN model frameworks, and the potential advances that the combination of both models can bring to the current context of the malware field.

Keywords: Malware · GAN · Adversarial · CNN · Neural network · Heatmap · Family · Classification

1 Introduction

Malicious software, commonly known as **malware**, is referred to any program created to cause harm on a computer system or device, usually affecting to the integrity of the data and control of the system. There are several types of malware, such as **Ransomware** (ransom software), which is a subset of malware designed to restrict access to a system or data until a requested ransom amount from the attacker is satisfied. Another type of malware is **Spyware**, which is software that illegally gathers information and sends it to an attacker's Command and Control server , among others. As we can see, each type of malware attacks these networks and devices in different ways. Thus, the security of these systems is significantly affected, ranging from loss of relevant data on a single device to massive attacks on whole organizations.

Traditional methods for malware detection include analyzing the code of potentially malicious software through both static and dynamic analysis techniques. On the one hand, static code analysis works by disassembling and searching for malicious patterns during the code. On the other hand, dynamic code analysis works by executing the code in a virtual environment and analyzing its behavior, generating a behavioral report based on the execution trace [1].

Generative adversarial networks (GANs) are a type of deep learning model invented by Ian Goodfellow. They are inspired by a two-player zero-sum game, consisting of a **generator** and a **discriminator**, both trained through adversarial learning. The objective of GANs is to approximate the potential distribution of real data samples and generate new samples that mimic this distribution [2]. In this way, GANs are crucial in the field of cybersecurity, as they allow the creation of new malware samples to train and improve current security systems.

Convolutional Neural Networks (CNNs) are a type of feedforward neural networks that can automatically extract features from data using convolutional layers. Unlike traditional feature extraction methods, CNNs do not require manual feature extraction. As neural networks, they are based on artificial neurons, which work as biological neurons. **CNN kernels**, also called convolutional layers, act as different receptors that respond to various features, applying different filters to the data input. **Activation functions** simulate the process where only neural electric signals exceeding a certain threshold are transmitted to the next neuron. **Loss functions** and **optimizers** are designed to train the entire CNN system to learn the desired outcomes [3].

Combining all of the above, the main purpose of this survey is to present an alternative way of recognizing and classifying different types of malware using **GANs** (Generative Adversarial Networks) and **CNNs** (Convolutional Neural Networks), diving into their functioning and nature throughout the different sections of this survey. Thus, the contributions of this work can be summarized in two parts. First, we present the fundamentals of GAN applied to malware. Second, we show how malware can be classified by using CNNs. This survey will end with a section related to future projects and its final conclusions.

2 Malware Generation by Using GANs

In this section, we will explain the process the generator and the discriminator follow in order to generate malware followed by the existing types of GAN, to end by mentioning some advantages GANs can bring to malware generation.

2.1 Fundamentals of GAN Applied to Malware

Description of the Process: On the generator side, it creates an image through nonlinear transformations on a Z vector, using deconvolution layers [4] that allow the network to learn through nonlinear operations. To stabilize the training process, Batch Normalization [5] is used to normalize the inputs in order to have a unit variance and mean of 0. Finally, to scale the output to a specific range, it uses RLU [6] in the layers.

On the discriminator side, it applies nonlinear transformations to distinguish between real and synthetic. For it to learn, convolution layers with stride [7] are used to replace the deterministic pooling functions and the network learns its own spatial downsampling. It also uses Batch Normalization [5], but to establish the gradient flow [8]. To improve the flow, Leaky ReLu [9] is used in modeling higher resolutions because it allows some negative gradients to pass through.

Types of GAN: Due to various weaknesses that GANs may have, such as mode-dropping during training, GAN variants have been developed. In this section, we briefly introduce some types of GANs, and they will be explained in depth at a later date together with further variants.

- **GAN + noise:** This variant of GAN [10] adds noise to the training process to stabilize it and improve the quality of the generated samples as it helps the discriminator to improve further by having more inputs. This often helps to cope with problems such as mode-collapse.
- **WGAN:** Addresses issues such as mode collapse and training stability by improving gradient computation and numerical stability. This is achieved through minimizing the Wasserstein distance [11] (loss metric used) between the probability distribution of real data and that generated by the generator [11].
- **DCGAN:** Uses convolutional layers to handle images and learn spatial features, allowing them to produce sharper and more realistic images. When it comes to generating high-quality images, they are easier to train than ordinary GANs [12].

Advantages of Using GANs: The use of GANs to generate malware brings with it many advantages. For example, the realistic and varied generation of malware. The use of Batch Normalization [5] helps to make the mean and variance 0 through a normalization process, thus preventing samples from being unvaried and therefore undetectable. An example of camouflaged malware is MalFox [13], a sample capable of camouflaging itself to Black-Box Detectors.

Another advantage of GANs is the ability to be trained and learn constantly and not become obsolete. In addition, the ways to train them are not limited at all. An example of a GAN training method can be hyperspectral image classification [14].

Finally, the ability of GANs to discover attack patterns and techniques that may not have been considered by security researchers is important. This can lead to the creation of malware with more advanced and sophisticated tactics.

Case Studies. Several studies have addressed malware detection using various approaches, such as transforming binary information into grayscale images for detection and using Windows Audit Logs and API calls [15–19]. Machine learning techniques like hidden Markov models, SVMs, and random forests have been applied, with BM Khamas using a random forest classifier to detect ransomware [20] and Abien Fred M. Agarap using SVM for malware classification [21].

Advancements in computing have enabled the use of deep learning and GANs for malware detection. Pascatu et al. used recurrent neural networks for time-series data, while Ye et al. developed a heterogeneous deep learning model [22]. Kim et al. used converted malware images as CNN inputs [16], and Yan et al. combined CNNs with long short-term memory networks [23].

Despite these advancements, many approaches are limited by their focus on specific malware variants and difficulties with new obfuscation techniques.

Hybrid techniques combining static and dynamic analyses offer better detection but are complex and time-consuming. For zero-day malware detection, Alazab *et al.* achieved high accuracy using data mining and Windows API calls with Naïve Bayes and SMO [24], while Gandotra *et al.* combined static and dynamic analysis for better accuracy and reduced modeling time [25]. Kim *et al.* developed a transferred deep-convolutional GAN (tDCGAN) to enhance zero-day attack detection [26], though these methods often struggle with the diversity and quality of generated samples.

PlausMal-GAN. In this section, we discuss the study by Won, Jang, and Lee (2023) on using generative adversarial networks (GANs) to create plausible malware for enhancing zero-day malware detection. Their research, “PlausMal-GAN: Plausible Malware Training Based on Generative Adversarial Networks for Analogous Zero-Day Malware Detection” [27], introduces a method that uses generated data to train detection models, emphasizing the role of malware data augmentation and GANs in addressing new cybersecurity threats.

Zero-day malware refers to newly discovered or unseen malicious software that escapes detection by current anti-malware solutions [28]. The PlausMal-GAN framework generates high-quality and diverse images of such malware from existing data. A generator and discriminator are trained with real and generated malware images, with the discriminator learning malware characteristics to improve zero-day malware detection.

1. First phase:
 - **Generator:** Produces analogous malware samples from real malware data.
 - **Discriminator:** Learns to distinguish between real and generated malware samples and classifies the malware types.
2. Second phase:
 - Generator is fixed, and the discriminator is retrained using real and generated data to refine its zero-day malware detection capabilities.

The PlausMal-GAN framework demonstrated higher and more stable performance in detecting analogous zero-day malware images. Several representative GAN models (DC-GAN, LSGAN, WGAN-GP, and E-GAN) were evaluated, showing that the framework is beneficial for the detection and prediction of numerous analogous zero-day malware data.

1. Quality and diversity:
 - **Frechet inception distance (FID):** FID is used as a quantitative metric to evaluate the generator’s convergence [29]. Lower FID implies a smaller distribution distance between real and generated images, indicating high quality and diversity of the generated samples.
 - The proposed framework achieved the lowest FID scores compared to other models, indicating high quality and diversity in the generated malware samples.

2. Classification performance:
 - Ten-fold cross-validation was used for all methods in the existing malware classification experiment. The proposed framework achieved an average accuracy of 95.56%, significantly outperforming previous methods.
 - In zero-day malware attack experiments, the proposed framework maintained high and stable performance even with significant variations in existing malware.
3. Evaluation with limited training data:
 - Additional experiments were conducted under limited training data conditions, showing that the proposed framework maintained stable and high performance (>99%) even with halved training data with E-GAN and a SSIM around 0.6–0.7 with ratios of 8:2 and 7:3. The Structural Similarity Index Measure (SSIM) is a method for measuring the similarity between two images.

The study demonstrates that the use of the PlausMal-GAN framework is advantageous for the detection and classification of zero-day malware, providing a robust method to enhance computer security and develop more effective malware detection systems. This approach can be integrated into current detection systems to improve the ability to detect zero-day attacks and unknown malware.

GAN Models. In this section, we provide an overview of various GAN models, highlighting their key characteristics and advantages, particularly in image synthesis and malware generation.

1. **DCGAN.** Deep Convolutional Generative Adversarial Networks (DCGAN) use deep convolutional neural networks for both the generator and discriminator. Introduced by Alec Radford *et al.*, DCGANs generate realistic images and stabilize training by employing specific convolutional architectures, avoiding pooling layers, and using strided convolutions. Batch Normalization and ReLU activations in the generator, and LeakyReLU in the discriminator, contribute to model stability [30].
2. **WGAN.** Wasserstein GANs (WGAN), proposed by Martin Arjovsky *et al.*, utilize the Wasserstein distance instead of the Jensen-Shannon divergence to measure the difference between real and generated data distributions. This results in a more stable loss function and improved sample quality, facilitating generator optimization [31].
3. **WGAN-GP.** Improved WGAN (WGAN-GP) introduces a gradient penalty to maintain the critic function in the 1-Lipschitz space, avoiding issues with weight clipping. This enhances training stability and sample quality, allowing more complex architectures without intensive hyperparameter tuning [32].
4. **LSGAN.** Least Squares GANs (LSGAN) use least squares loss instead of binary cross-entropy loss for the discriminator's cost function, improving training stability and generating higher quality samples by reducing the vanishing gradient problem [33].

5. **E-GAN.** Energy-Based GANs (E-GAN) integrate an energy term into the cost function to stabilize training, resulting in improved training stability and sample quality [34].
6. **PGGAN.** Progressive Growing GANs (PGGAN) propose a progressive training approach that starts with low-resolution images and gradually increases the resolution. This allows for more stable training and high-resolution image generation. PGGAN has been pretrained on the CelebA-HQ dataset [35].
7. **GAN Inversion.** GAN Inversion techniques aim to invert the generative process to map real examples to their corresponding latent space representations, allowing for better understanding and manipulation of generated features, facilitating the analysis and adjustment of generated malware.
8. **BigGAN.** BigGAN expands traditional GAN architecture and model size, optimizing for training with large batch sizes. It achieves outstanding results in generating realistic, high-resolution images and has been pretrained on ImageNet [36].
9. **Style-based GANs.**
 - **StyleGAN.** Introduced by Karras et al., allows for granular control over generated image features using a style-based approach [37].
 - **StyleGAN2.** Improves image quality and training stability compared to its predecessor [38].
 - **StyleGAN2-Ada.** Introduces adaptations to improve image generation in low-data training conditions [39].
 - **StyleGAN3.** Continues to improve stability and quality, addressing pixel alignment issues present in earlier versions [40].

3 Malware Classification with CNNs

3.1 CNN Application for Malware Classification

CNN's application to classify malware is very effective as it is capable of being dynamically trained. The training process goes through three phases: An initial preprocessing stage, the classification module and the evaluation of the neural network performance [41].

Preprocessing: the malware file is converted into an array of values that can be processed by the CNN. This is done by transforming the binary file into an image. Each byte of the file is converted into a pixel value, allowing the file to be directly represented as a grayscale or RGB image [41].

Classification: This classification module takes the preprocessed images and uses them to train the CNN or evaluate the images using the CNN. A common problem is that malware images often vary in size, and a CNN usually accepts only one input size, and varying this size can lead to pixel loss which results in loss of information. Attackers can predict this and inject malicious code into those pixels that would be deleted and thus remain undetected [41]. To avoid

this problem, the use of SPP [42] is common as it allows input images to be arbitrarily sized, eliminating the need to crop or warp images to fit a fixed size, thus avoiding data loss. The principal functionality of this layer is to produce a fixed-size feature matrix by using pooling-operations [43] that is processed by the fully connected layers for final malware classification.

Evaluation: To evaluate and validate the performance of the CNN, techniques such as bilinear interpolation [44] are used. To perform this evaluation action, it uses the classification module to classify undetected malware files and evaluates its success. The CNN output comes out with a result from 0 to 1 depending on the probability that the file is malicious.

3.2 CNN Architectures Used for Classification

In a study for discriminative localization [45] we are shown different CNN architectures for testing Class Activation Mapping. In this section we will give a brief explanation of each one because, depending on the field of use, one or the other may be more interesting:

- **AlexNet:** AlexNet [46] consists of eight learned layers, of which five are convolutional layers and three are fully connected layers, the latter three layers are fully connected, allowing the network to combine features extracted in the previous layers to perform the final classification. The architecture also includes max-pooling, linear rectifiers [47] (ReLUs) as activation functions, and regularization techniques such as dropout [48] and local contrast normalization. As for the network, it takes 224×224 pixel images as input. To increase the amount of training data, data augmentation was used, including cropping and random horizontal reflections of the images.
- **GoogleNet:** GoogLeNet [49] has a depth of 22 layers, but uses fewer parameters compared to other deep networks such as AlexNet. This is achieved by 1×1 convolutions that reduce dimensionality before performing more computationally expensive operations. This makes GoogLeNet more efficient in terms of memory and processing speed. These layers can be convolutional layers, pooling layers, and Inception modules [50], the latter modules being its distinguishing feature. The network also uses normalization and regularization techniques, such as ReLU [47] (Rectified Linear Units) for activation functions, and dropout [48] to prevent overfitting.
- **VGGNet:** The VGGNet [51] architecture consists of 13 convolutional layers which are executed by rectified linear units [47] (ReLU), which introduce essential nonlinearities for multi-layer networks and substantially accelerate the training phase. These are followed by three fully connected layers (FC6, FC7 and FC8) which have a dimension of 4096 units each. The output of the last fully connected layer is fed to a 1000-way softmax [52] layer, which produces probabilities over 1000 object categories. This network uses small 3×3 pixel convolutions, which allow detailed image features to be captured.

- **ResNet:** ResNet [53] is composed of several residual blocks. Each block has some convolutional layers and a shortcut connection that skips those layers. The network starts with a convolutional layer that processes the input image, followed by residual blocks, each of which has convolutional layers and a connection that skips these layers. Finally, it ends with a fully connected layer that performs the final classification.

The key innovation of ResNet is residual connections [54]. Instead of simply stacking convolutional layers one after another, ResNet includes connections that skip some layers and directly connect the input of a layer block to its output. These “shortcut” connections allow information to flow more easily through the network, making training more efficient and allowing the network to be deeper without losing accuracy.

3.3 Analysis and Functioning of the CNN

Following the explanation on how *CNN* models work in the context of malware, the application and potential uses will be explained. *CNN* models are employed for the classification of malware images into malware families, as evidenced by Sharma et al. [55]. Typically, the output of the *CNN* models is in form of images representing the bytes of the malware program, such as greyscale image form or heatmap form. This experimentation is possible due to the usage of different malware image datasets and malware file datasets which can be found on the internet [56]. Malware classification will be beneficial for the identification of malware mutations in malware families and the learning of new forms of malware in order to classify them. Using *CNNs* for this task will be useful for the automation of malware classification.

In order to improve the classification of malware of the *CNN* models, the usage of the aforementioned *GAN* models will be useful for the training of the *CNN* models with newly generated malware [55]. The *GAN* models will create new forms of malware by creating new malware images, which will be beneficial for the training and incorporation of new malware forms from the same family. The combination of both models will be helpful in order to develop an effective malware detection system, enabling the development and automation of classification processes leading to a faster reaction to combat malware.

4 Challenges and Future Directions

The future challenges and projects will be carried out according to the present situation on *GNNs* and *CNNs* analysed on this survey. On future projects, it would be interesting to develop a *GNN* which generates new forms of malware, as well as a *CNN* that could easily categorise malware into families, being this combination of both models the most important idea of this survey.

For future projects, a set of challenges should be taken into account. The use of the existing datasets will be essential for the development of these future scenarios, being these malware datasets not outdated and without containing replicated and excessive information, being this the potential cause for overfitting.

These two aspects are important for future projects as the malware generation and classification should be effective to the present and future contexts.

On the development of a *CNN* model on malware, it is known that classification on adversarial or obfuscated malware is usually bypassed [56]. This is a significant challenge for the training of the GAN model, in which adversarial examples of malware should be considered.

5 Conclusions

This survey presents an analysis on how two types of neural networks work in the context of malware. The survey started with an overview of GAN models which are created for malware generation. Using the PlausMal-GAN model, the malware generation and learning functioning is explained and it was compared with another models. Then, the usage of CNN models in the context of malware is explained. A general overview of the model functioning is provided, and its usage for malware classification is explained.

In the context of CNN studies, the potential use of the combination of both models to develop synthetic malware and strengthen malware classification into families is considered to be the most important point of interest in this survey. Therefore, the combination of both GAN and CNN models in the context of malware is considered in the final section of the survey, which also includes a description of the future projects which will be continued in the near future.

References

1. Nataraj, L., Karthikeyan, S., Jacob, G., Manjunath, B.S.: Malware images: visualization and automatic classification. In: Proceedings of the 8th International Symposium on Visualization for Cyber Security, pp. 1–7 (2011)
2. Wang, K., Gou, C., Duan, Y., Lin, Y., Zheng, X., Wang, F.-Y.: Generative adversarial networks: introduction and outlook. *IEEE/CAA J. Automatica Sinica* **4**(4), 588–598 (2017)
3. Li, Z., Liu, F., Yang, W., Peng, S., Zhou, J.: A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE Trans. Neural Netw. Learn. Syst.* **33**(12), 6999–7019 (2022)
4. Ye, C., et al.: Network deconvolution. In: Conference on Neural Information Processing Systems (NeurIPS), pp. 865–875 (2018)
5. Bjorck, J., Gomes, C., Selman, B., Weinberger, K.Q.: Understanding batch normalization. arXiv preprint [arXiv:1806.02375](https://arxiv.org/abs/1806.02375) (2018)
6. Arora, R., Basu, A., Mianjy, P., Mukherjee, A.: Understanding deep neural networks with rectified linear units. arXiv preprint [arXiv:1806.05341](https://arxiv.org/abs/1806.05341) (2018)
7. Riad, R., Teboul, O., Grangier, D., Zeghidour, N.: Learning strides in convolutional neural networks (2022)
8. Santambrogio, F.: Euclidean, metric, and Wasserstein gradient flows: an overview. *Bull. Math. Sci.* **7**(1), 87–154 (2017)
9. Maniatopoulos, A., Mitianoudis, N.: Learnable leaky ReLU (LeLeLU): an alternative accuracy-optimized activation function. *Information* **12**(12), 513 (2021)

10. Tran, L.D., Nguyen, S.M., Arai, M.: GANbased noise model for denoising real images. In: Proceedings of the Asian Conference on Computer Vision (2020)
11. Weng, L.: From GAN to WGAN. arXiv preprint [arXiv:1904.08994](https://arxiv.org/abs/1904.08994) (2019)
12. Fang, W., Zhang, F., Sheng, V.S., Ding, Y.: A method for improving CNN-based image recognition using DCGAN. Comput. Mater. Continua **57**(1), 167–178 (2018)
13. Zhong, F., Cheng, X., Dongxiao, Yu., Gong, B., Song, S., Jiguo, Yu.: MalFox: camouflaged adversarial malware example generation based on Conv-GANs against black-box detectors. IEEE Trans. Comput. **73**, 980–993 (2023)
14. Zhang, F., Bai, J., Zhang, J., Xiao, Z., Pei, C.: An optimized training method for GAN-based hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. **18**(10), 1791–1795 (2020)
15. Nataraj, L., Karthikeyan, S., Jacob, G., Manjunath, B.: Malware images: visualization and automatic classification, July 2011
16. Kim, J.-Y., Cho, S.-B.: Obfuscated malware detection using deep generative model based on global/local features. Comput. Secur. **112**, 102501 (2021)
17. Vasan, D., Alazab, M., Wassan, S., Safaei, B., Zheng, Q.: Image-based malware classification using ensemble of CNN architectures (IMCEC). Comput. Secur. **92**, 101748 (2020)
18. Berlin, K., Slater, D., Saxe, J.: Malicious behavior detection using windows audit logs. CoRR, abs/1506.04200 (2015)
19. Ndibanje, B., Kim, K.H., Kang, Y.J., Kim, H.H., Kim, T.Y., Lee, H.J.: Cross-method-based analysis and classification of malicious behavior by API calls extraction. Appl. Sci. **9**(2), 239 (2019)
20. Khammas, B.M.: Ransomware detection using random forest technique. ICT Express **6**(4), 325–331 (2020)
21. Agarap, A.F., Pepito, F.J.H.: Towards building an intelligent anti-malware system: a deep learning approach using support vector machine (SVM) for malware classification. CoRR, abs/1801.00318 (2018)
22. Ye, Y., Chen, L., Hou, S., Hardy, W., Li, X.: DeepAM: a heterogeneous deep learning framework for intelligent malware detection. Knowl. Inf. Syst. **54**, 265–285 (2018)
23. Yan, J., Qi, Y., Rao, Q.: Detecting malware with an ensemble method based on deep neural network. Secur. Commun. Netw. **2018**(1), 7247095 (2018)
24. Venkatraman, S., Watters, P.A., Alazab, M., Alazab, M., et al.: Zero-day malware detection based on supervised learning algorithms of API call signatures. AusDM **11**, 171–182 (2011)
25. Gandotra, E., Bansal, D., Sofat, S.: Zero-day malware detection. In: 2016 Sixth International Symposium on Embedded Computing and System Design (ISED), pp. 171–175. IEEE (2016)
26. Kim, J.-Y., Seok-Jun, B., Cho, S.-B.: Zero-day malware detection using transferred generative adversarial networks based on deep autoencoders. Inf. Sci. **460**, 83–102 (2018)
27. Won, D.-O., Jang, Y.-N., Lee, S.-W.: PlausMAL-GAN: plausible malware training based on generative adversarial networks for analogous zero-day malware detection. IEEE Trans. Emerg. Top. Comput. **11**(1), 82–94 (2023)
28. Deldar, F., Abadi, M.: Deep learning for zero-day malware detection and classification: a survey. ACM Comput. Surv. **56**(2), 1–37 (2023)
29. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc. (2017)

30. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks (2016)
31. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: International Conference on Machine Learning, pp. 214–223. PMLR (2017)
32. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of Wasserstein GANs. *Adv. Neural Inf. Process. Syst.* **30** (2017)
33. Mao, X., Li, Q., Xie, H., Lau, R.Y.K., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2794–2802 (2017)
34. Zhao, J., Mathieu, M., LeCun, Y.: Energy-based generative adversarial network. arXiv preprint [arXiv:1609.03126](https://arxiv.org/abs/1609.03126) (2016)
35. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. arXiv preprint [arXiv:1710.10196](https://arxiv.org/abs/1710.10196) (2017)
36. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. arXiv preprint [arXiv:1809.11096](https://arxiv.org/abs/1809.11096) (2018)
37. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4401–4410 (2019)
38. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of styleGAN. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8110–8119 (2020)
39. Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., Aila, T.: Training generative adversarial networks with limited data. *Adv. Neural. Inf. Process. Syst.* **33**, 12104–12114 (2020)
40. Karras, T., et al.: Alias-free generative adversarial networks. *Adv. Neural. Inf. Process. Syst.* **34**, 852–863 (2021)
41. He, K., Kim, D.S.: Malware detection with malware images using deep learning techniques, pp. 95–102 (2019)
42. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition, pp. 346–361 (2015)
43. Kobayashi, T.: Global feature guided local pooling, pp. 123–131 (2019)
44. Mastyło, M.: Bilinear interpolation theorems and applications. *J. Approx. Theory* **172**, 92–107 (2013)
45. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2921–2929 (2016)
46. Chen, H.-C., et al.: AlexNet convolutional neural network for disease detection and classification of tomato leaf. *Electronics* **11**(6), 951 (2022)
47. Daubechies, I., DeVore, R., Foucart, S., Hanin, B., Petrova, G.: Nonlinear approximation and (deep) ReLU networks. *Constr. Approx.* **55**(1), 127–172 (2022)
48. Salehin, I., Kang, D.-K.: A review on dropout regularization approaches for deep neural networks within the scholarly domain. *Electronics* **12**(14), 3106 (2023)
49. Ballester, P., Araujo, R.M.: On the performance of GoogleNet and AlexNet applied to sketches **30**(1), 1124–1128 (2016)
50. Aswathy, P., Mishra, D., et al.: Deep GoogleNet features for visual object tracking. In: 2018 IEEE 13th International Conference on Industrial and Information Systems (ICIIS), pp. 60–66. IEEE (2018)
51. Muhammad, U., Wang, W., Chattha, S.P., Ali, S.: Pre-trained VGGNet architecture for remote-sensing image scene classification, pp. 1622–1627 (2018)

52. Hu, R., Tian, B., Yin, S., Wei, S.: Efficient hardware architecture of softmax layer in deep neural network. In: 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP), pp. 1–5. IEEE (2018)
53. Krichen, M.: Convolutional neural networks: a survey. *Computers* **12**(8), 151 (2023)
54. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31 (2017)
55. Sharma, O., Sharma, A., Kalia, A.: MIGAN: GAN for facilitating malware image synthesis with improved malware classification on novel dataset. *Expert Syst. Appl.* **241**, 122678 (2024)
56. Chen, B., Ren, Z., Chao, Yu., Hussain, I., Liu, J.: Adversarial examples for CNN-based malware detectors. *IEEE Access* **7**, 54360–54371 (2019)



Evaluating Sniffers, IDS, and IPS: A Systematic Literature Mapping

Raúl Bayón-Martínez^(✉), Laura Inyesto-Alonso, Adrián Campazas-Vega,
Gonzalo Esteban-Costales, Claudia Álvarez-Aparicio,
Ángel Manuel Guerrero-Higueras, and Vicente Matellán-Olivera

Grupo de robótica de la universidad de León, MIC building. Campus de Vegazana,
s/n, 24071 León, Spain
{rbaym,linya,acamv,gestc,calvaa,agueh,vmato}@unileon.es

Abstract. The increasing number of cyber threats has created the necessity of using intrusion detection systems, as well as, preventing these threats. These systems enable companies to be alerted to ongoing attacks thanks to the analysis of corporate networks. In this article, a systematic mapping of the literature has been carried out with the objective of identifying the most used sniffers, IDSs and IPSs, and determining which ones offer the best results. A total of 262 articles were reviewed, of which 10 were analysed in depth. The results demonstrated that Snort is the most commonly used IDS in the literature. However, Suricata is the IDS that performs better when the network handles a higher volume of traffic.

Keywords: Detection Methods · Network · IPS · IDS · NIDS · HIDS

1 Introduction

The number of cyberattacks continues to rise year after year. The 2023 Spanish Annual Report on Cybersecurity, reveals a 24% increase in incidents compared to the previous year, with 83,517 cybersecurity incidents managed. Of these, over 22,000 affected to private companies [1]. This increase demonstrates the critical need, for organizations, to invest in cybersecurity.

A fundamental pillar of the cybersecurity systems is intrusion detection. This enables the monitoring of events occurring in a computer system or network and analysing them for signs of possible incidents [2]. Consequently, solutions such as sniffers, intrusion detection systems (IDSs) and intrusion prevention systems (IPSs), become an essential component for the preservation of digital assets, ensuring their integrity, confidentiality and availability. Therefore, it is imperative that the cybersecurity managers assess their system necessities and capabilities in order to determine the most effective tool to defend their digital environment. In light of this, there is a need for a literature review of available solutions to enable these professionals to make evidence-based decisions.

The current literature contains articles that analyse both intrusion detection methods and the existing types of IDSs and IPSs, as well as their functions.

The research proposed by Thakare, Ingle and Meshram [3], analyses the intrusion detection methods and types of IDSs, listing key functionalities of these. In his article [4], Kumar employs a more specific perspective by focusing on network-based IDS (NIDS). It makes a distinction between the signature-based systems and anomaly-based systems, and analyses their respective strengths and weaknesses. Furthermore, the article by Thapa and Mailewa [5], introduces IPSs, describing their importance alongside IDSs in order to protect the systems. Finally, Karthikeyan and Indra [6] examine the characteristics of various IDS tools and the types of attacks they are able to detect.

The preceding paragraph presents a number of research studies that have been conducted with the objective of gathering information about intrusion detection, IDSs and IPSs. It is noteworthy that the aforementioned articles do not address practical questions such as which sniffers, IDSs or IPSs are the most commonly used, nor which configurations yield optimal results, or which one adapts more effectively depending on the network traffic volume. The answers could be of value to several professionals. Moreover, it is important to recognize that the available literature was published prior to 2020, at the time of writing.

The objective of this paper is to conduct a systematic mapping of the literature with the aim of identifying the unknowns in the previous paragraph. In order to achieve this objective, the following research questions are posed:

- RQ1: In the field of the cybersecurity, what are the most commonly used sniffers, IDSs or IPSs in threat detection over the last three years?
- RQ1.1: What settings are most conducive to optimal results for each tool?
- RQ1.2: What tool is suitable depending on the volume of network traffic?

In order to address these questions, the general structure of the PRISMA guide, which facilitates the reporting of systematic reviews and meta-analyses [7], was followed. Section 2 describes the concepts of sniffers, IDSs and IPSs, accompanied by some examples. Section 3 provides a summary of the review protocol employed and its planning. Section 4 presents the findings of the systematic mapping. Section 5 responds to the proposed research questions. Finally, Sect. 6 presents a summary of the conclusions.

2 Context

It is a common occurrence for organizations to attempt to regulate the entry of entities and individuals into their installations, with the aim of determining the purpose of such entry and the reasons behind it. It would be logical to extend this measure to the digital domain. In this context, the sniffers, IDSs and IPSs are employed to fulfil this function. All three systems are designed to protect communication networks, by monitoring traffic entering or leaving a network. However, each of them has features that confer advantages and disadvantages.

A sniffer is an application or a device that can read, monitor, and capture network data exchanges and read network packets [8]. This is of critical importance for system administrators and security engineers, as it enables them to:

troubleshoot network problems, optimise performance and identify anomalous behaviours. Consequently, an IDS is a software that automates the intrusion detection process [2]. It accomplishes this function by employing signatures, heuristics, and behavioural analysis to detect patterns. There are two types of IDSs: network-based (NIDS), which monitors network traffic; and host-based (HIDS), which monitors the system on which it is installed. IPSs automate the same process, with the distinction that they are capable of attempting to prevent potential incidents [2]. IPSs function in conjunction with IDSs, utilising analogous detection techniques. However IPSs are capable of initiating immediate action, blocking malicious or suspicious traffic. Among the most popular IDSs and IPSs are: Suricata, Snort, Zeek, Fireeye and Zscaler.

3 Methodology

This investigation has been conducted following the methodology proposed by Kitchenham, Budgen and Brereton [9], which outlines a series of guidelines for conducting systematic reviews in the field of Software Engineering. The objective of this document is to create a systematic mapping of the most commonly used sniffers, IDSs and IPSs in the cybersecurity field. Thus, this study is based on pair-reviewed articles that compare the features of various sniffers, IDSs or IPSs, or that conduct some type of study or reference to threat or anomaly detection.

3.1 Search Process

The process commenced with a preliminary literature search related to the research questions posed. Despite this search, as previously stated in Sect. 1, no conclusive evidence was found to answer those questions.

Therefore, an automatic search was planned in the following bibliography references and citation databases: IEEE Digital Library, Scopus and Web of Science. In each database, a search was carried out, filtering for articles that belong to a journal or conference, written in English, between January 2020 and September 2023. This filter was also refined by specifying terms in the title, abstract, and keyword sections.

Additionally, a search string that could answer the posed questions was created. Its construction was based on the use of keywords extracted after applying the PICOC strategy [10,11]. In this way, the following search string was obtained: (*“Intrusion detection” OR “Anomaly detection” OR “Anomaly detection systems” OR “Intrusion detection systems”*) AND (*sniffer OR IDS OR IPS*) AND (*Nmap OR Suricata OR Snort OR Zeek OR Fireeye OR Zscaler*).

3.2 Selection Process

The results of the search string were subjected to a series of phases with the objective of filtering the candidate evidence and obtaining the relevant ones for

Table 1. Inclusion and exclusion criteria

Type	ID	Description
Inclusion	IC1	Contains information about any sniffer, IDS o IPS
	IC2	Compares the characteristics of various sniffers, IDS o IPS
	IC3	It refers to threat or anomaly detection, or a study on the subject is being carried out
	IC4	Uses objective metrics, such as detection methods, license types, performance, support, etc.
Exclusion	EC1	It does not belong to cybersecurity field
	EC2	The research field of the article is mobile technologies
	EC3	The research field of the article is industrial settings

the research questions. To achieve this, a set of inclusion and exclusion criteria, specified in Table 1, was established.

The fundamental inclusion criteria are: if the article's text compares the characteristics of various sniffers, IDSs or IPSs; or if a study or reference to the detection of threats or anomalies is carried out. With regard to the exclusion criteria, the fields of use have been taken into account, discarding those articles that do not belong to the cybersecurity field or that are related to mobile technologies or industrial settings.

The selection process was conducted in several stages. Initially, the duplicated articles were removed. Secondly, a preliminary screening was performed by reading the title, *abstract* and keywords of each selected article, taking into account the previous criteria. Finally, copies of the remaining articles were obtained for a thorough reading and review of their content, proceeding to their elimination if they did not meet all criteria.

3.3 Extraction Process and Data Synthesis

The data extraction process was conducted electronically via the online tool Parsif.al [12]. To complete it, a specific form was defined to obtain the relevant data from each article.

The extracted data, which includes information such as the tool name, type of license, detection methods, etc., have been summarized in Table 2 to graphically display the data obtained.

Table 2. Form fields

Extracted data	Description
Name	Name of the used tool
Developer	Name of the developer of the tool
License	Type of license
Supported operative systems	OS in which can be deployed
Installation	Installation complexity
IPS	If it has the capability of an IPS
Detection methods	Methods used for threat detection
Configuration GUI	If it has a configuration Graphical User Interface
High-speed network capability	Capacity level for high-speed networks

In order to present the extracted information, the data will be synthesised into two tables. The first one will be a synthesis of the extracted data, divided by each tool. The second table will show a comparison of the IDSs, IPSs and sniffers referenced in the articles, presenting the name of the tool along with the associated data at CPU, RAM and extra information level.

4 Results

Figure 1 provides a summary of the application of the methodology described in Sect. 3. After applying the search string to the three selected databases, a total of 262 articles were obtained. Of these, 110 articles were removed due to duplication. The remaining 152 articles were reviewed. Of these 152 articles, 134 were excluded based on the title and abstract. Of the 18 remaining articles, 8 were discarded due to meeting any exclusion criteria. After applying the criteria specified in this study, the following 10 articles were finally selected for further analysis [13–22].

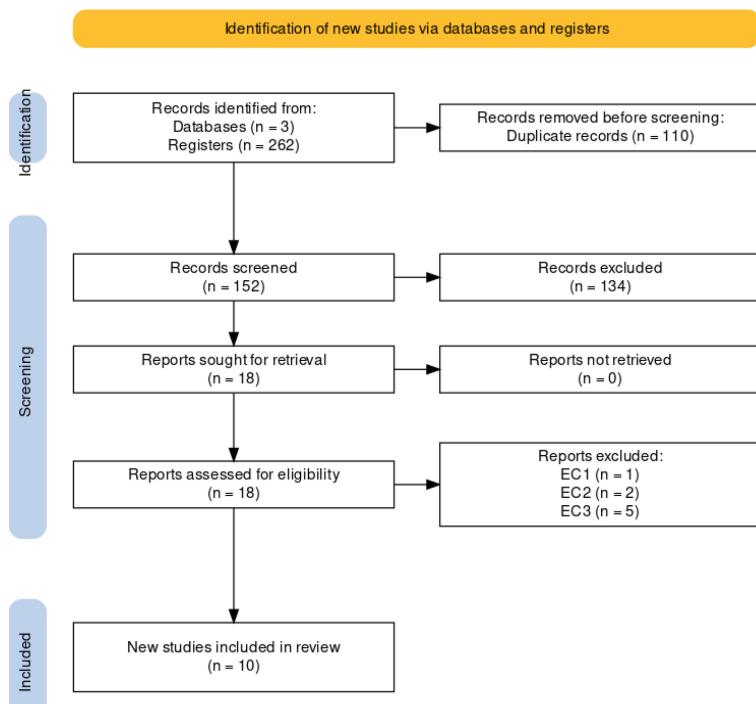


Fig. 1. PRISMA diagram.

Figure 2 illustrates the number of accepted articles per year. As can be observed, the year with the most accepted publications was 2022.

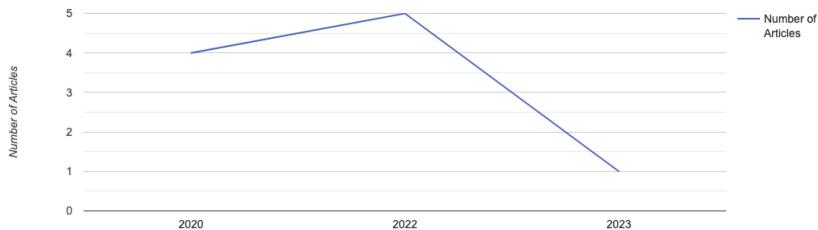


Fig. 2. Number of articles accepted per year.

5 Discussion

5.1 Research Question RQ1

The objective of research question RQ1 is to find the most commonly used sniffers, IDS or IPS in threat detection. The set of tools found are Snort, Suricata, Zeek, Bro IDS, Security Onion and OSSEC. The most popular and widely used according to the reviewed literature are Suricata and Snort. To answer this question, it has been broken down into two more specific questions, with the aim of obtaining a more concrete and specific response.

Research question RQ1.1. The aim of question RQ1.1 is to find which configuration is the most optimal for achieving the most effective results with each tool. In most of the selected articles, we can find the configurations used along with an analysis of the performance of the tool. The percentage of articles talking about configuration is 50%. The articles talking about this are [14–16, 21, 22].

Research question RQ1.2. The purpose of question RQ1.2 is to determine which tool offers better performance in terms of network volume. As observed in the conclusions of article [14], Suricata performs better than the others when exposed to high volume networks. Regarding lower volume networks, Suricata generally offers better performance, but it depends on the software version. The percentage of articles talking about traffic volume is 50%. The articles talking about this are [13, 14, 16, 18, 21].

The characteristics of the selected articles are summarized below.

In the work carried out in [13], the authors propose an IP flow-based IDS to monitor and protect IoT networks from threats. The proposed framework collects IP flows from an IoT network and analyses them to monitor and detect attacks, intrusions, and other types of anomalies based on certain flow characteristics. The authors simulated the network's operation in various scenarios, with measurements based on a 50 Mbps bandwidth probe. As an experiment, they compared their proposal with the leading IDSs on the market (Snort, Suricata and Zeek), demonstrating that their solution consumes fewer resources (CPU, RAM and persistent memory).

In [14], the authors compare the performance in terms of detection and resource consumption of the IDSs Suricata and Snort on 100Gb/s networks. Specifically, the authors studied system resource usage, packet processing speed, packet loss rate, and detection accuracy. The authors also provided an exhaustive analysis to show the impact on the performance of the IDSs using different configurations and traffic volumes. In the first experiment, the authors used Iperf3 to generate multiple TCP flows and two different libraries, Libpcap and AF_PACKET, to capture packets. In the second experiment, the authors used Iperf3 and the tool Pytbull to generate malicious traffic. Both IDSs used the default rule set and configuration. In the third experiment, they used real traffic templates generated by the tool TRex. Packets with different sizes and protocols were used. As a conclusion, they demonstrated that Suricata outperforms Snort in terms of performance when handling high traffic volumes, showing that the best combination is Suricata using the AF_Packet library.

In the study [15], the authors propose an experimental comparison between the most used open-source NIDS, namely, Snort, Suricata and Bro IDS. The objective is to find the best NIDS for smart homes in terms of detection accuracy and resource usage, including CPU and memory usage. The authors used Snort 3.0, Suricata 3.0.1 and ID Bro 2.5. The experiments were conducted on an Ubuntu 16.0.x virtual machine with 8 GB of RAM, a 40 GB hard drive and a 2,6 GHz Intel Xeon CPU E5-2650 v2. The authors used different pcaps with network attacks.

In the investigation carried out in [16] a performance comparison between Suricata 4.1.2 and Snort 2.9.12 (Single-Thread) is conducted. The authors used different packet sizes (512, 1024 and 1400 bytes) and traffic thresholds of 10,000, 20,000, 30,000, 40,000 and 50,000. They concluded that Suricata offers better results at each traffic threshold and for each packet size than Snort in terms of the number of packets it is capable of analysing per second and RAM usage. In terms of CPU usage, Snort offered lower usage.

In article [17], the authors evaluated the behaviour of the IDS Snort against fake network packets generated with the tool Scapy. Among the various attacks conducted in the experiments, we find fake packet injection attacks, DoS attacks and MITM attacks. In this study, the authors used whitelists and blacklists in Snort as a traffic filtering technique.

In the study conducted in [18], the authors analysed two open-source IDSs, Snort IDS and Zeek IDS, to evaluate their performance based on various parameters, such as detection rate, packet loss rate and delay. The authors used different types of denial of service attacks and port scans to test the IDSs. The results demonstrated that Snort shows a better detection rate regardless of the traffic volume and a better packet elimination rate, but a higher delay.

The authors of [19] evaluated the impact of five popular IDSs in terms of their efficiency and effectiveness in information security. These IDSs are Zeek, Suricata, Security Onion, OSSEC and Snort. The authors used the Analytic Hierarchy Process (AHP) and the fuzzy Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS), based on the integrated methodology of

Multiple-Criteria Decision Making (MCDM), to evaluate the efficiency of the most popular IDSs. As a conclusion, Suricata offers significant advantages over Snort.

The objective in [20] is to carry out tests on IDS Snort 3.0 and IDS Suricata. This investigation was conducted on a virtual machine with 1-core, 2-core and 4-core processors. The attack scenario is divided into two parts: DoS attack scenario using 1 physical computer, and a DDoS attack scenario using 5 physical computers. The study concluded that in terms of resource usage, Suricata demonstrated better results than Snort.

In [21], the authors analysed three open-source Intrusion Detection and Prevention Systems (IDPS): Snort (in both its single-thread and multi-thread versions), Suricata and Zeek, using performance parameters similar to those normally used for evaluating commercial IDPS solutions. The authors evaluated the typical deployment of Network-based Intrusion Detection and Prevention Systems (NIDPS) in Small and Medium-sized Enterprises (SMEs) and assessed the system using various factors, such as the packet capture modules, the detection engine algorithms, the packet size, and the rule set size. In the experiments, they used the Libpcap and AF_PACKET libraries. Subsequently, the authors conducted tests with different packet sizes, specifically 512, 1024 and 1538 bytes. The authors compared the Aho-Corasick, Boyer Moore, and Wu Manber pattern matching algorithms. The results demonstrated that the best algorithm was the ac-split version of Aho-Corasick. Both Suricata and Snort have this algorithm preconfigured by default.

In the study carried out in [22], the authors presented an analysis of the variability of the open-source IDS, Snort and Suricata, to help security architects adjust/deploy these IDSs. They analysed two types of variability in these IDSs: configuration and functionality. In the analysis of the configuration diversity, they investigated the diversity in the rule sets and the Blacklisted IP Addresses (BIPA) that these IDSs use in their configurations. The analysis of functional diversity investigates the difference in the alert behaviours of these IDSs while analysing real network traffic, and how these differences evolve. In terms of rules, Snort has two rule sets: the Community Rules Set and the Snort Subscriber Rule

Table 3. Characteristics of IDS

Characteristics	Snort	Suricata	Zeek	Bro IDS
Developer	Cisco	OISF	ICSI	Vern Paxson
License	GNU GPL license	GNU GPL license	BSD license	BSD license
Supported Operative Systems	Win/Unix/Mac	Win/Unix/Mac	Unix/Mac	Unix
Installation	Easy	Intermediate	Typical	Typical
IPS	Si	Si	No	No
Supported traffic	IPv4/IPv6	IPv4/IPv6	IPv4	IPv4
Detection methods	SIDS, AIDS	SIDS, AIDS	Anomalies detection	SIDS, AIDS
Configuration GUI	Si	Si	No	No
High-speed networks capability	Mid	High	High	High

Table 4. Comparison of IDS/IPS/sniffer used in the articles

Article	IDS/IPS/sniffer	CPU	RAM	Extra
[13]	Snort	(19%)	150 Mbps	Processes (132), disk (3 GB),
	Suricata	(16%)	350 Mbps	Processes (138), disk (3.1 GB).
	Zeek	(14%)	400 Mbps	Processes (133), disk (3.2 GB)
[14]	Snort	(46%-40%)	(0.1%-0.2%)	(10GBs): Libpcap and AF_Packet; packets lost (0%-0%)
	Suricata	(30%-10%)	(0.2%-0.1%)	(10 GBs): Libpcap and AF_Packet; packets lost (0%-0%)
	Snort	(96%)	(10%)	(60GBs): AF_Packet; packets lost (0.1%)
	Suricata	(60%)	(10%)	(60GBs): AF_Packet; packets lost (0.1%)
[15]	Snort 3.0	(60%-70%)	(60%-80%)	
	Suricata 3.0.1	(20%-35%)	(20%-40%)	
	Bro IDS 2.5	(20%-40%)	(20%-34%)	
[16]	Snort	(40%)	(14.7%)	
	Suricata		(11%)	Better packet analysis (15% more)
[17]	Snort			100% accuracy on whitelists 90% of packets blocked by blacklists
	Zeek			
[18]	Snort			Better detection rate (15% more) and lower packet loss (10% less) than Zeek
	Zeek			Lower delay (7% less) than Snort
[19]	Suricata/Snort			Accuracy (0.658)/(0.469)
	Security Onion			Accuracy (0.4585)
	OSSEC/Zeek			Accuracy (0.446)/(0.395)
[20]	Snort	(99% 1 core, 58% 4 cores)	(+600MB from 1 to 4 cores)	Packet loss (99.43%)
	Suricata	(40% 1 core, 30% 4 cores)	(+1000MB from 1 to 4 cores)	Packet loss (7.37%)
[21]	Snort 2.9			Libpcap losses packets at 200 MBps AF_PACKET does not lose packets at 1000 MBps
	Snort 3.1.7	(55%)	(28.7%)	AF_PACKET does not lose packets at 1000 MBps With packets of sizes 1538/1024/512B, loses 0%/0%/49.5% packets. As IPS does not detect 64.8% of the packets
	Suricata	(53%)	(29.74%)	Libpcap at 200MBps starts to lose AF_Packet does not lose packets at 1000 MBps. With packet sizes 1538/1024/512B, lose 0%/ 0%/34.75% packets. As IPS does not detect 25.87% of the packets
	Zeek			Does not accept AF_Packet, but does not lose packets at 1000 MBps. With packet sizes 1538/ 1024/512B, loses 0%/20.7%/60.35% packets
[22]	Snort			With 16753 rules, 46701 Blacklisted IPs.
	Suricata			With 19061 rules, 135791 Blacklisted IPs

Set, which is a paid solution for companies. In terms of the number of rules, Suricata had 19,061 in the experiment, whereas the free version of Snort had 8,416 and the paid version had 8,337.

In Table 3, a summary of the characteristics of the systems mentioned in the previous articles can be observed.

On Table 4, you can see together which software has been reviewed in which articles and what their main characteristics are.

6 Conclusions

Sniffers, IDSs and IPSs are highly useful tools for the purpose of securing computing environments. To address the research questions posed in Sect. 3, a systematic mapping of the literature was conducted, analysing 262 articles in its initial phase. From the initial 262 articles, and after applying various quality criteria, 10 articles were analysed in depth.

The results obtained demonstrated that the most used IDS in the literature is Snort, present in 9 out of the 10 articles.

In terms of the network packet collection carried out by the IDSs for subsequent analysis, it has been concluded that the AF_PACKET library offers significantly higher performance than the Libpcap library, greatly limiting the number of lost packets when handling large amounts of network traffic. It is worth nothing that both Snort and Suricata are compatible with both libraries.

Finally, in terms of resource consumption of the different IDSs, both Suricata and Zeek offer better results than Snort. Similarly, Suricata demonstrated a higher capacity than Snort when handling high-volume network traffic. Therefore, Suricata becomes the best choice for monitoring networks that handle a large amount of traffic.

Acknowledgements. This publication is part of the CIBERLAB and DEITREAL projects, financed “by European Union NextGeneration-EU, the Recovery Plan, Transformation and Resilience, through INCIBE”; and part of the project EDMAR, PID2021-126592OB-C21, funded by MCIN/AEI/10.13039/501100011033 and by ERDF A way of making Europe”.

References

1. INCIBE. Balance de ciberseguridad 2023 (2024). Accessed 10 May 2024
2. Peter, M., Scarfone, K.: Intrusion detection and prevention systems. In: 2010 Handbook of Information and Communication Security, pp. 177–192. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-04117-4_9
3. Thakare, S., Ingle, P., Meshram, B.B.: IDS: intrusion detection system the survey of information security. Int. J. Emerg. Technol. Adv. Eng. **2**(8), 86–90 (2012)
4. Kumar, S.: Survey of current network intrusion detection techniques (2007)
5. Thapa, S., Mailewa, A.: The role of intrusion detection/prevention systems in modern computer networks: a review (2020)
6. Kr, K., Indra, A.: Intrusion detection tools and techniques—a survey. Int. J. Comput. Theory Eng. **2**(6), 901 (2010)
7. Moher, D., Liberati, A., Tetzlaff, J., Altman, D.G., Group*, P.: Preferred reporting items for systematic reviews and meta-analyses: the prisma statement. Ann. Internal Med. **151**(4), 264–269 (2009)
8. Sinha, U., Hadi, A.A., Faika, T., Kim, T.: Blockchain-based communication and data security framework for iot-enabled micro solar inverters. In: 2019 IEEE CyberPELS (CyberPELS), pp. 1–5 (2019)
9. Kitchenham, B.A., Budgen, D., Brereton, P.: Evidence-Based Software Engineering and Systematic Reviews, vol. 4. CRC Press, Boca Raton (2015)

10. Roehrs, A., Da Costa, C.A., da Rosa Righi, R., De Oliveira, K.S.F.: Personal health records: a systematic literature review. *J. Med. Internet Res.* **19**(1), e5876 (2017)
11. Schardt, C., Adams, M.B., Owens, T., Keitz, S., Fontelo, P.: Utilization of the pico framework to improve searching pubmed for clinical questions. *BMC Med. Inform. Decis. Mak.* **7**(1), 1–6 (2007)
12. Parsifal (2023). Accessed 27 Sept 2023
13. Santos, L., Gonçalves, R., Rabadao, C., Martins, J.: A flow-based intrusion detection framework for internet of things networks. *Cluster Comput.* 1–21 (2021)
14. Hu, Q., Yu, S.-Y., Asghar, M.R.: Analysing performance issues of open-source intrusion detection systems in high-speed networks. *J. Inf. Secur. Appl.* **51**, 102426 (2020)
15. Alsakran, F., Bendiab, G., Shiaeles, S., Kolokotronis, N.: Intrusion detection systems for smart home IoT devices: experimental comparison study. In: Thampi, S.M., Martinez Perez, G., Ko, R., Rawat, D.B. (eds.) *SSCC 2019. CCIS*, vol. 1208, pp. 87–98. Springer, Singapore (2019). https://doi.org/10.1007/978-981-15-4825-3_7
16. Gupta, A., Sharma, L.S.: Performance evaluation of *Snort* and *Suricata* intrusion detection systems on ubuntu server. In: Singh, P.K., Kar, A.K., Singh, Y., Kolekar, M.H., Tanwar, S. (eds.) *Proceedings of ICRIC 2019. LNEE*, vol. 597, pp. 811–821. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-29407-6_58
17. Pansari, N., Saiya, R.: Fake packet generation, detection and its analysis using network security. In: *2022 IEEE 7th International conference for Convergence in Technology (I2CT)*, pp. 1–9. IEEE (2022)
18. Niknami, N., Inkrott, E., Wu, J.: Towards analysis of the performance of idss in software-defined networks. In: *2022 IEEE 19th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*, pp. 787–793. IEEE (2022)
19. Alyami, H., et al.: Effectiveness evaluation of different idss using integrated fuzzy mcdm model. *Electronics* **11**(6), 859 (2022)
20. Fadhilah, D., Marzuki, M.I.: Performance analysis of ids snort and ids suricata with many-core processor in virtual machines against dos/ddos attacks. In: *2020 2nd International Conference on Broadband Communications, Wireless Sensors and Powering (BCWSP)*, pp. 157–162. IEEE (2020)
21. Waleed, A., Jamali, A.F., Masood, A.: Which open-source ids? snort, suricata or zeek. *Comput. Netw.* **213**, 109116 (2022)
22. Asad, H., Gashi, I.: Dynamical analysis of diversity in rule-based open source network intrusion detection systems. *Empir. Softw. Eng.* **27**, 1–30 (2022)



Performance Analysis of NTT Algorithms

David García Lleyda¹, Víctor Gayoso Martínez^{1(✉)}, Luis Hernández Encinas², Agustín Martín Muñoz², and Óscar Castillo Campo³

¹ Centro Universitario de Tecnología y Arte Digital (U-tad), Data Research and Computation Group (DRACO), 28290 Las Rozas de Madrid, Spain

david.lleyda@live.u-tad.com, victor.gayoso@u-tad.com

² Institute of Physical and Information Technologies (ITEFI), Spanish National Research Council (CSIC), 28006 Madrid, Spain

{luis.h.encinas, agustin.martin}@csic.es

³ Department of Industrial Engineering, Universidad Nebrija, 28015 Madrid, Spain
ocastillo@nebrija.es

Abstract. The Number Theoretic Transform is an important tool in Cryptography, as it can be used to multiply polynomials in the rings $\mathbb{Z}_q[x]/\langle x^n - 1 \rangle$ and $\mathbb{Z}_q[x]/\langle x^n + 1 \rangle$. In its plain version, the NTT is slower than the direct multiplication method for these rings. To overcome this disadvantage, some optimised variants of the NTT have been proposed in recent years. This contribution analyses one of these variants and presents an empirical comparison of the running time for the three methods considered in the study, which allows to determine the length of the polynomials for which each method is the fastest.

Keywords: Number Theoretic Transform · post-quantum cryptography · polynomial multiplication · quotient rings · software implementation

1 Introduction

The Number Theoretic Transform (NTT) is a mathematical tool particularly well known in the field of Cryptography, more specifically in the areas of Post-Quantum Cryptography (PQC) and Homomorphic Encryption (HE). In PQC, it is used intensively by lattice-based algorithms for the multiplication of polynomials [8, 10].

In Mathematics, a ring is an algebraic structure that consists of a set of elements equipped with two binary operations, typically referred to as addition and multiplication, which satisfy certain properties. A ring over a finite field is a ring where the elements belong to a finite field such as $\mathbb{F}_q = \{[0], [1], \dots, [q-1]\}$. In \mathbb{Z}_q , the elements of the field are the residue classes of \mathbb{Z} . Given $a \in \mathbb{Z}$, the residue class $[a]_q$ (or simply $[a]$) contains all the integer numbers that are congruent to each other modulo q , i.e., the integers that have the same remainder when divided by the modulus q .

The NTT is very closely related to the Discrete Fourier Transform (DFT) but operates using modular arithmetic over finite rings. The NTT represents polynomials as vectors, so polynomial multiplications can be calculated alternatively by combinations of direct and inverse transformations and point-to-point operations.

Compared to traditional polynomial multiplication methods, optimised variants of the NTT can offer significant advantages in terms of efficiency and scalability. However, the threshold at which the NTT is faster than traditional algorithms is not clear in the scientific literature.

In this article, a detailed description of the NTT algorithm and a recursive variant is presented. In addition to that, this contribution includes a performance comparison of the algorithms considered in the study using an application developed in the Java programming language, so it can be checked how the theoretical performance advantages translate into software implementations.

There are several works related to this contribution. Anna Bakkebø's master thesis [2], published in 2020, describes the “divide and conquer” technique for the NTT that is used in this study. She also developed C code to test the performance of the implementation. In comparison to that work, in this contribution we have not only analysed the ring $\mathbb{Z}_q[x]/\langle x^n + 1 \rangle$, but also $\mathbb{Z}_q[x]/\langle x^n - 1 \rangle$, which requires fewer multiplications. In addition to that, in this work we have included the plain version of the NTT in the comparison and have taken into account the initialisation time for the NTT, not only the multiplication time.

Zhichuang Liang and Yunlei Zhao published a survey of the NTT in 2022 [6]. This survey mentions several algorithms and tricks for improving the performance of the NTT, including the so-called Cooley-Tukey and Gentleman-Sande techniques [7]. These techniques are closely related to the one used by Bakkebø, and have been described in prior works (e.g., in [3]). However, only the mathematical description of the techniques is included in [6], as it is a theoretical survey.

Satriawan et al. have recently published a complete introductory guide for the NTT [9]. In their article, they describe the Cooley-Tukey and the Gentleman-Sande techniques. As mentioned above, our contribution is based more directly on Bakkebø's recursive design than on the algorithms described in [9], although they are similar. In addition, the authors of [9] did not include a software performance comparison.

The rest of this article is organised as follows: Sect. 2 provides an introduction to the direct multiplication method in the rings $\mathbb{Z}_q[x]/\langle x^n - 1 \rangle$ and $\mathbb{Z}_q[x]/\langle x^n + 1 \rangle$, where $x^n - 1$ and $x^n + 1$ are called the cyclotomic and anticyclotomic polynomials, respectively. Section 3 explains the plain version of the NTT and the optimised version described in [2]. After that, Sect. 4 presents the results of the performance comparison using the Java application developed for this study. Finally, conclusions are summarised in Sect. 5.

2 Polynomial Multiplication in the Ring $\mathbb{Z}_q[x]/\langle f(x) \rangle$

Given a prime number $q \in \mathbb{Z}$, the polynomial ring $\mathbb{Z}_q[x]$ is the set of polynomials with coefficients in the finite field \mathbb{Z}_q . In this context, $\mathbb{Z}_q[x]/\langle f(x) \rangle$ represents the quotient ring of $\mathbb{Z}_q[x]$, which consists of the residue classes of all polynomials that are congruent to each other modulo $f(x)$, i.e., the ring of the polynomials with coefficients in \mathbb{Z}_q and degree, at most, $\deg(f(x)) - 1$. The size of the quotient ring $\mathbb{Z}_q[x]/\langle f(x) \rangle$ is equal to the degree of $f(x)$. This means that there are $q^{\deg(f(x))}$ elements in the ring.

If $f(x)$ is irreducible (i.e., it cannot be factored into the product of two or more polynomials of smaller degree), then the quotient ring $\mathbb{Z}_q[x]/\langle f(x) \rangle$ is a finite field. In comparison, if $f(x)$ is not irreducible, then the quotient ring is instead a ring with zero divisors.

This contribution focuses on two polynomial rings, namely $\mathbb{Z}_q[x]/\langle x^n - 1 \rangle$ and $\mathbb{Z}_q[x]/\langle x^n + 1 \rangle$, where $n = 2^m$ for some $m \in \mathbb{N}$. These rings, specially the second one, are commonly used in lattice-based cryptography by algorithms such as NewHope [1] or Kyber [4].

Given $a(x) = a_0 + a_1 x + \dots + a_{n-1} x^{n-1}$ and $b(x) = b_0 + b_1 x + \dots + b_{n-1} x^{n-1}$, polynomial addition in $\mathbb{Z}_q[x]/\langle f(x) \rangle$ is a straightforward operation, since the coefficients of $a(x) + b(x)$ are simply computed by adding them modulo q , which is a very fast operation from a computational point of view:

$$a(x) + b(x) = \sum_{i=0}^{n-1} ((a_i + b_i)(\text{mod } q)) x^i.$$

In contrast, polynomial multiplication is more difficult to perform, as after the regular multiplication of $a(x)$ and $b(x)$ it is necessary not only to reduce the coefficients modulo q , but also to calculate the remainder of the polynomial when divided by $f(x)$.

If we consider $f(x) = x^n - 1$, given that $x^{k+n} \equiv x^k \pmod{x^n - 1}$, the multiplication of the polynomials $a(x)$ and $b(x)$ in $\mathbb{Z}_q[x]/\langle x^n - 1 \rangle$ can be represented by the following formula:

$$a(x) \cdot b(x) = \left(\sum_{i=0}^{n-1} \sum_{j=0}^{n-1-i} a_i \cdot b_j \cdot x^{i+j} + \sum_{i=1}^{n-1} \sum_{j=n-i}^{n-1} a_i \cdot b_j \cdot x^{i+j-n} \right) \pmod{q}.$$

On the other hand, if $f(x) = x^n + 1$, as $x^{k+n} \equiv -x^k \pmod{x^n + 1}$, the multiplication of the polynomials $a(x)$ and $b(x)$ in $\mathbb{Z}_q[x]/\langle x^n + 1 \rangle$ can be represented by this formula:

$$a(x) \cdot b(x) = \left(\sum_{i=0}^{n-1} \sum_{j=0}^{n-1-i} a_i \cdot b_j \cdot x^{i+j} - \sum_{i=1}^{n-1} \sum_{j=n-i}^{n-1} a_i \cdot b_j \cdot x^{i+j-n} \right) \pmod{q}.$$

For polynomials with n coefficients, it is clear from the previous formulas that $n \times n$ multiplications need to be performed, so the complexity of the direct multiplication method is $O(n^2)$.

3 NTT

The NTT algorithm allows to transform finite sequences into other finite sequences. In the context of this study, two versions of the NTT are used: the *regular* NTT (also known as Positive-Wrapped NTT or PW-NTT in the scientific literature) and the *negacyclic* NTT (also known as Negative-Wrapped NTT or NW-NTT). The inverse transformations will be referred to as regular INTT and negacyclic INTT, respectively.

In what follows, we will assume that n is a power of 2, so $n = 2^m$ for some $m \in \mathbb{N}$. This is an assumption usually contemplated in the definition of cryptographic algorithms, and allows to include additional improvements in some computations.

3.1 Regular NTT

Let q be a prime number such that $q \equiv 1 \pmod{n}$, and let ω be a primitive root of order n in \mathbb{Z}_q , such that $\omega^n \equiv 1 \pmod{q}$. The regular NTT of the polynomial $a(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1} \sim [a_0, a_1, \dots, a_{n-1}] \in \mathbb{Z}_q^n$ over the ring $\mathbb{Z}_q[x] / \langle x^n - 1 \rangle$ is denoted as $\tilde{a}(x) = \text{NTT}(a(x))$, where the coefficients of $\tilde{a}(x)$ are computed in the following way [9]:

$$\tilde{a}_i = \sum_{j=0}^{n-1} a_j \omega^{ij} \pmod{q}, \quad \forall i \in \{0, 1, \dots, n-1\}.$$

In the regular INTT, where $a(x) = \text{INTT}(\tilde{a}(x))$, the coefficients of the resulting polynomial are calculated as follows:

$$a_i = \frac{1}{n} \sum_{j=0}^{n-1} \tilde{a}_j \omega^{-ij} \pmod{q}, \quad \forall i \in \{0, 1, \dots, n-1\}.$$

Using the NTT, polynomial multiplication in the ring $\mathbb{Z}_q[X] / \langle x^n - 1 \rangle$ can be achieved in this way, where \circ represent the point-to-point multiplication of two vectors:

$$a(x) \cdot b(x) = \text{INTT}(\text{NTT}(a) \circ \text{NTT}(b)).$$

Before performing the multiplication, it is necessary to compute two matrices of $n \times n$ elements with the powers of ω and ω^{-1} , respectively. In principle, n^2 multiplications should be performed to obtain each matrix, but in practice the number is lower because the elements in both the first row and the first column are always 1. Furthermore, some symmetries in the powers of both elements can be exploited in order to reduce the number of multiplications.

Regarding the complexity of the polynomial multiplication itself, it is clear that in the regular NTT n^2 multiplications are needed. If in the regular INTT a double matrix with the products $n^{-1}\omega^{-ij}$ is used instead of just ω^{-ij} (which

adds some cost to the precomputation phase), then again n^2 multiplications are required.

In total, $3n^2$ operations would be needed for the whole process, which means that the complexity of the regular NTT is $O(n^2)$.

3.2 Negacyclic NTT

Let q be a prime number such that $q \equiv 1 \pmod{2n}$, and let ω be a primitive root of order n in \mathbb{Z}_q . In addition to that, let ψ be a primitive root of order $2n$ in \mathbb{Z}_q , such that $\psi^2 = \omega$. If $a(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1} \sim [a_0, a_1, \dots, a_{n-1}] \in \mathbb{Z}_q^n$, then let us define $\hat{a} = [a_0, \psi a_1, \dots, \psi^{n-1} a_{n-1}]$. With these elements, the coefficients of the negacyclic NTT of the polynomial $a(x)$ over the ring $\mathbb{Z}_q[x] / \langle x^n + 1 \rangle$ are computed as follows [7]:

$$\tilde{a}_i = \sum_{j=0}^{n-1} \hat{a}_j \omega^{ij} \pmod{q}, \quad \forall i \in \{0, 1, \dots, n-1\}.$$

Regarding the inverse operation, the coefficients of $a(x) = \text{INTT}(\hat{a}(x))$ are computed in this way:

$$\hat{a}_i = \frac{1}{n} \sum_{j=0}^{n-1} \tilde{a}_j \omega^{-ij} \pmod{q}, \quad \forall i \in \{0, 1, \dots, n-1\}.$$

After that operation, in order to obtain the coefficients of the polynomial $a(x)$ it is necessary to consider that $a(x) = [\hat{a}_0, \psi^{-1} \hat{a}_1, \dots, (\psi^{-1})^{n-1} \hat{a}_{n-1}]$.

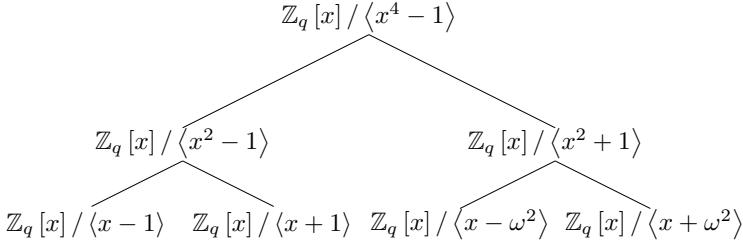
Once again, the multiplication of polynomials in the ring $\mathbb{Z}_q[X] / \langle x^n - 1 \rangle$ can be achieved in this way:

$$a(x) \cdot b(x) = \text{INTT}(\text{NTT}(a) \circ \text{NTT}(b)).$$

Similar considerations can be made to calculate the complexity of the whole process, although in this case the multiplications by the powers of ψ and ψ^{-1} require $2n$ additional multiplications. However, since $2n$ is negligible compared to n^2 for large values of n , the overall complexity of the negacyclic NTT remains $O(n^2)$.

3.3 Optimized NTT

In 2001, Daniel Bernstein showed that it is possible to apply a variant of the Cooley-Tuckey and Gentleman-Sande techniques, typically associated with the Fast Fourier Transform (FFT), to the NTT [3]. By applying these techniques, the time complexity of the NTT/INTT can be reduced from $O(n^2)$ to $O(n \log_2(n))$. This is possible because of the special form of $x^n - 1$ and $x^n + 1$, where n is a power of two. For example, if $n = 4$, ω would be a primitive root of order 4, ω^2 would be a primitive root of order 2, etc., so the polynomial could be decomposed as follows:



Following with the $\mathbb{Z}_q[X]/\langle x^n - 1 \rangle$ example, the double matrix with the powers of ω would have the following aspect:

$$\begin{pmatrix} \omega^0 & \omega^0 & \omega^0 & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \omega^3 \\ \omega^0 & \omega^2 & \omega^4 & \omega^6 \\ \omega^0 & \omega^3 & \omega^6 & \omega^9 \end{pmatrix} \equiv \begin{pmatrix} \omega^0 & \omega^0 & \omega^0 & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \omega^3 \\ \omega^0 & \omega^2 & \omega^0 & \omega^2 \\ \omega^0 & \omega^3 & \omega^2 & \omega^3 \end{pmatrix}$$

With this matrix in mind, the equations to obtain the coefficients of $\tilde{a}(x)$ are the following ones:

$$\begin{aligned} \tilde{a}_0 &= \omega^0 a_0 + \omega^0 a_1 + \omega^0 a_2 + \omega^0 a_3 \\ \tilde{a}_1 &= \omega^0 a_0 + \omega^1 a_1 + \omega^2 a_2 + \omega^3 a_3 \\ \tilde{a}_2 &= \omega^0 a_0 + \omega^2 a_1 + \omega^0 a_2 + \omega^2 a_3 \\ \tilde{a}_3 &= \omega^0 a_0 + \omega^3 a_1 + \omega^2 a_2 + \omega^3 a_3 \end{aligned}$$

However, taking into account that $\omega^2 = -1$ and $\omega^3 = -\omega$, the previous computations can be arranged in a more convenient way:

$$\begin{aligned} \tilde{a}_0 &= (a_0 + \omega^0 a_2) + \omega^0 (a_1 + \omega^0 a_3) \\ \tilde{a}_1 &= (a_0 - \omega^0 a_2) + \omega^1 (a_1 - \omega^0 a_3) \\ \tilde{a}_2 &= (a_0 + \omega^0 a_2) - \omega^0 (a_1 + \omega^0 a_3) \\ \tilde{a}_3 &= (a_0 - \omega^0 a_2) - \omega^1 (a_1 - \omega^0 a_3) \end{aligned}$$

Out of the 16 multiplications initially needed, only 4 multiplications must now be performed: $\omega^0 a_2$, $\omega^0 a_3$, $\omega^0 (a_1 + \omega^0 a_3)$, and $\omega^1 (a_1 - \omega^0 a_3)$. Even though some of those multiplications may seem trivial, given that one of the operands is $\omega^0 = 1$, the reader must take into account that this example is related to $f(x) = x^4 - 1$. As the degree of the polynomial increases, trivial multiplications become a minority.

Taking this technique as starting point, it is possible to implement this procedure as a recursive function, as shown in [2]. In this optimisation, each polynomial can be recursively decomposed into sub-polynomials with terms of even and odd order being treated in the same way. In order to simplify the notation, in the following formulas $a(x) = X$, the subindex e means *even* and the subindex

o means *odd*.

$$\begin{aligned}\text{NTT}(X)[k] &= \text{NTT}(X_e)[k] + \omega_k \text{NTT}(X_o)[k] \mod q \\ \text{NTT}(X)\left[k + \frac{n}{2}\right] &= \text{NTT}(X_e)[k] - \omega_k \text{NTT}(X_o)[k] \mod q\end{aligned}$$

Using the polynomial $a(x) = 2 + 5x + 3x^2 + x^3$ with $n = 4$, $q = 17$, and $\omega = 4$, its NTT could be obtained as it is described in the next paragraphs.

$$\begin{aligned}X_o &= [5, 1] \rightarrow X_{o_o} = [1], X_{o_e} = [5] \\ X_e &= [2, 3] \rightarrow X_{e_o} = [3], X_{e_e} = [2]\end{aligned}$$

Given that $\omega = 4$, it is necessary to compute its powers for the omega array:

$$\begin{aligned}\omega^0 = 4^0 &\equiv 1 \mod 17 & \omega^1 = 4^1 &\equiv 4 \mod 17 \\ \omega^2 = 4^2 &\equiv 16 \mod 17 & \omega^3 = 4^3 &\equiv 13 \mod 17\end{aligned}$$

Now it is possible to compute the next step:

$$\begin{aligned}\text{NTT}(X_o)[k] &= \text{NTT}(X_{o_e})[k] + \omega_k \text{NTT}(X_{o_o})[k] \mod q \\ \text{NTT}(X_o)\left[k + \frac{n}{2}\right] &= \text{NTT}(X_{o_e})[k] - \omega_k \text{NTT}(X_{o_o})[k] \mod q\end{aligned}$$

Thus:

$$\begin{aligned}\text{NTT}(X_o)[0] &= 5 + 1 \cdot 1 \mod 17 \equiv 6 \\ \text{NTT}(X_o)[1] &= 5 - 1 \cdot 1 \mod 17 \equiv 4 \\ \text{NTT}(X_e)[0] &= 2 + 1 \cdot 3 \mod 17 \equiv 5 \\ \text{NTT}(X_e)[1] &= 2 - 1 \cdot 3 \mod 17 \equiv 16\end{aligned}$$

In this way, $\text{NTT}([5, 1]) = [6, 4]$ and $\text{NTT}([2, 3]) = [5, 16]$. In the last step:

$$\begin{aligned}\text{NTT}(X)[0] &= 5 + 1 \cdot 6 \mod 17 \equiv 11 \\ \text{NTT}(X)\left[0 + \frac{4}{2}\right] &= 5 - 1 \cdot 6 \mod 17 \equiv 16 \\ \text{NTT}(X)[1] &= 16 + 4 \cdot 4 \mod 17 \equiv 15 \\ \text{NTT}(X)\left[1 + \frac{4}{2}\right] &= 16 - 4 \cdot 4 \mod 17 \equiv 0\end{aligned}$$

As it was expected, $\text{NTT}([2, 5, 3, 1]) = [11, 15, 16, 0]$. Following this procedure, only $\frac{1}{2}n \log_2(n)$ multiplications are necessary for the regular NTT. As it is shown in [6], the regular INTT and the negacyclic NTT have the same number of operations, while the negacyclic INTT requires $\frac{1}{2}n \log_2(n) + n$ multiplications. With this optimisation, the overall time complexity of polynomial multiplication is $O(n \log_2(n))$.

4 Experimental Comparison

In order to compare the performance of these algorithms in practice, a Java application has been developed specifically for this contribution. The code is available at [5].

The application is flexible enough so the three algorithms described in this contribution (direct multiplication, NTT and the optimized version of NTT) can be tested in both $\mathbb{Z}_q[x]/\langle x^n - 1 \rangle$ and $\mathbb{Z}_q[x]/\langle x^n + 1 \rangle$. It is also possible to select the number of coefficients for the polynomials and the number of multiplications to be performed for each algorithm.

The time required to initialise the data structures for the regular NTT and the optimised NTT has been included in the tables presented in this section, so the reader can check how this initialisation time affects the running time. In order to obtain more accurate results and to avoid possible interferences in the measurement process, batches of 100 consecutive tests were performed with each algorithm using a computer with an Intel Core i7-10750H processor.

Table 1 contains the parameters used in the different tests. In all of them $q = 12289$, which is a value that allows to test a wide range of cases. Another reason for using this value is that it is employed by many PQC algorithms [6].

Table 1. Parameters used in the tests

n	$n^{-1} \pmod{q}$	ω	$\omega^{-1} \pmod{q}$	ψ	$\psi^{-1} \pmod{q}$
4	9217	1479	10810	4043	5146
8	10753	4043	5146	5736	11567
16	11521	722	6553	2545	1212
32	11905	1212	2545	5023	2975
64	12097	563	5828	2747	9238
128	12193	81	11227	9	2731
256	12241	9	2731	3	8193
512	12265	3	8193	1321	8633
1024	12277	49	1254	7	8778
2048	12283	7	8778	5846	9613

Table 2 shows the results for the tests with the ring $\mathbb{Z}_q[x]/\langle x^n - 1 \rangle$. The table provides two values in each cell (except for the column on the direct multiplication method): the mean time to perform one multiplication and the mean time plus the initialisation time, which is of interest in cases where only one multiplication has to be performed for a given set of parameters. The figures in bold represent the lower values for each combination of polynomial length and option (including or excluding initialisation time).

Table 3 shows the results in the same testing conditions, with the only difference that, in this case, the ring $\mathbb{Z}_q[x]/\langle x^n + 1 \rangle$ has been used.

Table 2. Mean time in nanoseconds for one multiplication (regular NTT)

n	Direct multiplication	Schoolbook NTT	Optimized NTT
4	2118	4045/14956945	7078/38878
8	6164	10167/21156967	12648/38748
16	14980	29517/12195517	13617 /52417
32	31338	90685/13137985	29617 /65317
64	83867	274300/11703600	61453 /104153
128	220970	821035/8190935	118805 / 186505
256	586513	2871617/15330617	273837 / 367737
512	2237543	10821181/34928081	373777 / 472277
1024	12164922	42695440/103778740	758134 / 874234
2048	54659706	170061732/359191632	1109974 / 1285174

Table 3. Mean time in nanoseconds for one multiplication (negacyclic NTT)

n	Direct multiplication	Schoolbook NTT	Optimized NTT
4	2380	6730/14137630	6399/201499
8	6072	15292/23220692	10865/276665
16	18394	32866/11246966	15993 /210393
32	36487	93322/13553622	32446 /279346
64	94732	271904/11272804	65731 /230331
128	220948	826926/8523026	123388 /228288
256	575243	2885876/17490876	200262 / 325262
512	2357461	10823606/34045006	390973 / 588373
1024	11041841	42687105/104100905	577396 / 760896
2048	55976373	169737361/351619561	1353031 / 1859531

The results show that, in our tests, direct multiplication is the fastest method for polynomials with up to 8 coefficients in both the regular and negacyclic cases when no initialisation time is considered. However, when initialisation time is taken into account, direct multiplication is the fastest method for polynomials with up to 64/128 coefficients in the regular/negacyclic case. For the rest of the polynomials, the optimised version of NTT offers a great advantage. For example, when $n = 2048$ its computation time is less than 5% of the time needed by the direct multiplication method.

On the other hand, the plain NTT implementation is around three times slower than the direct multiplication method in most tests when no initialisation is considered. This is to be expected, as each NTT and INTT operation takes approximately the same time as the entire direct multiplication. When

initialisation time is taken into account, the plain NTT method is at least seven times slower than the direct method.

When comparing tests for close values of n , in some cases there is a noticeable variability. This could be due to the nature of the tests, as a new set of random coefficients is generated for each value of n . In addition, both the NTT and its optimised version use the `BigInteger` class, while the direct multiplication method uses only basic datatypes (the integer type `long`), so in the latter case no object management is required.

Regarding the initialisation time, it ranges from 26 picoseconds for $n = 8$ to 175 picoseconds for $n = 2048$. The initialisation values for $n \in \{4, 8, 16, 32, 64\}$ are very similar, which means that the instantiation of the `Random` class takes more time compared to the time spent by successively calling the `nextInt` method. From $n = 128$ onwards, the time needed for calling the `nextInt` method n times exceeds the time needed to instantiate the class and follows a linear trend.

5 Conclusions

The NTT is a relevant algorithm in the field of Cryptography. It can be used to speed up polynomial multiplication in the rings $\mathbb{Z}_q[x]/\langle x^n - 1 \rangle$ and $\mathbb{Z}_q[x]/\langle x^n + 1 \rangle$. Even though there have been previous studies about the theoretical performance of both the regular NTT and some of its variants, it is important to obtain accurate data on the performance of software implementations of these algorithms.

In this contribution, three algorithms for computing polynomial multiplications in the rings $\mathbb{Z}_q[x]/\langle x^n - 1 \rangle$ and $\mathbb{Z}_q[x]/\langle x^n + 1 \rangle$ have been compared: the direct method, the plain NTT version and the optimised NTT variant described in [2]. A Java implementation was created to test these algorithms.

The tests used values defined in actual PQC specifications, such as the value of q and the length of the polynomials. With this data, it can be stated that the direct method is faster for polynomials with a small number of coefficients. However, for polynomials with a large number of coefficients, the optimised version of NTT is significantly faster. In fact, for polynomials with more than 1000 coefficients, its running time is as much as 5% of that of the direct method.

Regarding the plain NTT implementation, it is the worse option in terms of performance, and should not be used in any cryptographic implementation.

Acknowledgements. This work was supported in part by the Spanish State Research Agency (AEI) of the Ministry of Science and Innovation (MICINN), project P2QProMeTe (PID2020-112586RB-I00/AEI/10.13039/501100011033).

References

1. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key exchange—a new hope. In: 25th USENIX Security Symposium (USENIX Security 2016), pp. 327–343 (2016). <https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim>
2. Bakkebø, A.: Implementation of the Number Theoretic Transform for Faster Lattice-Based Cryptography. <https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/2778380/no.ntnu>
3. Bernstein, D.J.: Multidigit multiplication for mathematicians (2001). <https://cr.yp.to/papers/m3-20010811-retypeset-20220327.pdf>
4. Bos, J., et al.: CRYSTALS-Kyber: a CCA-secure module-lattice-based KEM. In: 2018 IEEE European Symposium on Security and Privacy (2018). <https://doi.org/10.1109/eurosp.2018.00032>
5. Gayoso Martínez, V.: NTT performance test. <https://github.com/victor-gayoso/ntt-test>
6. Liang, Z., Zhao, Y.: Number theoretic transform and its applications in lattice-based cryptosystems: a survey (2022). <https://arxiv.org/pdf/2211.13546.pdf>
7. Longa, P., Naehrig, M.: Speeding up the number theoretic transform for faster ideal lattice-based cryptography (2016). <https://eprint.iacr.org/2016/504>
8. Peikert, C.: Lattice cryptography for the internet. In: Mosca, M. (ed.) PQCrypto 2014. LNCS, vol. 8772, pp. 197–219. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11659-4_12
9. Satriawan, A., Mareta, R., Lee, H.: A complete beginner guide to the number theoretic transform (NTT). Cryptology ePrint Archive, Paper 2024/585 (2024). <https://eprint.iacr.org/2024/585>
10. Zhang, J., Zhang, Z.: Lattice-Based Cryptosystems. Springer, Singapore (2020). <https://doi.org/10.1007/978-981-15-8427-5>



Exploring the Landscape of Honeypots in the Fight Against Cyber Threats: A Systematic Mapping of Literature

Alberto Miguel-Diez¹(✉), Rodrigo González-Fernández²,
Gonzalo Esteban-Costales¹, Christian Vega-González³,
Adrián Campazas-Vega¹, Vicente Matellán-Olivera¹,
and Angel Manuel Guerrero-Higueras¹

¹ Robotics Group. University of León, MIC building. Campus de Vegazana, s/n,
24071 León, Spain

{amigd,gestc,acamv,vmato,agueh}@unileon.es

² Supercomputación Castilla y León (SCAYLE), CRAI-TIC building. Campus de
Vegazana, s/n, 24071 León, Spain

rodrigo.gonzalez@scayle.es

³ Spanish National Cybersecurity Institute (INCIBE), Avenida de Jose Aguado 41,
24005 León, Spain

christian.vega@incibe.es

Abstract. The steady increase in cyber threats highlights the need to strengthen cybersecurity defenses. In this context, honeypots emerge as crucial tools for detecting and mitigating such threats. A honeypot is a computer decoy that simulates having certain vulnerabilities that are attractive to a cybercriminal, allowing observation and analysis of the attack and the techniques used on the decoy. This article presents a systematic literature review of honeypots, analyzing the most commonly used characteristics and protocols following the PRISMA guidelines. A total of 258 scientific articles were reviewed, with 17 of them analyzed in depth. The results showed that the most commonly used protocols were HTTP and HTTPS, and the areas with the most available honeypots are those related to web applications and IoT devices. Additionally, lists have been compiled grouping honeypots by field of action, type of interaction, and protocol, with the aim of identifying aspects of honeypot development that require further investigation in future researches.

Keywords: systematic mapping · literature review · honeypots · malware · decoys

1 Introduction

According to the report “Radiography of the State of Cybersecurity”, 60% of Spanish companies experienced a system attack from January to June 2023,

Type of contribution: *Original research*

with ransomware being one of the most common attacks [1]. In order to mitigate such threats, organizations must review their security policies and enhance and develop efficient defense mechanisms, including honeypots.

A *honeypot*, defined by the National Cybersecurity Institute (INCIBE) [2], is a security tool designed as bait to attract cybercriminals, aiming to monitor and detect potential cyber attacks. It deploys fake services like web servers or databases, enticing attackers whose engagement provides valuable attack data for analysis. Research in this field covers assessing efficacy and real-world deployment, continuously gathering evidence from existing literature.

Works such as Bringer, Chelmecki, and Fujinoki's [3] illustrate how advances in honeypots are beginning to pave the way for new future trends, including the emergence of new honeypots that address new security vulnerabilities, as well as the utilization of honeypot data to enhance threat detector accuracy. On the other hand, Nawrocki, Wählisch, Schmidt, et al. [4] provide a more detailed insight into the software used in the field of honeypots, ranging from methodologies to ethical issues. Similarly, Franco, Aris, Canberk, et al. [5] conduct an analysis of honeypots related to IoT environments, Industrial IoT, and CPS, pinpointing crucial aspects in their design. Meanwhile, Ikuomenisan and Morgan [6] examine previous research on honeypots, summarizing a set of articles following the meta-review methodology. Lastly, Ilg, Duplys, Sisejkovic, and Menth [7] offer an analysis of honeypot solutions, highlighting open-source solutions.

Despite a comprehensive review of existing literature, no conclusive evidence has been found regarding the attributes of honeypots and their protocols. The article presents a systematic literature mapping to highlight the characteristics and attributes of various honeypots, providing a foundation for further research and understanding of implemented properties. Thus, the following research questions (RQ) are proposed:

- **RQ1:** What are the properties and attributes of the honeypots studied within the academic literature?
- **RQ1.1:** Which protocols can each honeypot emulate?

To detail the answers, the rest of the work follows the general structure of the PRISMA guide for conducting systematic reviews [8]. Section 2 provides an introduction to honeypots and their various types. Section 3 summarizes the planned review protocol for this study, while Sect. 4 presents the results obtained from the research. Based on the evidence found, Sect. 5 addresses the research questions raised and identifies the study's limitations. Finally, the conclusions of this review are presented in Sect. 6.

2 Background

Organizations employ a wide variety of traditional defensive security mechanisms, such as *firewalls*, access controls, or intrusion detection systems, to mitigate threats [5]. However, many of these systems lack the capability to detect threats early or collect detailed information about the methods and techniques

employed by attackers. This is why honeypots have emerged as security decoys specifically designed to attract and exclusively engage attackers. They have the ability to record both the activities performed in an attack and information about the agent carrying them out. Currently, honeypots are commonly deployed in combination with other defense mechanisms, such as firewalls or intrusion detection and prevention systems, to strengthen organizations' overall security [6].

2.1 Types of Honeypots

In the context of this work, honeypots will be classified according to the level of interaction of the attacker with the computer system. This involves evaluating the critical point at which the real system could be compromised, and consequently, the information that could be exposed.

For this reason, in the literature, three different types of honeypots are distinguished based on the interaction an attacker can exert: low, medium, and high [3–7].

1. **Low-Interaction Honeypots** simulate basic services, offering predefined responses to attacks without risking real system security, while minimizing interaction.
2. **Medium-Interaction Honeypots** provide limited command-line interfaces for higher interaction and more detailed attacker information collection.
3. **High-Interaction Honeypots** offer realistic operating systems and services, allowing comprehensive attacker interaction and detailed tactic capture but demanding more resources and setup complexity.

2.2 Deployment Scenarios

On the other hand, the deployment objectives of honeypots fall into two types: research or production. Research honeypots are used to monitor existing attacks and identify new attack vectors. However, production honeypots are aimed at mimicking the settings of a component actually functioning within the company and being able to alert the presence of an attacker [7].

Another possible classification depends on their execution environment: real server or virtual machine. The former option is preferable due to its response times and a system environment closer to reality.

3 Methodology

The present study employs the methodology outlined by Kitchenham, Budgen, and Brereton [9] for systematic reviews in computer science. It aims to provide a rapid overview of honeypot literature from journals and conferences, focusing on peer-reviewed articles defining honeypot attributes, simulated protocols, application fields, and interaction levels.

3.1 Search Process

The search began with a set of semi-reference articles [10–14], informally collected before the systematic mapping process, based on the methodology proposed by Zhang, Babar, and Tell [15]. Automatic searches were conducted in IEEE Digital Library, Scopus, and Web of Science databases up to February 2024, filtering for English or Spanish articles with specific terms appearing in titles, abstracts, and keywords.

Subsequently, a search string was developed to address the two proposed research questions (RQ1 and RQ1.1), as both are related. As a result, the following search string was obtained: “*honeypot*” AND (“*properties*” OR “*characteristics*” OR “*features*”) AND (“*protocols*” OR “*services*”).

Finally, the automatic search was complemented with a *snowball sampling process*, which consisted of checking the references of those articles reviewed after the selection process and that directly addressed the research questions.

3.2 Selection Process

The selection process for filtering search results involved inclusion and exclusion criteria detailed in Table 1. Initially, titles, abstracts, and keywords were reviewed; however, a second phase was conducted for articles needing further scrutiny. Articles focusing on honeypots were included, while those on *honeynets*-a network consisting of the implementation of multiple individual honeypots following a specific network topology-were excluded [16] and lacking honeypot definitions were excluded, along with those with inaccessible full texts.

Table 1. Set of inclusion and exclusion criteria

Criteria type	Description
Inclusion	The content focuses on the description of one or more honeypots
Exclusion	The content focuses on honeynets Characteristics are not defined The full article has not been obtained

3.3 Quality Assessment Process

Due to the large number of articles that have passed the selection process, it was decided to develop a quality assessment process with the aim of obtaining only those that would best answer the research questions. This means that a rating between 0 and 4 points was assigned to each article, and a threshold of acceptance of 2.5 points was established. In this way, those articles that did not exceed the threshold would be discarded. The list of control questions is as follows:

1. Does the work define the protocols/services addressed by the honeypot?
2. Are the characteristics obtained clearly (e.g., through a list, figure, or table)?
3. Is the source code available in a publicly accessible repository for free?
4. Are real environments evaluated/experimented with and favorable results obtained?

Favorable outcomes in the last question were defined as honeypots successfully deceiving attackers or capturing malware, while unfavorable outcomes included compromise, allowing attackers to escape. Responses were categorized as “Yes”, “Partially”, or “No”, each contributing 1, 0.5, or 0 points to the rating, respectively.

3.4 Data Extraction and Synthesis Processes

The systematic mapping aims to address research inquiries, employing data extraction facilitated by an electronic form, specifically devised using the online tool Parsifal¹, intended for systematic literature reviews/mappings. The extraction process seeks pertinent details from each article, including the honeypot’s name, utilized protocol, application/field, interaction type (low, medium, or high), and the authors’ delineated characteristics.

4 Results

The result of the search and selection processes, upon execution, can be summarized in the PRISMA diagram shown in Fig. 1.

From the automatic search in the established information sources, a total of 258 articles were identified. Regarding the selection process, a first phase was carried out in which 79 duplicate articles were discarded, followed by the review of the remaining 179. Next, 168 articles were discarded: 154 for meeting some exclusion criteria and another 14 for not reaching the quality threshold. Therefore, the set of main articles was composed of 11 works, to which another 4 semi-reference set works had to be added. Subsequently, a complementary search was conducted on the references of these 15 primary articles, reviewing only their title and abstract. In this process, 2 more relevant articles were obtained. Therefore, a total set of 17 primary articles were identified at the end of the entire review process. The articles are published between the years 2004 and 2023.

5 Discussion

Question RQ1 aims to find the properties and attributes of the different honeypots found in the literature. While there is a large number of articles by

¹ <https://parsif.al>.

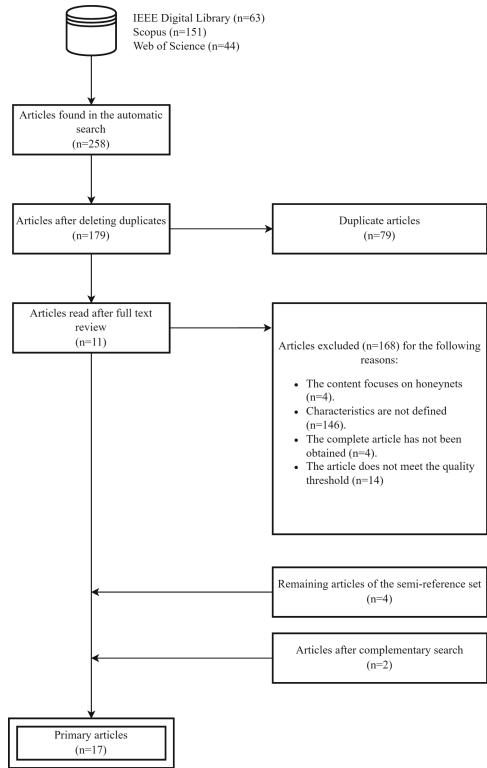


Fig. 1. PRISMA Diagram

researchers using these types of decoys, few have created a honeypot and described its characteristics, resulting in a relatively small number of articles.

The interaction type of honeypots is a feature used to group the different decoys. This is one of the most interesting characteristics along with the emulated protocol, as it is often crucial when choosing one option over another. Among all the articles found, there are 6 that do not specify their level of interaction ([13, 17–20], and [14]), but in some cases, it can be inferred from their functioning.

Next, we describe the characteristics of the articles found in the previous search:

IoTCMal [21] is a honeypot composed of a low-interaction part implementing Telnet/SSH protocols and a high-interaction part emulating IoT device services. The authors aim to capture malware, so all commands that initiate downloads such as `wget` or `curl` save the downloaded files to obtain binaries. Regarding the high-interaction component, the authors mapped ports of real vulnerable services exposed on the Internet.

In “Design and implementation of web service honeypot” [22], the authors propose a high-interaction honeypot for emulating web services using Axis and .NET technologies. This system captures all incoming and outgoing messages and analyzes them using *machine learning* techniques to detect and facilitate the study of attacks, avoiding the need for manual log review. The developed network extracts data such as source IP, message length, response time, etc., from the packets. The SVM classifier implemented has a 95% detection rate for those analyzed and classified as malicious data.

The HoneyIM solution [17] is based on the concept of detecting instant messages that may contain malware or a dangerous URL. This proposal is functional in both public and private IM² services and consists of creating a contact that users must add to their contact list. If this contact receives a file or URL at any time, there is a high probability that it is an attacker. Additionally, a malware detection module and a suppression mechanism are implemented to detect incidents and inform the administrator.

In Pasithea [23], a honeypot is developed with the aim of emulating the behavior of a web server. This system is designed to collect a variety of information about clients (request date, HTTP method, the address the client attempted to access, the client’s IP address, etc.). One of Pasithea’s main objectives is to maintain its indistinguishability to the attacker. Therefore, despite its low-interaction characteristic and returning a 404 *Not Found* error response, it proposes to return a valid response in JSON format. Another aspect to highlight is its operation and defense against DDoS attacks and XSS commands.

IoTZeroJar [18] is a decoy that serves for detecting malicious activities and analyzing zero-day attacks in IoT. This solution originates from IoTCandyJar [24] but improves zero-day capture. The architecture is based on four modules: IoTScanner, SQL database, honeypot instances, and the reinforcement learning module. The first module is responsible for creating a *pool* of responses from real IoT devices, which aims to ensure that the attacker does not suspect.

Honeyboost [19] is an innovative solution designed to anticipate attacks within local area networks (LAN), employing a protocol that captures broadcast ARP requests and those TCP or UDP packets directed to the system. Additionally, it incorporates a Network Anomaly Detection System (NADS). Honeyboost employs two approaches: a horizontal one, which treats data from each node as a time series and finds anomalies based on specific node characteristics, and a vertical one, which finds anomalies in each protocol space.

On the other hand, NeuPot [20] is a decoy for industrial control systems based on Modbus. It is based on three types of entities: the industrial agent, the server, and the honeypot node. The industrial agent transmits parameters of the physical process, from industrial control systems (ICS), to the server through storage media. The server is responsible for honeypot configuration, data storage and visualization, as well as for creating time series forecasting models and malicious traffic detection. The honeypot node interacts with attackers through port 502

² Instant Messaging.

of the Modbus protocol and generates responses based on a configuration file, while collecting malicious traffic.

HoneyCY [25] is a low-interaction honeypot that integrates a set of open-source honeypots. It consists of honeyPi (Dionaea, Kippo, and Glastopf), HoneySrv, and HoneyVm, the latter of which includes integrated Cuckoo Sandbox for analyzing found binaries.

On the other hand, in HoneyC [26], the authors present a low-interaction decoy for the HTTP protocol. In this proposal, several extra components can be implemented, such as the Yahoo API for URL search or rule-based analysis with Snort. This framework, conceived with inspiration from Unix flexibility, consists of three modular components: Queuer, Visitor, and Analysis Engine, which interact through a command redirection operator.

U-Pot [13] is a system designed to detect threats on IoT devices, particularly focusing on vulnerabilities in a specific type of smart switch and its UPnP architecture. U-Pot's paper outlines its operational process, which includes device analysis, state scanning, and emulation for user interaction. U-Pot utilizes openHAB to compare results with real devices, ensuring its effectiveness in deceiving both users and search engines. However, it notes a drawback regarding the response time, where the simulated device is faster than the real one, potentially allowing attackers to identify the emulation.

In Honeyware [27], a low-interaction client-based web honeypot is introduced. This honeypot interacts with a given URL simulating a specific web browser through its user-agent string and then downloads the response from the target website. This response is examined for threat detection. The collected information will contain data about the agent, such as the web browser name, its version, and the host operating system name. Additionally, this solution is open source and is considered a hybrid approach combining Capture-HOC with the honeypot to improve response time.

The solution in [14] introduces a web honeypot for IPv6 networks, consisting of a weak password module and an SQL injection module to gather attacker information. An analysis component, the Index of Security Awareness, determines the criticality of attacked areas using assigned weights. The honeypot logs information like timestamps, usernames, and passwords, and is deployed via Docker for efficient attack capture. Additionally, captured IP addresses can be blacklisted by the network's firewall.

In the article proposed by Cernica and Popescu, a honeypot module for Wordpress is presented [11]. This module is highly interactive, generating response messages based on real system resources and installing plugins based on attackers' requests, potentially containing known vulnerabilities. Its architecture includes a proxy server to receive and forward attacker requests to the internal network, ensuring that direct connections to the Internet from the internal network are avoided, thus reducing the risk of malicious activities if the server is compromised.

Thingpot [12] is a medium-interaction decoy for IoT. This one presents a difference from the rest of the honeypots in this field in that it implements the complete IoT platform and not just the common protocols. REST API is used to build the backend, the XMPP protocol for real-time communication between IoT devices, and the frontend is an HTTP web service. All the services described above operate in isolated virtual environments, preventing an attacker from reaching the real system under which they are hosted.

In the work proposed by Sethia and Jeyasekar [10], the Dionaea honeypot is used to capture zero-day attacks and malware in a low-interaction environment. This system operates with various protocols such as SMB, HTTP, FTP, TFTP, MSSQL, and VoIP. Its architecture includes a logging installation, an incident handler (ihandler) that provides real-time information about ongoing attacks, and the malware binaries obtained. This honeypot emulates vulnerable services and has proven effective in capturing droppers and trojans, mainly through the FTP protocol.

J-Honeypot [28] implements a simulation of virtual servers that works on different platforms, Windows or UNIX, workstations, or mobile devices. This low-interaction honeypot includes an intrusion detection system based on rules and a web interface to monitor the tool. Additionally, its ability to store all packets in an SQL database, record all incoming packets in a tcpdump-formatted file, and configure the honeypot in real-time without the need for a restart (hot reload) stands out, significantly facilitating the developer's task.

Finally, S7commTrace [29] is a proposal applicable in the field of industrial control systems. The authors created a high-interaction honeypot divided into four modules, among which the TCP communication module stands out, which has port 102 open, and the one containing the simulation module responsible for crafting the user's request response. This is based on the "user template" module in which the PLC serial, manufacturer, etc., are configured. Finally, the storage module saves all requests and responses for later analysis.

Table 2 shows the previously mentioned honeypots ordered by their field and year. This allows for a quick understanding of the fields for which there are honeypots currently and the type of interaction they implement.

5.1 Research Question RQ1.1

The research question RQ1.1 seeks to understand the variety of protocols that existing honeypots can simulate. Table 2 provides a clear overview, revealing that HTTP/HTTPS protocols are most commonly emulated, while less common ones like S7Comm and UPnP are also implemented, despite their importance in cyber-physical or IoT systems. Furthermore, instant messaging has been included as a protocol, acknowledging the existence of honeypots operating in this domain, although it's not strictly categorized as such.

Table 2. Set of articles sorted by field and year

Field	Year	Article	Interaction Type	Protocols
IoT	2018 [13]	—	—	UPnp
IoT	2018 [12]	Medium	—	MQTT, XMPP, HTTP/HTTPS
IoT	2020 [21]	Medium	—	SSH, Telnet
IoT	2022 [18]	—	—	CWM, Cryptocurrency Dash, HTTP/HTTPS Kerberos, Telnet, TorPark
Web	2007 [26]	Low	—	HTTP/HTTPS
Web	2010 [27]	Low	—	HTTP/HTTPS
Web	2011 [22]	High	—	HTTP/HTTPS
Web	2018 [11]	High	—	HTTP/HTTPS
Web	2020 [14]	—	—	HTTP/HTTPS
ICS	2018 [29]	High	—	S7Comm
ICS	2023 [20]	—	—	Modbus
IM	2007 [17]	—	—	Instant messaging
REST API	2017 [23]	Low	—	HTTP/HTTPS
LAN	2022 [19]	—	—	ARP, TCP, UDP
Varios	2004 [28]	Low	—	ARP, ICMP, IP, TCP, UDP
Varios	2015 [25]	Low	—	FTP, HTTP/HTTPS, MSSQL, SSH, SMB, TFTP, VoIP
Varios	2019 [10]	Low	—	FTP, HTTP/HTTPS, MSSQL, SMB, TFTP, VoIP

6 Conclusions

Honeypots, vital in cybersecurity amidst rising internet threats, have evolved significantly since their 2004 inception [30], transitioning from low-interaction models to sophisticated systems seamlessly integrated into diverse computing environments. This study, through a systematic mapping approach, initially sourced 258 articles, culminating in the review of 17 articles after comprehensive evaluation and supplementary searches.

The examination of these articles reveals a prevalent focus on honeypots' role in capturing malware and uncovering zero-day vulnerabilities. Many honeypot implementations prioritize disguising their virtualized nature from attackers, often by tweaking system parameters like RAM size, ROM memory, and installed programs to enhance deception and efficacy in threat detection.

Furthermore, the review has confirmed the existence of decoys for a wide variety of protocols, with HTTP/HTTPS being particularly prominent. However, lesser-known or more specific protocols such as S7Comm, CWM, or TorPark, among others, also have honeypots to detect potential vulnerabilities, as can be seen in Table 2.

In conclusion, the results presented in this literature review are particularly relevant for researchers who wish to understand the current landscape of honeypots. The features presented can serve to identify ideas that can be replicated in future developments and the current approaches used. Furthermore, knowing the protocols with fewer projects can motivate the development of solutions

to address these gaps. Additionally, no systematic review in the literature has reported results similar to those obtained in this article.

Acknowledgments. This publication is part of the project CIBERLAB, financed “by European Union NextGeneration-EU, the Recovery Plan, Transformation and Resilience, through INCIBE”; and part of the project EDMAR, PID2021-126592OB-C21, funded by MCIN/AEI/10.13039/501100011033 and by ERDF A way of making Europe.

References

1. Sierra, M.: España sufre en 2023 la mayor ola de ciberataques de su historia. *Voz Populi* **12**, 2023 (2023)
2. INCIBE. Honeypot: una trampa para los ciberdelincuentes — empresas — incibe (2023). <https://www.incibe.es/empresas/blog/honeypot-una-trampa-para-los-ciberdelincuentes>
3. Bringer, M., Chelmecki, C., Fujinoki, H.: A survey: recent advances and future trends in honeypot research. *Int. J. Comput. Netw. Inf. Secur.* **4**, 09 (2012)
4. Nawrocki, M., Wählisch, M., Schmidt, T.C., Keil, C., Schönfelder, J.: A survey on honeypot software and data analysis (2016). <http://arxiv.org/abs/1608.06249>
5. Franco, J., Aris, A., Canberk, B., Uluagac, A.S.: A survey of honeypots and honeynets for internet of things, industrial internet of things, and cyber-physical systems (2021). <https://arxiv.org/abs/2108.02287>
6. Ikuomenisan, G., Morgan, Y.: Meta-review of recent and landmark honeypot research and surveys. *J. Inf. Secur.* **13**, 181–209 (2022)
7. Ilg, N., Duplys, P., Sisejkovic, D., Menth, M.: A survey of contemporary open-source honeypots, frameworks, and tools. *J. Netw. Comput. Appl.* **220**, 103737 (2023). <https://www.sciencedirect.com/science/article/pii/S108480452300156X>
8. Moher, D., Liberati, A., Tetzlaff, J., Altman, D.G., Group, T.P.: Preferred reporting items for systematic reviews and meta-analyses: the prisma statement. *PLOS Med.* **6**(7), 1–6 (2009). <https://doi.org/10.1371/journal.pmed.1000097>
9. Kitchenham, B.A., Budgen, D., Brereton, P.: Evidence-Based Software Engineering and Systematic Reviews. Chapman and Hall/CRC, Boca Raton (2015)
10. Sethia, V., Jeyasekar, A.: Malware capturing and analysis using dionaea honeypot. In: 2019 International Carnahan Conference on Security Technology (ICCST), pp. 1–4 (2019)
11. Cernica, I., Popescu, N.: Wordpress honeypot module. In: 2018 IEEE 16th International Conference on Embedded and Ubiquitous Computing (EUC), pp. 9–13 (2018)
12. Wang, M., Santillan, J., Kuipers, F.: Thingpot: an interactive internet-of-things honeypot (2018)
13. Hakim, M.A., Aksu, H., Uluagac, A.S., Akkaya, K.: U-pot: a honeypot framework for upnp-based iot devices. In: 2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC), pp. 1–8 (2018)
14. Wang, K., Tong, M., Yang, D., Liu, Y.: A web-based honeypot in IPv6 to enhance security. *Information* **11**(9), 440 (2020)
15. Zhang, H., Babar, M.A., Tell, P.: Identifying relevant studies in software engineering. *Inf. Softw. Technol.* **53**(6), 625–637 (2011). <https://www.sciencedirect.com/science/article/pii/S0950584910002260>

16. Fan, W., Du, Z., Fernandez, D.: Taxonomy of honeynet solutions. In: 2015 SAI Intelligent Systems Conference (Intellisys), HPCC Syst; nVIDIA; Siemens; IEEE; Inst Engn & Technol; BigML; Stratified Med, New York, NY 10017 USA, Proceedings Paper, pp. 1002–1009, sAI Intelligent Systems Conference (IntelliSys), Sci & Informat Org, London, England, 10–11 November 2015. IEEE (2015)
17. Xie, M., Wu, Z., Wang, H.: Honeyim: fast detection and suppression of instant messaging malware in enterprise-like networks. In: Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007), pp. 64–73 (2007)
18. Ellouh, M., Ghaleb, M., Felemban, M.: Iotzerojar: towards a honeypot architecture for detection of zero-day attacks in iot. In: 2022 14th International Conference on Computational Intelligence and Communication Networks (CICN), pp. 765–771 (2022)
19. Kandanaarachchi, S., Ochiai, H., Rao, A.: Honeyboost: boosting honeypot performance with data fusion and anomaly detection. *Expert Syst. Appl.* **201**, 117073 (2022). <https://www.sciencedirect.com/science/article/pii/S0957417422004821>
20. Shan, Y., Yao, Y., Zhao, T., Yang, W.: NeuPot: a neural network-based honeypot for detecting cyber threats in industrial control systems. *IEEE Trans. Ind. Inf.* **19**(10), 10512–10522 (2023)
21. Wang, B., Dou, Y., Sang, Y., Zhang, Y., Huang, J.: Iotcmal: towards a hybrid iot honeypot for capturing and analyzing malware. In: ICC 2020 - 2020 IEEE International Conference on Communications (ICC), pp. 1–7 (2020)
22. Ghourabi, A., Abbes, T., Bouhoula, A.: Design and implementation of web service honeypot. In: SoftCOM 2011, 19th International Conference on Software, Telecommunications and Computer Networks, pp. 1–5 (2011)
23. Leaden, G., Zimmermann, M., DeCusatis, C., Labouseur, A.G.: An api honeypot for ddos and xss analysis. In: IEEE MIT Undergraduate Research Technology Conference (URTC) 2017, pp. 1–4 (2017)
24. Luo, T., Xu, Z., Jin, X., Jia, Y., Xin, O.: Iotcandyjar: towards an intelligent-interaction honeypot for iot devices (2017). <https://api.semanticscholar.org/CorpusID:33436536>
25. Christoforou, A., Gjermundrød, H., Dionysiou, I.: Honeycy: a configurable unified management framework for open-source honeypot services. In: Proceedings of the 19th Panhellenic Conference on Informatics, ser. PCI '15, pp. 161–164. Association for Computing Machinery, New York (2015). <https://doi.org/10.1145/2801948.2802021>
26. Seifert, C., Welch, I., Komisarczuk, P.: Honeycy - the low-interaction client honeypot (2007)
27. Alosefer, Y., Rana, O.: Honeyware: a web-based low interaction client honeypot. In: 2010 Third International Conference on Software Testing, Verification, and Validation Workshops, pp. 410–417 (2010)
28. Mai, Y., Upadrashta, R., Su, X.: J-honeypot: a java-based network deception tool with monitoring and intrusion detection. In: International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004, vol. 1, pp. 804–808 (2004)
29. Xiao, F., Chen, E., Xu, Q.: S7commTrace: a high interactive honeypot for industrial control system based on S7 protocol. In: Qing, S., Mitchell, C., Chen, L., Liu, D. (eds.) ICICS 2017. LNCS, vol. 10631, pp. 412–423. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-89500-0_36
30. Qassrawi, M.T., Zhang, H.: Client honeypots: approaches and challenges. In: 4th International Conference on New Trends in Information Science and Service Science, pp. 19–25 (2010)



Phishing and Spam Prevention Powered by Jetson Nano

Jackson Diaz-Gorrin^(✉) and Pino Caballero-Gil

Department of Computer Engineering and Systems, University of La Laguna, La Laguna, 38200 Tenerife, Spain
{alu0100536744,pcaballe}@ull.edu.es

Abstract. Detecting spam emails is crucial for mitigating phishing attacks. In this research, this problem has been addressed as a classification task, using machine learning techniques to distinguish between emails categorized as spam and non-spam. Thus, this research focuses on classifying emails based on their content to effectively identify messages considered spam as potential phishing attempts. The applied methodology began with a dataset comprising classified emails and thorough data preprocessing to ensure data quality and validity. Various neural network setups were assessed and compared using Scikit-Learn, exploring diverse parameterizations to enhance performance. This study proposes a low-power computing solution aimed at reducing email scams carried out through phishing attacks. Besides, a specific cybersecurity solution based on machine learning was designed and successfully deployed on the NVIDIA Jetson Nano platform.

Keywords: Cybersecurity · Artificial Intelligent · Phishing Prevention · Spam Prevention · Internet of Things · Jetson Nano

1 Introduction

Phishing is a fraudulent activity designed to deceive users into disclosing personal information such as passwords, credit card numbers, or banking details. It therefore poses a significant threat in many different environments. Hence, to protect users and organizations from cyber threats that involve the theft of sensitive information it is crucial to develop efficient methods. Techniques used to detect phishing attempts include email analysis, inspection of suspicious websites, advanced email filters and trained machine learning models to identify phishing patterns. In particular, through the detection of spam messages, machine learning can effectively identify potential phishing attacks. This is facilitated by leveraging datasets that include messages used in phishing attempts. Machine learning algorithms excel at recognizing patterns and common characteristics indicative of phishing, such as the presence of malicious URLs, requests for personal information, and various social engineering techniques. Training models on labeled datasets of such messages enables systems to differentiate

between legitimate communications and those posing potential risks, thereby enhancing automated detection and prevention measures.

The Phishing Activity Trends Report from the Anti-Phishing Working Group (APWG) reveals alarming statistics. In March 2022, APWG observed 384,291 attacks, marking the highest monthly total in its reporting history. Throughout the first quarter of 2022, APWG monitored a total of 1,025,968 phishing attacks, making it the worst quarter ever recorded by the organization and the first time quarterly attacks surpassed one million. This figure exceeds the previous record of 888,585 attacks observed in Q4 2021. Since early 2020, the number of phishing attacks has more than tripled, with APWG observing between 68,000 and 94,000 attacks per month during that period [1]. The traditional spam filters may encounter challenges in early detection to prevent users from falling victim to sophisticated phishing tactics, exacerbated by the advancements in generative AI models that can generate convincing emails. This underscores the critical need for more advanced and adaptable filtering methods.

The motivation of this work is to enhance email security by investigating a novel system designed to detect spam and prevent potential phishing threats. This study explores the integration of AI-driven neural networks for spam detection within the Jetson Nano [2], a compact and powerful computing platform capable of running AI applications. The GPU in the Jetson Nano enables efficient processing of AI models, ensuring accurate and rapid detection of spam and phishing emails. Additionally, its low-power consumption makes it a cost-effective choice for continuous operation, providing robust and reliable spam detection without significantly increasing energy costs. This combination of high performance and energy efficiency enhances the overall security of email communications, making the Jetson Nano an excellent option for deploying advanced email filtering systems.

An in-depth analysis was conducted to identify the most promising algorithm for spam detection. The study employed a Multi-Layer Perceptron (MLP) [3], a neural network characterized by multiple layers of interconnected nodes, including an input layer for initial data intake, one or more hidden layers for complex data processing, and an output layer for final classification or prediction. Subsequently, a Python script was developed to implement the trained MLP model on the Jetson Nano. Additionally, a scheduled task was configured to execute the script every minute, offering an automated solution for continuous spam detection in email systems. This implementation empowers the Jetson Nano to efficiently classify incoming emails, significantly reducing the risk of users falling victim to phishing attacks.

The scientific contribution of this research lies in providing a solution that aids in combating cyber threats by implementing an additional security layer for emails through customized trained models deployed on low-power devices. This approach facilitates rapid and customized updates to the filtering process, enhancing the security measures for spam detection implemented in email services.

This paper is organized as follows. Section 2 outlines the applied methods, providing a detailed description of the dataset of this study, an overview of the used neural network, and a discussion of the obtained results. Finally, Sect. 3 presents some conclusions and open problems.

2 Methodology

An automated email classification system has been implemented, using a scheduled task through Jetson Nano. This task runs every minute, enabling the classifier Python script to be automatically and periodically activated, connecting via the Gmail API. The program interacts with the Gmail account, gathering incoming emails and classifying them based on predefined algorithms and rules. This automation ensures constant review of messages without manual intervention, optimizing productivity and improving accuracy in email categorization. Additionally, the task has been configured to store messages in a file, facilitating program tracking, monitoring, and identification of potential issues or errors. A Python script was developed for execution on the Jetson Nano platform. To avoid retraining the model with each execution, the trained model file in .pkl format was imported and is used in every program run. Furthermore, credential and access token management have been established to avoid user authentication at each execution. Utilizing a neural network, the program classifies emails, and upon detecting spam emails. This automated, scheduled task-based approach ensures efficient and consistent email classification, enhancing inbox management.

Integration with the Gmail API facilitated access to email messages and their associated metadata, including sender, subject, date, and other pertinent details. Employing predefined algorithms and rules, the classifier program meticulously analyzed each email and categorized it based on established criteria.

This automated process ensured continual review of incoming emails, eliminating the need for manual intervention. By conducting periodic classifications, the program promptly processed new messages, streamlining inbox management.

The implementation of scheduled tasks for automatic classification optimized productivity by saving time and effort in organizing and categorizing emails. Moreover, it enhanced accuracy and consistency in message classification, leading to improved efficiency in email communication management.

Within the scheduled task, configurations were set to archive messages generated during each program execution to a file. This logging feature proved invaluable for verifying and evaluating the program's performance under various circumstances. Saving execution logs to a file provided a detailed history of past executions, aiding in the identification and resolution of potential issues or errors. Additionally, it offered a means to monitor the program's activities over time, facilitating analysis and performance enhancements. With this logging functionality, comprehensive information about each execution, including dates, times, generated messages, and any other pertinent data configured for logging, could be readily accessed.

The trained model was saved with the .pkl file extension and subsequently used in a Python script on the Jetson Nano.

To obviate the need for user authentication during each program execution, credential and access token files were configured. This allowed the scheduled task to execute seamlessly without requiring access credentials for each run.

Prior to the process, it is important to note that within the Gmail account, two labels were created: one labeled legitimate_mails and the other spam_emails. The goal is to streamline inbox management by automatically categorizing incoming emails into these two categories. Leveraging a neural network, the program executed the email classification process. Depending on the classification outcome, emails were assigned either the spam_email or legitimate_email label, ensuring a consistently spam-free inbox (see Fig. 1).

This automated email classification system ensures efficient inbox management by automatically removing spam email, saving time and effort. Additionally, by utilizing machine learning techniques based on a neural network, classifier accuracy and adaptability are improved, resulting in a more reliable and effective classification process.

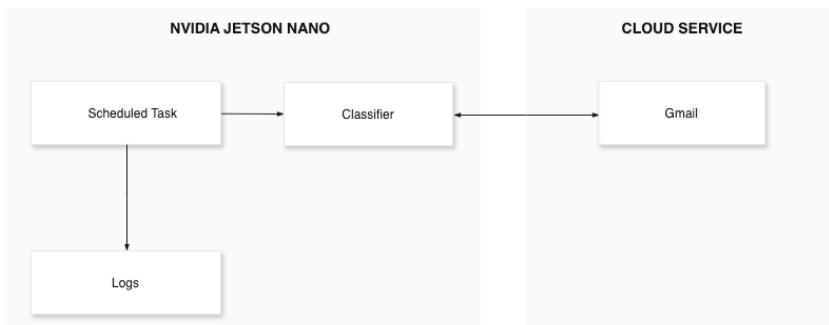


Fig. 1. Email Classification

A Python script relies on several key components. Firstly, the MLP neural network model was trained, enabling it to learn patterns and features relevant to email classification.

Furthermore, the dataset underwent word vectorization, a natural language processing technique that transforms words into numerical vectors. This process enables their representation in a format understandable by the neural network model. By capturing semantic and contextual information within the dataset, word vectorization enhances the model's capacity to effectively comprehend and classify emails.

In addition, the script utilizes a Gmail access file, which contains the necessary credentials and settings to establish a connection with the Gmail account. This connection enables the script to access the inbox, retrieve emails, and apply the trained model to classify them into predefined categories.

Finally, the Python script has been successfully deployed on a Jetson Nano device, demonstrating its ability to efficiently classify emails. Leveraging the computational power of the Jetson Nano, the script seamlessly processes incoming emails and accurately assigns them to the appropriate categories.

2.1 Dataset

The dataset used for this research was derived from the email classifications performed by the Apache SpamAssassin open-source project [4], formatted as a .csv file. To ensure the dataset's suitability for model training, extensive data preprocessing and exploratory analysis were conducted. The preprocessing procedure included checking for the presence of null values and ensuring no correlation between the variables. The variables are *message* and *label*. Furthermore, due to a class imbalance in the dataset (see Fig. 2a), a downsampling process [5] was conducted to address the imbalance between spam and non-spam instances (see Fig. 2b). Each email message was labeled with a value of 1 for spam and 0 for non-spam. After preprocessing, each class was populated with 2,658 messages.

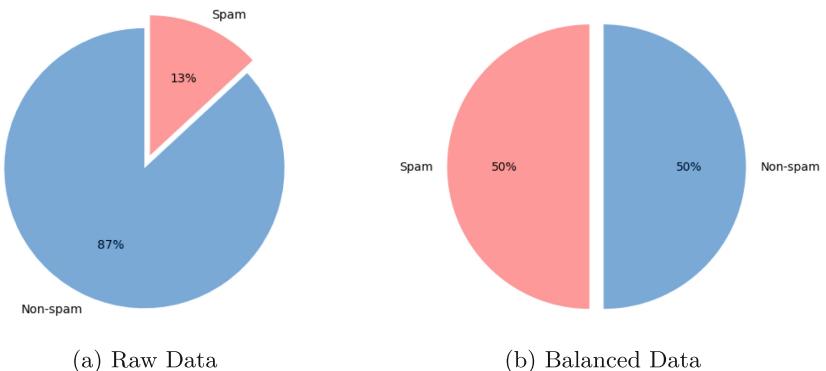


Fig. 2. Dataset

The model training was conducted using a Google Colab notebook, leveraging the Python programming language and the Scikit-Learn library. The model underwent training using a labeled dataset for supervised learning, where each example is paired with a known outcome, enabling the model to learn patterns and relationships between input features and target labels. To assess its performance, a cross-validation technique was implemented, dividing the dataset into 70% for training and 30% for testing. This method ensures rigorous training and evaluation of the model, protecting against overfitting.

Upon completion of training, the model was exported in .pkl format, commonly used for storing machine learning models, enabling their subsequent use in classifying new emails. The exported file was then integrated into a Python script

deployed on the Jetson Nano, allowing efficient email classification. This setup leverages the pre-trained model to accurately detect spam emails and potential phishing attacks.

2.2 Metrics

In the context of evaluating classification models, it is important to consider metrics such as accuracy, precision, recall, and F1 score. These metrics provide a measure of the model's performance in terms of its classification capabilities according to the equations below [6].

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (1)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3)$$

$$\text{F1 Score} = \frac{2 \cdot (\text{Precision} \cdot \text{Recall})}{\text{Precision} + \text{Recall}} \quad (4)$$

where:

- TP represents True Positives, which are the number of instances correctly classified as positive (e.g., correctly classified spam emails).
- TN represents True Negatives, which are the number of instances correctly classified as negative (e.g., correctly classified non-spam emails).
- FP represents False Positives, which are the number of instances incorrectly classified as positive (e.g., non-spam emails incorrectly classified as spam).
- FN represents False Negatives, which are the number of instances incorrectly classified as negative (e.g., spam emails incorrectly classified as non-spam).

2.3 Neural Network

A neural network is a machine learning model inspired by the structure and functioning of the human brain. It consists of processing units called neurons, organized into interconnected layers. Each neuron receives inputs, performs a calculation, and produces an output. The output of one neuron is used as input for the neurons in the next layer, forming a network of connections.

The MLP is a type of feedforward neural network, meaning that information flows in one direction, from the input layer to the output layer. In the context of email classification, the model analyzes email message characteristics to classify them as spam or non-spam.

An example of email classification using a neural network (see Fig. 3), where n represents any value for the number of neurons required for the input layer. Each hidden layer contains a set of neurons that process the input information

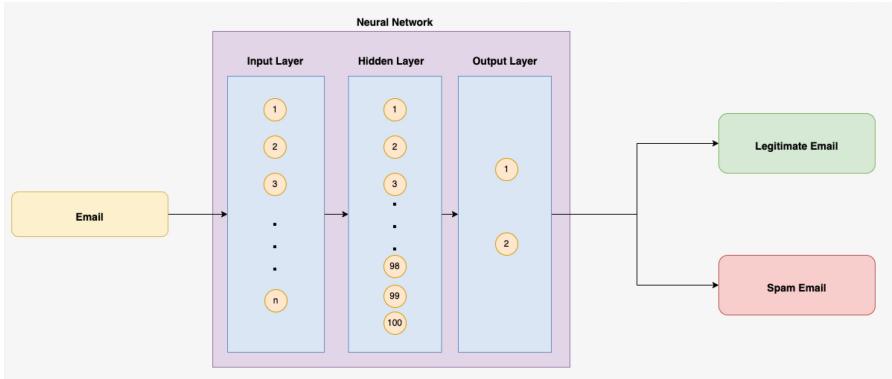


Fig. 3. Neural Network

and generate an output. These neurons are connected with synaptic weights, which determine the influence of each input on the neuron's output.

During training, the MLP automatically adjusts the synaptic weights to minimize the difference between the predicted outputs and the desired outputs. It uses an algorithm called error backpropagation to propagate the error from the output layer to the previous layers, gradually adjusting the weights to improve prediction accuracy.

In the case of email classification, the model can learn to recognize patterns and relevant features in the text of messages to distinguish between legitimate emails and spam. It uses information extracted from the text through vectorization [7] as input to the neural network, with the output layer indicating whether the message is classified as spam or non-spam.

Training is performed with a dataset that includes email messages previously labeled as spam or non-spam. During this process, the neural network adjusts its weights to find the best configuration that allows accurate predictions on unseen messages.

The model is trained using a dataset containing email messages labeled as either spam or non-spam. Throughout the training process, the neural network undergoes adjustments to its weights to find the optimal configuration for accurate predictions. ReLU [8] is employed as the activation function, and Adam is used as the optimizer. Furthermore, the network architecture incorporates a hidden layer with 100 neurons.

2.4 Results

The MLP has been chosen for implementation in the classification script due to its high performance in accuracy, precision, recall, and F1 score metrics, as evaluated using Scikit-Learn. These results are shown in Table 1. With an accuracy of 0.96, the model demonstrates a high success rate in classifying the data. Additionally, achieving a precision of 0.98 and a recall of 0.95 highlights its capability

to precisely identify positive instances while correctly capturing the majority of them. The F1 score of 0.96 demonstrates a good balance between precision and recall, indicating that the model is capable of classifying with high accuracy while effectively detecting positive cases.

Table 1. MLP Performance Metrics

Hidden Layers	Neurons in Each Hidden Layer	Accuracy	Precision	Recall	F1 Score
1	50	0.95	0.97	0.93	0.95
1	100	0.96	0.98	0.95	0.96
2	50	0.95	0.96	0.95	0.96
2	100	0.95	0.97	0.94	0.96
3	20	0.95	0.94	0.96	0.95
3	50	0.95	0.94	0.96	0.95

The analysis has been conducted using the MLP with a specific configuration of one layer and 100 neurons. As part of this analysis, a Receiver Operating Characteristic (ROC) curve has been generated (see Fig. 4a). Additionally, the performance was assessed using the confusion matrix (see Fig. 4b).

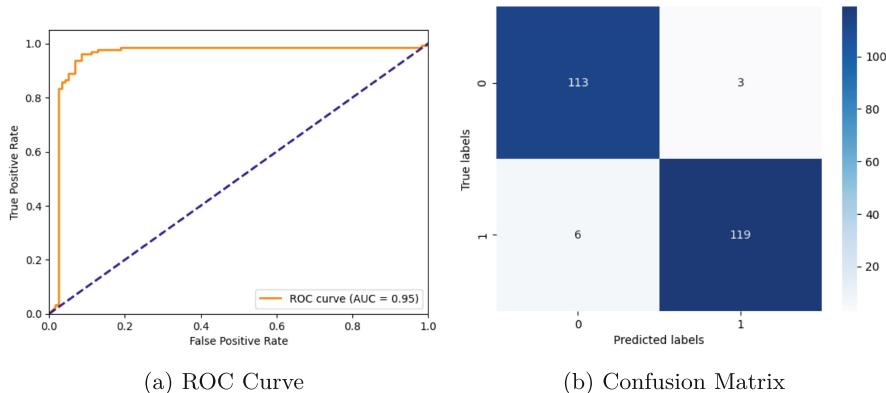


Fig. 4. MLP Performance Charts

The ROC curve is a visual representation that illustrates the relationship between the true positive rate and the false positive rate as the classification threshold varies. In this case, the ROC curve has been generated using the results of the MLP with the mentioned configuration. Upon analyzing the ROC curve, an Area Under the Curve (AUC) value of 0.95 has been obtained. The AUC is a measure of the model's discriminative ability, where a value of 1 indicates perfect classification and a value of 0.5 indicates random classification.

A confusion matrix is a performance measurement tool used in machine learning and statistics to evaluate the accuracy of a classification model. It is a table that summarizes the prediction results by comparing the true values with the model's predicted values. The chart typically consists of four main components: true positives, true negatives, false positives, and false negatives. These components help in understanding how well the model distinguishes between different classes and identify areas where the model may be making errors, providing insights into its precision, recall, and overall accuracy.

2.5 Discussion

The performance evaluation of the MLP with different configurations of hidden layers and neurons provides valuable insights into its effectiveness for this classification task.

Increasing the number of neurons in a single hidden layer generally enhances both accuracy and precision. For example, configurations featuring 100 neurons consistently achieved higher accuracy, reaching up to 0.96, compared to other setups. This indicates the model's improved ability to capture intricate data patterns.

Furthermore, variations in recall and F1 score across different configurations highlight nuances in the model's performance. Notably, configurations with multiple hidden layers exhibit stable metrics, suggesting robustness in capturing complex relationships within the dataset.

Overall, these findings underscore the adaptability of the MLP to diverse data complexities through its architecture, highlighting its suitability for tasks that demand nuanced classification capabilities.

3 Conclusions

The extensive adoption of email underscores the paramount importance of implementing robust security measures to shield users from various forms of cyber threats, with particular emphasis on phishing attacks. This study has made significant progress in enhancing the security of cloud-based email services by introducing an additional layer of defense aimed at identifying and thwarting phishing attempts.

The successful implementation of the neural network underscores the effectiveness of machine learning in detecting and classifying spam emails. By continuing to innovate and refine our approach to email security, leveraging cutting-edge technologies such as Jetson Nano, we can effectively safeguard users against the ever-evolving threat landscape of spam emails and phishing attacks, ensuring a safer and more secure online communication environment for all.

Exploring advanced machine learning algorithms and optimizing their implementation for resource-constrained environments could enhance the scalability and performance of email security solutions.

Moreover, extending the scope of this study to include other cloud email services and platforms would provide valuable insights into the generalization and applicability of the proposed defense mechanism. Additionally, investigating the integration of additional security features, such as sender authentication protocols and content analysis algorithms, could further enhance the overall security posture of cloud-based email services.

Acknowledgments. This research has been possible thanks to the Cybersecurity Chair of the University of La Laguna and INCIBE, and the research grant PID2022-138933OB-I00 funded by MCIN/AEI/ 10.13039/501100011033 and the European Union NextGenerationEU/PRTT.

References

1. Anti-Phishing Working Group. Phishing Activity Trends Report for 1st Quarter 2022, p. 3 (2022). <https://apwg.org/trendsreports>
2. Gawate, E., Rane, P.: Empowering Intelligent Manufacturing with the Potential of Edge Computing with NVIDIA's Jetson Nano. In: 2023 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, pp. 375–380 (2023). <https://doi.org/10.1109/ICCCIS60361.2023.10425278>
3. Musa, A., Aliyu, F.: Performance evaluation of multi-layer perceptron (MLP) and radial basis function (RBF). In: 2nd International Conference of the IEEE Nigeria Computer Chapter (NigeriaComputConf), Zaria, Nigeria 2019, pp. 1–5 (2019). <https://doi.org/10.1109/NigeriaComputConf45974.2019.8949669>
4. Apache Software Foundation. SpamAssassin (2024). <https://spamassassin.apache.org/>
5. Zhang, G., et al.: Downsampled and Undersampled datasets in feature selective validation (FSV). IEEE Trans. Electromagn. Compat. **56**(4), 817–824 (2014). <https://doi.org/10.1109/TEMC.2014.2307355>
6. Bakhsh, S.A., et al.: Enhancing IoT network security through deep learning-powered Intrusion Detection System. Internet Things **24**, 100936 (2023). <https://doi.org/10.1016/j.iot.2023.100936>
7. Hasani, M.F.: Investigating the combination between pre-processing technique and text vectorization for machine learning model in investor comment classification. In: International Workshop on Artificial Intelligence and Image Processing (IWAIIP), Yogyakarta, Indonesia 2023, pp. 278–282 (2023). <https://doi.org/10.1109/IWAIIP58158.2023.10462834>
8. Tachibana, K., Otsuka, K.: Wind prediction performance of complex neural network with ReLU activation function. In: Annual Conference of the Society of Instrument and Control Engineers of Japan (2018). <https://doi.org/10.23919/SICE.2018.8492660>



Decentralized Edge-Based Detection of Label Flipping Attacks in Federated Learning

Nourah AlOtaibi^{1,2(✉)}, Muhamad Felemban^{1,3,4}, and Sajjad Mahmood^{1,4}

¹ Information and Computer Science Department, KFUPM, Dhahran, Saudi Arabia
g201906790@kfupm.edu.sa

² Imam Abdulrahman Bin Faisal University (IAU), Dammam, Saudi Arabia

³ Computer Engineering Department, KFUPM, Dhahran, Saudi Arabia

⁴ Interdisciplinary Research Center for Intelligent Secure Systems,
Dhahran, Saudi Arabia

Abstract. Federated learning (FL) enhances data privacy by enabling decentralized model training without sharing local data. However, FL is vulnerable to Label-Flipping-Attacks (LFA), where malicious clients flip data labels. Traditionally, cloud-centric centralized defenses against LFA pose security and efficiency challenges. We propose a novel edge-based decentralized LFA detection method using edge servers for efficient detection. The proposed method considers the overall and class-wise accuracy in identifying suspicious clients. Initially, we adopt a strict *zero-tolerance approach* by excluding the entire update from detected malicious clients. We then experiment with four aggregation techniques—subtracting, masking, clipping, and reweighting—to handle the malicious parts of updates by focusing on the final layer neurons corresponding to specific classes. Experiments using three datasets demonstrate the effectiveness, robustness, and efficiency of our method, showing improved model performance and reduced latency under adversarial conditions. This approach improves the security and reliability of FL systems while maintaining data privacy.

Keywords: Aggregation · Edge intelligence · Federated learning · Poisoning attack · Security and privacy

1 Introduction

Federated learning (FL) has emerged as a machine learning paradigm that enables multiple devices to collaboratively train models locally without sharing data with a centralized server [9]. As a result, FL is used to enhance the privacy of sensitive data. However, recent studies showed that FL is still vulnerable to various adversarial attacks aimed at disrupting the training process [8].

One of the most insidious forms of attack in FL is the targeted poisoning attack, specifically the Label Flipping Attack (LFA) [12]. In such attacks, malicious clients intentionally alter their data labels to inject poisoned data into the model. The consequences of such attacks can cause the model to learn incorrect features, leading to inaccurate predictions and an overall degradation of model performance and reliability. Such attacks are catastrophic in critical FL applications such as healthcare and finance, and autonomous driving. For instance, in an autonomous driving scenario, a traffic light color LFA could cause the model to misclassify ‘red’ traffic lights as ‘green,’ leading to dangerous outcomes. This example underscores the severity of LFAs in real-world applications, highlighting the necessity for robust defense mechanisms in FL systems.

Traditional defense mechanisms are typically cloud-centric and depend on centralized data processing and analysis. This approach can risk data privacy and cause data transmission and processing bottlenecks [11]. Cloud-centric defense mechanisms can be found in both centralized FL and Client-Edge-Cloud FL architectures [3,14]. Although this approach mitigates the immediate threat, it comes at a significant cost, i.e. the loss of potentially valuable data from legitimate updates from those clients, which can affect the overall accuracy of the model [5].

The need for more refined and balanced defense mechanisms is evident. Effective strategies must distinguish between malicious and legitimate updates while ensuring that no legitimate updates are discarded. Such solution entails a paradigmic shift by implementing the detection at the edge. By moving the detection process to the edge servers, we not only enhance data privacy but also reduce latency. This decentralized approach uses the computational capabilities of edge servers to perform detection while ensuring timely and efficient identification of malicious clients.

To this end, we present a decentralized detection method at the edge that simultaneously considers both the overall and the class-wise accuracy of the updates submitted by the clients to identify suspicious clients. Our experiments on LFA reveal that model updates submitted by malicious clients are more distinguishable in some classes’ performance than overall accuracy. Initially, we adopt strict defense methods, a *zero-tolerance approach*, by excluding the entire update from detected malicious clients. Furthermore, we developed an improved solution that examines the gradients of malicious clients in the final layer of the neural network, where each neuron corresponds to a specific class. We incorporate four aggregation techniques: *subtracting, masking, clipping, and reweighting*. These techniques, inspired by state-of-the-art methods, are uniquely applied within our framework to neutralize the impact of malicious updates while preserving the integrity of legitimate data contributions, thus maintaining the robustness and accuracy of the FL model.

The contributions of this paper are as follows.

1. We introduce a decentralized edge-based detection method in Client-Edge-Cloud FL to enhance the data privacy and reduce the detection latency. By decentralizing the detection process to the edge, we minimize centralized data

- breaches and use class-wise performance metrics to effectively identify malicious clients using edge server capabilities to maintain FL system integrity.
2. We adopt and analyze four techniques—subtracting, masking, clipping, and reweighting—that manage malicious contributions without discarding entire updates, ensuring the robustness and accuracy of the FL model.
 3. A set of experiments using three datasets to demonstrate the effectiveness of the proposed detection mechanism. Performance evaluation results show an improved defense against poisoning attacks, faster edge-level operations, and reduced communication rounds.

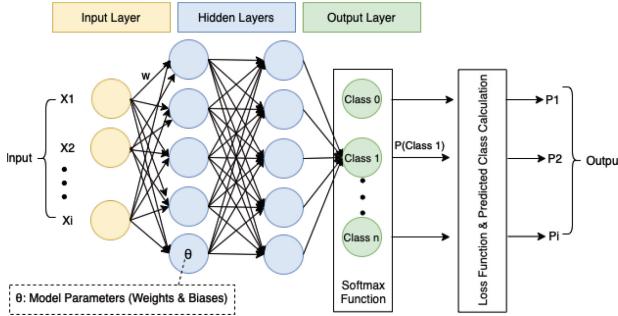


Fig. 1. A standard DNN consisting of an input layer, multiple hidden layers, and an output layer.

2 Background and Related Work

FL is based on the idea of collaborative training of a global model by optimizing the model parameters with local updates from several end users. Figure 1 illustrates a standard DNN model comprising multiple layers of nodes. Let D represent the training dataset, where each $x_i \in D$ consists of features f_i and an associated class label $class_i$. During the learning process, the model parameters f_i and the set of features θ are supplied to the model, producing predicted probabilities p_i . Each class probability $p_{c,i}$ is calculated by node n_c in the last layer, processed by the softmax function, producing a vector of confidence scores. The model predicts a probability distribution across all classes for a given input x_i . The loss function is calculated based on the difference between the predicted probability $p_{c,i}$ and the true label. The gradient of this loss with respect to the model parameters θ is then calculated. For each class c , the gradient $\nabla_{\theta} L(\theta; f_i, class_i)$ indicates the adjustments needed to minimize loss. These gradients are averaged across all samples to update the model parameters in a direction that reduces overall loss.

Existing Solutions: Poisoning attacks were identified as a significant threat well before FL emerged. Battista et al. showed that label flipping in the training set could severely degrade the accuracy of support vector machines (SVM) [1]. In FL, adversaries manipulate their local training data labels by using flipped labels instead of correct ones. This injects corrupted data into the FL process, disrupting the learning process and degrading the performance of the global model [4]. Numerous defenses have been proposed to counter such attacks. These include global aggregation methods, often called robust aggregation techniques [6, 7, 10], and local model assessment using anomaly detection methods. Anomaly detection strategies include approaches such as distance and clustering mechanisms [12, 15, 16]. Furthermore, some methods focus on evaluating the performance of local or global models using trusted datasets in the cloud [2, 3, 13].

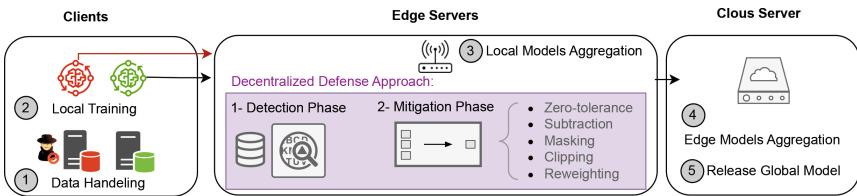


Fig. 2. Hierarchical FL System Framework: Illustrating a three-tier collaborative FL process with honest client models (green) and malicious client models (red). (Color figure online)

However, most current methods focus on cloud-centric solutions, which may lead to single central aggregation and longer latency. Our proposed defense addresses this gap by implementing detection closer to the source of the attack at the edge level within a Client-Edge-Cloud architecture. This approach improves data privacy, reduces latency, and utilizes the computational power of edge servers to quickly and efficiently identify malicious clients, thus ensuring the robustness and integrity of the FL model.

3 System Overview

3.1 Client-Edge-Cloud Framework

The Client-Edge-Cloud framework, illustrated in Fig. 2, includes multiple clients with some malicious clients (highlighted in red). The framework's hierarchical structure has the following: 1. **Clients:** Collect data and train local models using an initial model provided by the cloud. Clients periodically send updates to the edge servers. 2. **Edge servers:** Aggregate local updates, acting as intermediaries to produce the edge model. 3. **Cloud:** Combines aggregated edge models to create the global model, which is then redistributed to clients for the next training round. This hierarchical structure leverages the computational power of edge servers for efficient aggregation, improves data privacy, and reduces latency, ensuring the robustness and integrity of the FL model.

3.2 Attacker's Capability

A specific portion of the total clients act as malicious in our system. Malicious clients can only manipulate training data on their own devices. They have no control over the aggregation algorithm or honest clients' training process.

3.3 The Decentralized LFA Defense Approach

Edge-Detection Phase. The detection of malicious clients is employed at the edge within Client-Edge-Cloud framework. This involves validating the updates sent by clients during each training round and identifying suspicious updates based on their overall and class-wise performance (Algorithm 1).

Algorithm 1 Edge-Detection Phase

- 1: **Input:** Model updates from registered clients
- 2: **Output:** Identified malicious clients
- 3: Collect model updates from clients ▷ Edge Level
- 4: Evaluate updates using validation dataset at the edge
- 5: Calculate mean (μ_{overall}) and standard deviation (σ_{overall}) of overall accuracies

$$\tau_{\text{overall}} := \mu_{\text{overall}} - \sigma_{\text{overall}} \quad (1)$$

- 6: Calculate mean ($\mu_{\text{class-wise}}$) and standard deviation ($\sigma_{\text{class-wise}}$) of class-wise accuracies

$$\kappa_{\text{class-wise}} := \mu_{\text{class-wise}} - \sigma_{\text{class-wise}} \quad (2)$$

- 7: Identify suspicious clients based on their performance
-

The innovation of our approach lies in its decentralized application of detection at the edge and the integration of class-wise performance metrics.

Mitigation Phase: Initially, we implemented a strict defense strategy in the FL system by excluding the entire update from any client detected as malicious. This *zero-tolerance approach* ensured robust defense by maintaining the integrity of the FL system at various attack rates, effectively removing updates from identified malicious clients before edge aggregation.

Subsequently, we explore four different aggregation techniques. Instead of excluding updates from malicious clients entirely, these methods allowed us to process and combine updates in specific ways. This approach aims to mitigate the negative impact of excluding entire updates by selectively processing and integrating the valid portions of the updates from identified malicious clients. For clients identified as malicious ($i \in \mathcal{C}_m$), the malicious gradients $\mathbf{g}_{m,i}$ are

denoted as \mathcal{G}_m and the set of valid gradients as \mathcal{G}_v , which is aggregated using one of the four techniques shown in Fig. 3. *Gradient Subtraction* involves directly subtracting malicious gradients. *Gradient Masking* selectively nullifies malicious gradients. *Gradient Clipping* involves clipping malicious gradients to a very small value. *Gradient Reweighting* adjusts the contributions of gradients by assigning lower weights to malicious gradients. These techniques are used at the edge server utilizing the detection information, which depends on accurately isolating and mitigating the influence of malicious updates, ensuring the robustness and integrity of the global model.

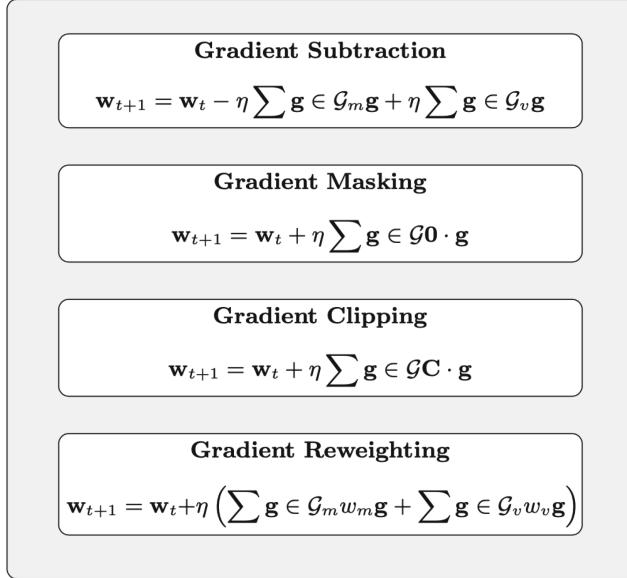


Fig. 3. Aggregation Techniques.

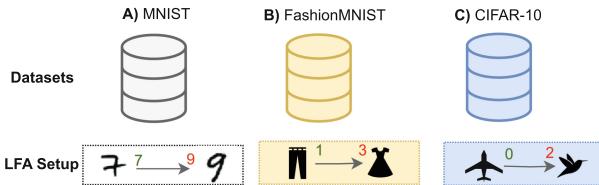


Fig. 4. LFA setup on experimented datasets.

4 Experiment Design

4.1 Datasets and LFA Setup

We conducted experiments on MNIST, Fashion-MNIST, and CIFAR-10 and different architectures for each dataset as in [12, 16]. Figure 4 shows the LFA setup.

Malicious clients replace instances of a source class (C_s) with a target class (C_t): digits 7 to 9 in MNIST, pants (1) to dresses (3) in Fashion-MNIST, and airplane (0) to bird (2) in CIFAR-10.

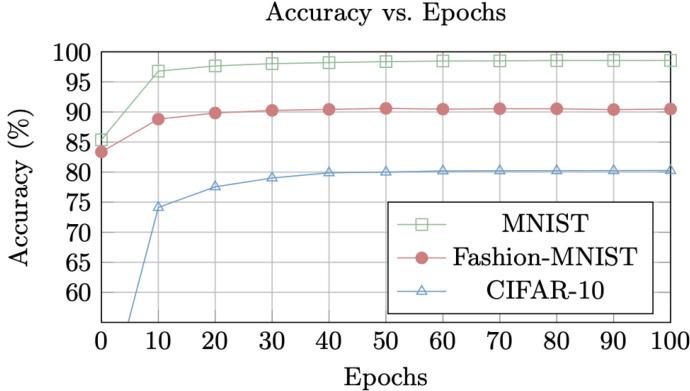


Fig. 5. Model accuracy on datasets over 100 training epochs.

4.2 Federated Learning Setup and Evaluation Metrics

The FL process was carried out in Python, using the following default parameters: a batch size of 20, 50 clients, a learning rate of 0.01, and 100 epochs. *Global Model Accuracy* (M^{ACC}) is defined as the percentage of instances in the test dataset D_{test} where the global model M with final parameters θ_R correctly predicts the true class label c_i of an instance x .

5 Result and Analysis

5.1 Standard Training Conditions

Establishing baseline model accuracy on datasets under standard training conditions provides a foundational benchmark to compare the impact of attack conditions on model performance across datasets. Figure 5 illustrates the baseline accuracy of the FL models in three datasets under standard training conditions.

The model initially improves slowly as it learns to distinguish the diverse and complex images. This slower improvement reflects the higher complexity of CIFAR-10 compared to MNIST (98.57%) and Fashion-MNIST (90.49%). However, the CIFAR-10 model achieves a notable maximum accuracy of approximately 80.26%, given the diversity and complexity of the dataset.

5.2 Attack Simulation Conditions

We follow the same assumptions as before [7, 16], which consider the number of malicious users to be less than half of the total of clients. Otherwise, it becomes impossible to reliably distinguish the honest gradients from the malicious ones without additional information.

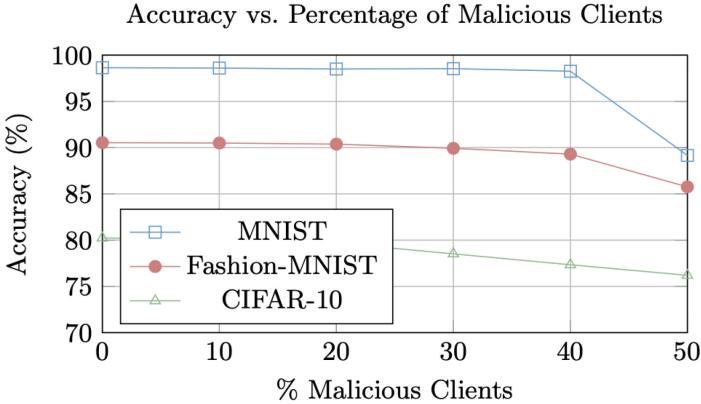


Fig. 6. Model accuracy on datasets with varying percentages of malicious clients.

Figure 6 demonstrates how the accuracy of the global model M^{ACC} changes as the percentage of malicious clients ranges from 0% to 50%.

5.3 Defense Simulation Conditions

Zero-Tolerance Approach. The strict nature of the zero-tolerance defense strategy is crucial in preserving the integrity of the aggregated model. The learning process remains on track with minimal interference by eliminating potentially poisoned data. However, this strict approach has been shown to be particularly severe in CIFAR-10.

Therefore, we analyze and aggregate the received updates instead of direct exclusion. This aggregation aims to mitigate the negative impact of excluding complete updates by selectively processing and integrating the valid portions of the updates from identified malicious clients. Four different aggregation techniques were experimented with the intent of maintaining high model accuracy while still effectively countering adversarial attacks. These techniques include subtraction, masking, clipping, and reweighting.

The impact of different defense mechanisms on model accuracy under various conditions for the datasets is summarized in Table 1. The results are presented for standard conditions, attack conditions (50%), the zero-tolerance approach, and four aggregation techniques: subtraction, masking, clipping, and reweighting.

Table 1. Impact of proposed approaches on model accuracy under 50% percentages of attack for MNIST, Fashion-MNIST, and CIFAR-10 datasets.

Attack Percentages	MNIST	Fashion-MNIST	CIFAR-10
Standard Conditions	98.57%	90.49%	80.26%
Attack Conditions (50%)	91.70%	85.57%	76.25%
Zero-tolerance Approach	93.50%	85.97%	75.51%
Subtraction	98.53%	89.46%	75.76%
Masking	98.56%	89.12%	75.75%
Clipping	98.57%	89.44%	75.74%
Reweighting	98.61%	89.67%	75.84%

The results indicate that the zero-tolerance approach, while effective in partially restoring accuracy for MNIST and Fashion-MNIST, appears too strict for CIFAR-10. The aggregation techniques, particularly reweighting, demonstrate superior performance across all datasets. These methods provide a more balanced approach by analyzing and aggregating updates rather than excluding them entirely, thereby preserving the integrity and accuracy of the FL system even under attack conditions.

For MNIST, the zero-tolerance approach improves attack conditions but is outperformed by all aggregation techniques. Clipping and reweighting techniques effectively restore accuracy to near-standard conditions, indicating their strong mitigation capabilities. For Fashion-MNIST, the zero-tolerance approach shows slight improvement, but aggregation techniques, especially reweighting, show significant improvements, almost restoring accuracy to standard conditions. CIFAR-10 remains the most challenging dataset, likely due to its complexity and the critical input from each client. The zero-tolerance approach is less effective, slightly degrading accuracy further. However, the aggregation techniques offer improvements, with reweighting showing the best results.

6 Conclusion and Future Works

This study tackled the challenge of LFA in FL systems. We proposed an edge-based detection method within a Client-Edge-Cloud FL architecture, leveraging edge servers to detect and mitigate malicious clients efficiently. This decentralized approach preserves data privacy and reduces latency. Our method employs four aggregation techniques—subtracting, masking, clipping, and reweighting—that handle malicious gradients selectively rather than discarding entire updates. We achieve a more refined and effective defense by focusing on specific class labels in the final layer neurons. Comprehensive experiments on MNIST, Fashion-MNIST, and CIFAR-10 datasets validate our method, demonstrating significant improvements in model performance and robustness under adversarial conditions.

For future work, we plan to enhance our proposed approach by using techniques such as uniform reweighting or aggregate normalized gradients to make the process more generalizable. Additionally, we will investigate the effectiveness of our approach against more stealthy and sophisticated attacks. Expanding the scope to non-IID data distributions will also be a priority, ensuring the broader applicability of our defense mechanisms in diverse FL environments.

References

1. Biggio, B., Nelson, B., Laskov, P.: Poisoning attacks against support vector machines. arXiv preprint [arXiv:1206.6389](https://arxiv.org/abs/1206.6389) (2012)
2. Cao, X., Zhang, Z., Jia, J., Gong, N.Z.: Flcert: provably secure federated learning against poisoning attacks. *IEEE Trans. Inf. Forensics Secur.* **17**, 3691–3705 (2022)
3. Fang, M., Cao, X., Jia, J., Gong, N.: Local model poisoning attacks to {Byzantine-Robust} federated learning. In: 29th USENIX security symposium (USENIX Security 2020), pp. 1605–1622 (2020)
4. Jebreel, N.M., Domingo-Ferrer, J., Sánchez, D., Blanco-Justicia, A.: Lfighter: defending against the label-flipping attack in federated learning. *Neural Netw.* **170**, 111–126 (2024)
5. Kairouz, P., et al.: Advances and open problems in federated learning. *Found. Trends® Mach. Learn.* **14**(1–2), 1–210 (2021)
6. Li, D., Wong, W.E., Wang, Yao, Y., Chau, M.: Detection and mitigation of label-flipping attacks in federated learning systems with kpca and k-means. In: 2021 8th International Conference on Dependable Systems and Their Applications (DSA), pp. 551–559. IEEE (2021)
7. Liu, X., Li, H., Xu, G., Chen, Z., Huang, X., Lu, R.: Privacy-enhanced federated learning against poisoning adversaries. *IEEE Trans. Inf. Forensics Secur.* **16**, 4574–4588 (2021)
8. Lyu, L., et al.: Privacy and robustness in federated learning: attacks and defenses. *IEEE Trans. Neural Netw. Learn. Syst.* (2022)
9. McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, pp. 1273–1282. PMLR (2017)
10. Muñoz-González, L., Co, K.T., Lupu, E.C.: Byzantine-robust federated machine learning through adaptive model averaging. arXiv preprint [arXiv:1909.05125](https://arxiv.org/abs/1909.05125) (2019)
11. Ren, J., Yu, G., He, Y., Li, G.Y.: Collaborative cloud and edge computing for latency minimization. *IEEE Trans. Veh. Technol.* **68**(5), 5031–5044 (2019)
12. Tolpegin, V., Truex, S., Gursoy, M.E., Liu, L.: Data poisoning attacks against federated learning systems. In: Chen, L., Li, N., Liang, K., Schneider, S. (eds.) ESORICS 2020. LNCS, vol. 12308, pp. 480–501. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58951-6_24
13. Upadhyay, A., Rawat, D.B.: Mitigating poisoning attack in federated learning. In: 2021 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 01–07. IEEE (2021)
14. Wang, Y., Zhu, T., Chang, W., Shen, S., Ren, W.: Model poisoning defense on federated learning: a validation based approach. In: International Conference on Network and System Security, pp. 207–223. Springer, Heidelberg (2020). https://doi.org/10.1007/978-3-030-65745-1_12

15. Xia, G., Chen, J., Yu, C., Ma, J.: Poisoning attacks in federated learning: a survey. *IEEE Access* **11**, 10708–10722 (2023)
16. Zhao, Y., Zhang, J., Cao, Y.: Manipulating vulnerability: poisoning attacks and countermeasures in federated cloud-edge-client learning for image classification. *Knowl.-Based Syst.* **259**, 110072 (2023)



Analysis of Post-Quantum Cryptographic Algorithms: A Systematic Literature Review

Francisco Javier de la Puente Secades^{1(✉)}, Gonzalo Esteban-Costales², Adrián Campazas-Vega², Claudia Álvarez-Aparicio², Vicente Matellán-Olivera², and Ángel Manuel Guerrero-Higueras²

¹ Supercomputación Castilla y León, Edificio CRAI-TIC. Campus de Vegazana, s/n, 24071 León, Spain

francisco.delapuente@scayle.es

² Robotics Group, University of León, Edificio MIC. Campus de Vegazana, s/n, 24071 León, Spain

<https://www.scayle.es>

Abstract. Due to the significant advancement in quantum computational capability, there has been a race to develop algorithms capable of withstanding attacks stemming from this emerging technology. In response to this need, the National Institute of Standards and Technology of the United States initiated a standardization process for such algorithms since 2016. However, the academic literature still does not provide comprehensive coverage on the number of published algorithms nor their comparison with other reports in terms of security and efficiency. To address this lack of information, this study presents a systematic literature review up to January 2024. As a result of this analysis, a total of 69 different algorithms and 7 security incident articles concerning them have been identified. These findings serve as a reference point for future research and developments in the field of post-quantum cryptography, especially regarding quantum key distribution and digital signature authentication.

Keywords: Algorithm · Key-Encapsulation Mechanism · Post-Quantum Cryptography · Public Key Encryption · Systematic Mapping

1 Introduction

In recent decades, the field of cryptography has experienced significant growth, especially with the advent of quantum computers, machines that exploit the principles of quantum mechanics to perform calculations exponentially faster than classical computers [7]. The prospect of large-scale quantum computing poses significant challenges to information security, as many current cryptographic systems become vulnerable to quantum algorithms [17]. To address this concern,

the field of post-quantum cryptography has emerged, aiming to develop cryptographic systems resilient to both quantum and classical computing attacks, and compatible with existing communication infrastructures [7].

Academic research on the potential impact of quantum computing dates back to at least 2001 [19]. In April 2016, the National Institute of Standards and Technology (NIST) of the United States published a report raising concerns that by the year 2030, quantum technology could jeopardize the security of the RSA (Rivest, Shamir, Adleman) algorithm, widely used today [1]. This forecast has sparked growing interest in the development of cryptographic primitives that are secure against quantum attacks. While symmetric primitives can be modified to withstand quantum computing, the primary focus has been on public-key cryptography, especially digital signatures and key-encapsulation mechanisms [12].

In December 2016, NIST initiated a standardization process by soliciting proposals for post-quantum algorithms [4]. This process has led to a significant increase in the number of scientific articles published in recent years addressing topics related to post-quantum cryptography. The standardization of these algorithms is crucial to ensure interoperability and security of cryptographic systems in an environment increasingly threatened by advancements in quantum computing.

The objective of this study is to conduct a comprehensive review of the existing literature on post-quantum cryptography up to the present date. Specifically, the aim is to identify relevant algorithms applicable to this area of cryptography. To achieve this, a systematic literature mapping will be conducted, following the guidelines proposed by Kitchenham, Budgen, and Brereton [13]. Through this analysis, we seek to address the following research question:

- Which algorithms have been published and are applicable for post-quantum cryptography?

To comprehensively break down the response, the rest of the document will follow the general structure outlined in the PRISMA guideline [15] for conducting systematic reviews. Section 2 provides a brief introduction to both quantum cryptography and post-quantum algorithms, while Sect. 3 summarizes the planned review protocol for this study. Section 4 presents the results derived from the review process. Drawing on the gathered evidence, Sect. 5 addresses the research question posed and highlights limitations found in this analysis. Finally, the conclusions and potential applications of this review are detailed in Sect. 6.

2 Context

The study of cryptography dates back to ancient times, but with the advancements in computing during the 20th century, new concerns about information security emerged, necessitating the protection of sensitive data from attacks. In the 1970s, NIST proposed standardizing encryption methods, leading to the

development of the Data Encryption Standard (DES) [2]. However, DES's small key size made it vulnerable, prompting the search for more secure alternatives.

The rise of quantum computers introduces significant challenges to information security. These machines exploit quantum mechanics to perform calculations much faster than classical computers, potentially compromising existing cryptographic systems [14]. In response, the field of post-quantum cryptography has emerged to develop systems resilient to both quantum and classical attacks.

Within this context, two key approaches are prominent: Public Key Cryptography (PKC) and Quantum Key Distribution (QKD). PKC relies on the difficulty of mathematical problems for security, yet algorithms such as RSA and ECC are vulnerable to quantum attacks, leading to research in post-quantum alternatives like Lattice-Based Cryptography (NTRU) and the Lamport-Diffie Hash Signature Scheme. Conversely, QKD utilizes quantum principles to secure communication through quantum key distribution, providing a potentially immune solution to quantum threats [7].

This review aims to critically analyze the existing literature on post-quantum cryptography to identify relevant algorithms and evaluate their suitability in a rapidly evolving cybersecurity landscape. Such assessment is crucial for understanding the challenges and opportunities in post-quantum cryptography and guiding future research in this dynamic field.

3 Methodology

In the present study, the methodology proposed by Kitchenham, Budgen, and Brereton [13] has been adopted, recognized for its applicability in conducting systematic reviews in the field of computer science. The specific purpose of this review lies in obtaining a comprehensive and concise overview of the current state of specialized literature, both in scientific journals and academic conferences, regarding post-quantum cryptographic algorithms. Therefore, any article presenting a detailed analysis of such algorithms, covering both their technical aspects and vulnerabilities studied, has been considered. The evaluation parameters focus on these fundamental characteristics and how they amalgamate to form a coherent and functional algorithm.

3.1 Search Process

The research protocol was outlined based on a corpus of semi-reference documents (see Table 1), grounded in specialized literature in the field, although informally acquired prior to the start of the systematic review. Considering this prelude, an automated search strategy was designed that spanned up to January 2024, using as primary sources the electronic databases: IEEE Xplore Digital Library, Scopus, and Web of Science. A comprehensive search for scientific articles from journals and conferences, written in English, was conducted on all these platforms. Additionally, these inquiries were confined to the fields of document titles, abstracts, and keywords.

Table 1. Set of semi-reference articles.

Reference	Title
[7]	A Survey of Post-Quantum Cryptography: Start of a New Race
[12]	Recent Trends on Research and Development of Quantum Computers and Standardization of Post-Quantum Cryptography
[5]	Status report on the third round of the NIST Post-Quantum Cryptography Standardization process
[6]	A survey on NIST PQ signatures
[17]	Security Comparisons and Performance Analyses of Post-quantum Signature Algorithms
[10]	A Survey on NIST Selected Third Round Candidates for Post Quantum Cryptography
[14]	Post Quantum Cryptography(PQC) - An overview
[4]	Status report on the first round of the NIST post-quantum cryptography standardization process

A search string was developed to address the research question, based on keywords from key documents identified during the iterative search for semi-reference articles. The search string is (*algorithm*) AND (*post-quantum*) OR *pqc* OR *quantum safe* OR *quantum-proof* OR *quantum-resistant*) AND (*key-encapsulation mechanism* OR *kem*) AND (“*public-key encryption*”).

To validate this string and the selection process, a criterion was established requiring at least 50% of articles to be included in the automated search results. Out of 8 semi-reference articles, 3 were found, corresponding to 37.5%. This percentage is justified because two of the articles are NIST reports [4,5] that have not undergone peer review. Excluding these, the appearance rate is 3 out of 6 articles, or 50%.

Additionally, to complement the automated search, a review of the bibliographic references of selected primary articles was conducted. These articles successfully passed the evaluation process and provide relevant answers to the research question.

3.2 Selection Process

To refine the results obtained in the searches, a selection process was devised supported by a set of inclusion and exclusion criteria detailed in Table 2. This process was conceived as a single phase, anticipating a reduced number of results due to the relatively new area of research addressed. Additionally, this decision was based on the consideration that defining the specific characteristics of the documents might be secondary compared to the hypothesis posed in the article. In this sense, evaluation based solely on the title and abstract might not be

sufficient to make an informed decision. Therefore, in the first instance, duplicate articles were discarded, and then copies of the remaining articles were obtained to conduct a thorough review of their textual content.

Table 2. Set of inclusion and exclusion criteria.

Criterion Type	Description
Inclusion	Contains characteristics describing an algorithm
	Contains an analysis of an algorithm vulnerability
	Contains an analysis of an algorithm's security
Exclusion	The described algorithm is not related to post-quantum
	Describes cryptographic schemes but not algorithms
	Does not contain characteristics describing an algorithm

3.3 Quality Assessment Process

Given the planning of the selection process, it was determined to implement a quality assessment system to be highly selective. In this regard, a quality score was assigned to each article, and a minimum threshold was established below which its inclusion in the study would be discarded. To carry out this assessment systematically, a checklist was designed that included the following key questions:

1. Does it include technical and functional aspects of the cryptographic system?
2. Does it report legal statements regarding the intellectual property of the cryptographic system?
3. Are tests conducted on the algorithm's operation?
4. Does it include implementations in C code or pseudo-code of the algorithm?
5. Does it report security features, cost/efficiency, or implementation characteristics of the algorithm?

Each question posed was addressed with a clear response, classified as “Yes”, “No” or “Partially”. Additionally, to quantify the quality of each evaluated article, numerical values were assigned to these responses: “1” for “Yes”, “0” for “No” and “0.5” for “Partially”. Thus, a maximum score of 5 points was established for each article, with a quality threshold set at 2.5 points. Consequently, articles that did not reach this threshold were excluded from the review process.

3.4 Data Extraction and Synthesis Processes

To address the research questions posed, a process involving data extraction via an electronic form was devised, using the web tool Parsifal¹. The elements to

¹ <https://parsif.al>.

be extracted from each article include: the bibliographic reference, the name of the algorithm, categorization within the Post-Quantum Cryptography (PQC) family, information related to the cost, performance, and/or implementation of the algorithm, associated security incidents, as well as its applicability in the context of Quantum Key Distribution (QKD).

4 Results

The outcome of the search and selection process is summarized in the PRISMA diagram in Fig. 1.

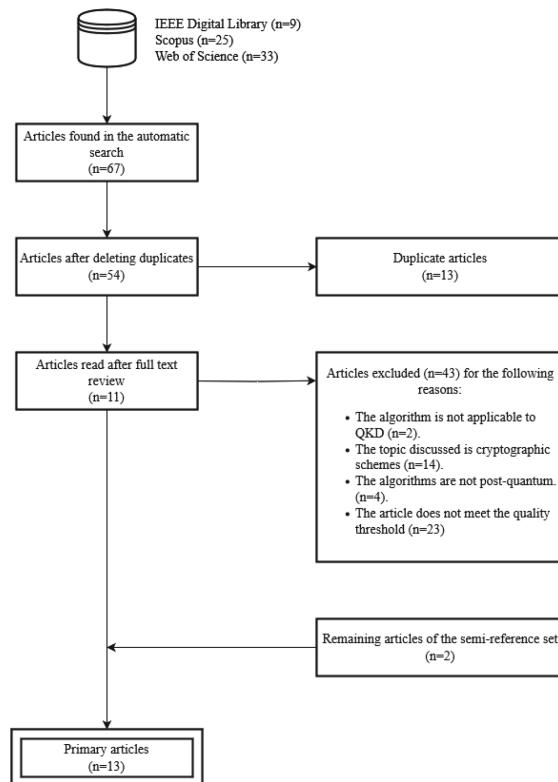


Fig. 1. Diagramma PRISMA

A total of 67 articles were identified through the automated search in the pre-defined sources of information, to which the articles obtained as semi-reference (Table 1) were added. The validity of the search was confirmed by finding 3 of the articles from the semi-reference set among the obtained results. In the selection process, initially 13 duplicate articles were eliminated, after which the

remaining 54 were reviewed. Of these, 43 were discarded due to exclusion criteria (20) or for not meeting the established quality threshold (23). As a result, the set of primary articles was composed of 13 articles. A complementary search of the references of these 13 primary articles was then carried out, limited to the analysis of titles and abstracts. However, no additional relevant references were identified. Consequently, the total number of selected primary articles was 13, which have been uniquely numbered following the convention S1 to S13. It is worth noting that all the articles included in this set were published between the years 2019 and 2023 (see Fig. 2).

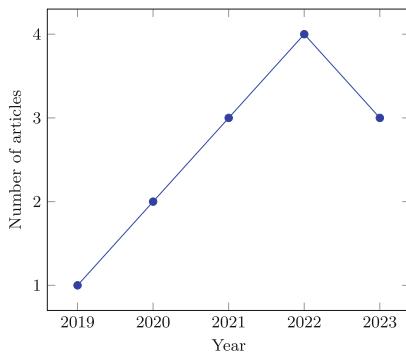


Fig. 2. Trend graph of selected articles

Regarding the quality assessment, Table 3 presents the scores obtained for the 13 selected primary articles. These articles are the ones that exceeded the

Table 3. Quality results of primary articles.

Reference	Work ID	Score
[17]	S1	3.0
[10]	S2	2.5
[16]	S3	2.5
[4]	S4	4.0
[5]	S5	3.5
[8]	S6	2.5
[3]	S7	3.0
[11]	S8	2.5
[18]	S9	2.5
[9]	S10	2.5
[14]	S11	2.5
[12]	S12	3.0
[6]	S13	2.5

selected quality threshold. It is worth noting that article S4 was the only one to achieve a rating of 4.0 points in the quality assessment.

Regarding data synthesis, Table 4 contains all extracted information related to types and families belonging to post-quantum cryptography algorithms.

Table 4. Post-quantum cryptography algorithms

Type	Public Key Encryption		Signature
Lattice-based	Compact LWE	LOTUS	CRYSTALS-DILITHIUM
	CRYSTALS-KYBER	NewHope	DRS
	Ding Key-Exchange	NTRUEncrypt	FALCON
	EMBLEM and R.EMBLEM	NTRU-HRSS-KEM	pqNTRUSign
	FrodoKEM	NTRU Prime	qTESLA
	KCL	Odd Manhattan	
	KINDI	Round5	
	LAC	SABER	
	LIMA	Three Bears	
	Lizard	Titanium	
Code-based	BIG QUAKE	McNie	pqsigRM
	BIKE	NTS-KEM	RaCoSS
	Classic McEliece	ROLLO	
	DAGS	QC-MDPC KEM	
	HQC	Ramstake	
	LEDAkem	RLCE-KEM	
	LEDApkc	RQC	
	Lepton		
Multivariate	CFPKM	DME	DME
	Giophantus		DualModeMS
			GeMSS
			Gui
			HiMQ-3
			LUOV
			MQDSS
Hash-based			Rainbow
			Gravity-SPHINCS
Supersingular elliptic curve isogeny	SIKE		SPHINCS+
Satirical submission	pqRSA	Picnic pqRSA	
Braid group		WalnutDSA	
Others	Guess Again Mersenne-756839		

5 Discussion

5.1 Which Algorithms Have Been Published and Are Applicable for Post-Quantum Cryptography?

This study aims to elucidate the existing algorithms suitable for post-quantum cryptography. From the analysis of 13 primary articles, we identified a total of 69 algorithms, as summarized in Table 4. This total corresponds with the algorithms highlighted in the first round of NIST's post-quantum cryptography standardization process [4], establishing a reference point for future applications in this emerging field.

Crystals-Kyber is notably the most frequently mentioned algorithm, cited 8 times (S1, S4, S5, S6, S7, S10, S11, S12), followed by Crystals-Dilithium with 4 mentions (S4, S5, S6, S12), and both HQC and Falcon, which appear 3 times each (S4, S5, S7 for HQC; S4, S5, S12 for Falcon). The prevalence of these algorithms can be attributed to their advancement to the third round of NIST standardization [5], which emphasizes their significance in current research.

The identified algorithms fall into two principal categories: Key Encapsulation Mechanism (KEM) and Signature. Within the KEM category, Crystals-Kyber is the most referenced, while Crystals-Dilithium leads in the Signature category. This prominence stems from their selection as candidates for standardization by NIST [5]. Importantly, Crystals-Kyber is the only KEM proposed for standardization, whereas in the Signature category, Crystals-Dilithium is joined by Falcon and SPHINCS+. Although Falcon is referenced in two of the analyzed articles, SPHINCS+ was not mentioned at all.

Furthermore, these algorithms can be subclassified into specific families of post-quantum cryptography (PQC), including Code-based, Hash-based, Isogeny-based, Lattice-based, and Multivariate schemes. The Code-based and Lattice-based families dominate the literature, appearing 3 times (S4, S5, S12) and 6 times (S1, S5, S7, S10, S11, S12) respectively. This prevalence underscores the reliability these families offer in terms of cryptographic security, with the lattice-based approach particularly favored due to extensive research supporting it.

In terms of security incidents, 7 articles discuss vulnerabilities (S2, S3, S5, S6, S7, S10). Three of these specifically analyze Crystals-Kyber (S2, S5, S7), while side-channel attacks are a focus in three articles as well (S2, S5, S7). This pattern reflects the algorithm's prominence as NIST's sole candidate for the KEM category, highlighting side-channel attacks as a critical concern for the security of these cryptographic systems.

Regarding computational efficiency, the algorithms with the highest citations typically demonstrate lower clock cycle requirements during execution [5, 14]. Among these, KYBER, DILITHIUM, and FALCON stand out, with citations of 8 (S1, S4, S5, S6, S7, S10, S11, S12), 4 (S4, S5, S6, S12), and 3 (S4, S5, S12) respectively. This trend aligns with NIST's evaluation in the third phase of candidate selection.

Crystals-Kyber is particularly noteworthy as a leading KEM due to its basis in the Module-LWE problem, which allows for small key sizes-1,632 bytes for its

basic version (Kyber512)-enhancing performance in high-speed environments. In the signature domain, Crystals-Dilithium employs the LWE and SIS problems, with public key sizes ranging from 1,312 to 2,592 bytes, demonstrating significant efficiency in signature verification.

Falcon is another key contender in digital signatures, utilizing the NTRU lattice structure, which provides substantial resistance against quantum attacks. Its public keys are compact, reaching only 897 bytes in the Falcon-512 version, making it an attractive option for applications requiring both security and efficiency.

Additionally, SPHINCS+ is recognized for its innovative hash-based signature generation method, which offers strong security guarantees, albeit with larger signature sizes compared to lattice-based schemes. Its design emphasizes statelessness, which is advantageous for specific applications that prioritize security over performance.

Overall, these findings highlight the diversity and robustness of post-quantum cryptographic algorithms, underscoring their critical role in securing communications against future quantum threats.

5.2 Limitations of the Review

Given the nature of the review, which was conceived to provide a broad and expedient overview of contemporary evidence, it's important to consider certain limitations that may primarily affect the comprehensiveness of the study. In this regard, imposed restrictions, such as language (limited to English articles) and the exclusion of books and grey literature, except for those articles considered semi-reference, entail the risk of not incorporating relevant work. Ideally, the processes of selection, quality assessment, and data extraction could be validated through a test-retest study conducted at a later time after the conclusion of the review. An area for improvement lies in expanding the assessment of study quality. It's suggested to broaden the set of questions proposed in Sect. 3.2, aimed at evaluating possible biases in the assessment of feature sets, as well as including inquiries related to the suitability of the algorithm for use in quantum key exchange, and the existence of tests conducted in real-world settings.

6 Conclusions

Quantum cryptography and quantum algorithms represent a promising area in information security, focused on ensuring the confidentiality and integrity of communications. This research examined key characteristics of these algorithms, particularly their resilience against quantum computer attacks and their ability to protect sensitive data.

We identified various defining features of quantum algorithms, including encoding, security, efficiency, and cost, which are crucial for developing advanced quantum cryptography systems that can safeguard information from threats.

To identify suitable algorithms for quantum cryptography, we conducted a comprehensive literature review, revealing 69 relevant algorithms. These were

assessed based on computational efficiency in key generation and operation, as well as security against brute-force attacks, with 6 articles discussing related security incidents.

The findings provide a solid foundation for future research in quantum cryptography. The list of identified algorithms serves as a starting point for practical applications, while the 13 primary articles analyzed offer references for evaluating the effectiveness of different approaches.

Looking forward, ongoing research is essential for refining these algorithms in light of evolving quantum threats. The regulatory framework established by NIST has heightened the focus on post-quantum cryptography, yet new attack vectors continue to emerge, raising questions about the future of standardization.

Adopting multiple models, similar to DES and AES, may be necessary to meet diverse application requirements and threat scenarios. A variety of cryptographic approaches will be vital for developing adaptive strategies that address current standards and proactively counter emerging vulnerabilities. Therefore, the path ahead requires balancing innovation with rigorous validation to ensure the resilience of quantum cryptography against an unpredictable future.

In conclusion, this study enhances knowledge in quantum cryptography by providing a comprehensive overview of available algorithms and their characteristics, potentially driving the development of more robust and reliable security systems for secure communications.

Acknowledgments. This publication is part of the project CINDERELLA-QKD, financed by ‘European Union NextGeneration-EU, the Recovery Plan, Transformation and Resilience, through INCIBE’.

References

1. Nist released nistir 8105, report on post-quantum cryptography. Technical report (2016). Accessed 5 Nov 2019
2. 46-3, F.: The official document describing the des standard. <http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf>
3. Aguilar-Melchor, C., Aragon, N., Bellini, E., Caullery, F., Makarim, R.H., Marcolla, C.: Constant time algorithms for rollo-i-128. *SN Comput. Sci.* **2** (2021)
4. Alagic, G., et al.: Status report on the first round of the nist post-quantum cryptography standardization process. Technical report (2019)
5. Alagic, G., et al.: Status report on the third round of the nist post-quantum cryptography standardization process. Technical report (2022)
6. Chiano, N.D., Longo, R., Meneghetti, A., Santilli, G.: A survey on nist pq signatures (2021)
7. Dam, D.T., Tran, T.H., Hoang, V.P., Pham, C.K., Hoang, T.T.: A survey of post-quantum cryptography: start of a new race. *Cryptography* **7**(3) (2023)
8. Ferradi, H., Naccache, D.: Integer reconstruction public-key encryption. In: Mu, Y., Deng, R.H., Huang, X. (eds.) CANS 2019. LNCS, vol. 11829, pp. 412–433. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-31578-8_23

9. Jiang, H., Zhang, Z., Ma, Z.: Key encapsulation mechanism with explicit rejection in the quantum random oracle model. In: Lin, D., Sako, K. (eds.) PKC 2019. LNCS, vol. 11443, pp. 618–645. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17259-6_21
10. Hasija, T., Ramkumar, K.R., Kaur, A., Mittal, S., Singh, B.: A survey on nist selected third round candidates for post quantum cryptography. In: 7th International Conference on Communication and Electronics Systems, ICCES 2022 - Proceedings, pp. 737–743 (2022)
11. Jiang, H., Han, J., Zhang, Z., Ma, Z., Wang, H.: Practical algorithm substitution attacks on real-world public-key cryptosystems. IEEE Trans. Inf. Forensics Secur. **18**, 5069–5081 (2023)
12. Kan, K., Une, M.: Recent trends on research and development of quantum computers and standardization of post-quantum cryptography (2022)
13. Kitchenham, B.A., Budgen, D., Brereton, P.: Evidence-Based Software Engineering and Systematic Reviews, vol. 4. CRC Press, Boca Raton (2016)
14. Kumar, M., Pattnaik, P.: Post quantum cryptography(pqc) - an overview. In: 2020 IEEE High Performance Extreme Computing Conference (HPEC) (2020), Virtual, 21–25 September 2020 (2020)
15. Moher, D., Liberati, A., Tetzlaff, J., Altman, D.G., et al.: Preferred reporting items for systematic reviews and meta-analyses: the prisma statement. PLoS Med. **6**(7), e1000,097 (2009)
16. Nosouhi, M.R., et al.: Weak-key analysis for bike post-quantum key encapsulation mechanism. IEEE Trans. Inf. Forensics Secur. **18**, 2160–2174 (2023)
17. Raavi, M., Wuthier, S., Chandramouli, P., Balytskyi, Y., Zhou, X., Chang, S.-Y.: Security comparisons and performance analyses of post-quantum signature algorithms. In: Sako, K., Tippenhauer, N.O. (eds.) ACNS 2021. LNCS, vol. 12727, pp. 424–447. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-78375-4_17
18. Xu, S., Sun, A., Ren, Z., Zhao, Y., Ni, Q., Tong, Y.: Enhanced post-quantum key escrow system for supervised data conflict of interest based on consortium blockchain. J. Comb. Optim. **45**, 116 (2023)
19. Zhu, H.: Survey of computational assumptions used in cryptography broken or not by shor's algorithm. Technical report (2001)

CISIS: Special Session: Artificial Intelligence for Protecting the Internet of Things



A Comparison of AI-Enabled Techniques for the Detection of Attacks in IoT Devices

Eduardo Manuel Cabeza-Lopez¹ , Ruben Ruiz-Gonzalez¹ , Alejandro Merino-Gomez¹ , Leticia Elena Curiel-Herrera² , and Jaime Andres Rincon³

¹ Departamento de Digitalización, Escuela Politécnica Superior, Universidad de Burgos, Avda. Cantabria s/n, 09006, Burgos Burgos, Spain
`ecl1009@alu.ubu.es, {ruben.ruiz,alejandromg}@ubu.es`

² Departamento de Ingeniería Informática, Escuela Politécnica Superior, Universidad de Burgos, Avda. Cantabria s/n, 09006, Burgos Burgos, Spain
`lcuriel@ubu.es`

³ Departamento de Digitalización, Escuela Politécnica Superior, Universidad de Burgos Ctra,Orón, 28, 09200 Miranda de Ebro, Burgos, Spain
`jarincon@ubu.es`

Abstract. The purpose of this research is to enhance the security of Internet of Things devices. The deployment of these gadgets has increased exponentially, nowadays they can be found in every sector from industrial environments to applications in residential homes. This technology is important for enterprises due to the ability to manage different kind of data and control critical operations in an efficiently way. Consequently, these devices have become frequent targets of cyberattacks, highlighting the necessity of robust detection methods. However it is a complicated task to find a uniform security solution due to the variety of devices, with different capabilities and requirements. This study evaluates some artificial intelligence supervised learning algorithms such as XGBoost and Random Forest using the TON_IoT dataset, which are new generations of Internet of Things and Industrial datasets for evaluating the fidelity and the efficiency of different cybersecurity applications based on Artificial Intelligence. The analysis shows both algorithms are effective in detecting cyberattacks, achieving accuracies close to 98%, with a minimal variation in terms of precision and recall. These algorithms outperform the results obtained in previous works with the same dataset, then they can provide an additional security layer, allowing accurate identification of potential attacks. This research shows the importance of artificial intelligence algorithms in cybersecurity and their potential to improve the protection of this kind of devices.

Keywords: Artificial Intelligence (AI) · Internet of Things (IoT) · XGBoost · Random Forests · attack detection · TON_IoT dataset

1 Introduction

There has been significant growth in the Internet of Things (IoT) in recent years [1], and this growth is likely to continue in the future. In moderately developed countries, the use of these devices, both active and passive, has become increasingly common. The rapid spread of IoT technologies has increased convenience and efficiency. However, it has also created critical security and privacy risks for its users. IoT networks have become a target for cybercriminals [4], and the vulnerabilities of these devices are many and varied, such as relying on proprietary operating systems, not updating software, transmitting data without proper encryption, connecting to open networks, or frequently changing network connections [13], and so on. In addition, the limited computing resources of IoT devices, combined with the mobile nature of these devices, amplify their risks.

The rest of the paper is organized as follows. In Sect. 2 machine learning for cybersecurity in IoT will be analyzed, focusing on models and datasets. In Sect. 3 a detailed problem description will be provided. Following this, the software specifications of the technologies employed will be outlined. Subsequently, the experimentation conducted will be delved into, and the results obtained will be presented. In Sect. 4, the results obtained from the different experiments are presented. Finally, conclusions are presented in Sect. 5, laying the groundwork for future research.

2 Related Work

This section presents a brief overview of the current state of the art in the application of machine learning techniques for the protection of IoT devices. A general overview of some of the work done in recent years will be provided to give context on the current landscape of the available literature on this topic. The work presented by Abdullah Alsaedi et al. [2] highlights the lack of datasets on IoT device performance and network traffic. Some of these datasets were analyzed and used for training various models; however, the obtained results were not always as expected. This underscores the need for more robust and representative datasets.

Complementarily, the research conducted by Nour Moustafa [15] reaffirms the findings of Abdullah Alsaedi et al. Moustafa identifies the absence of distributed architectures necessary to generate high-quality datasets that accurately capture the real behaviors of IoT networks in complex scenarios. In his work, Moustafa validates the “The TON_IoT datasets” [14], emphasizing their utility in addressing these challenges. Abdallah R. Gad et al. [11] propose enhancing the security of devices used for intelligent transportation control. They advocate for the utilization of “The TON_IoT datasets” to augment the capabilities of intrusion detection systems. Their approach involves applying techniques such as Chi-square for feature selection and SMOTE (Synthetic Minority Over-sampling Technique) for class balancing, resulting in favorable outcomes with some of the classification algorithms employed.

Mohammad Shahin et al. [17] introduce a slightly different approach from previous works by employing fully convolutional neural networks for IoT device protection in industrial and smart manufacturing environments. They achieve promising results across various datasets, including "The TON_IoT datasets". Abdallah R. Gad et al. [9] utilize "The TON_IoT datasets" to perform multiclass classification. Abdallah employed data on memory usage and the status of various running processes, on both Linux and Windows 7 and Windows 10 systems. The results obtained in their research reached accuracy rates of 100% in some cases. Rania Elsayed et al. [7] address the issue of IoT device security, aiming to integrate security tools into existing smart cities. By integrating powerful neural networks, they have achieved results exceeding 98%. Abdulkarim Bashaiwth et al. [5] utilized Long Short-Term Memory (LSTM) networks to classify DDoS attacks, employing various interpretation methods for model predictions and achieving accuracy rates above 97%.

Rania Elsayed et al. [8] published a study continuing the exploration of machine learning-based systems to protect IoT device networks and Software-Defined Networks (SDN). They emphasized the low accuracy and scalability issues of previously employed techniques. In this work, they used an enhanced LSTM network capable of distinguishing between malicious and benign traffic, identifying the attack category, and predicting specific attacks with accuracies above 96%. The work of Ramkumar Devendiran et al. [6] proposes the use of neural networks with a stochastic or chaotic optimization strategy. They employ the *Chaotic Honey Badger* algorithm for optimal feature selection and use the Dugat-LSTM (Gated Attention Dual Long Short-Term Memory) algorithm for classification, achieving an accuracy rate close to 99%.

3 Problem Description

The primary aim of this study is to investigate and assess the feasibility of implementing systems that utilize machine learning techniques to detect cyber attacks at early stages in IoT networks.

In this paper, we focused on detecting attacks using supervised learning techniques applied to the TON_IoT dataset. The architecture used for data acquisition is comprehensively detailed in the following reference [16].

To facilitate the detection of cyber attacks on IoT devices, this study employed the *TON_IoT* dataset[14]. This dataset comprises heterogeneous data sources collected from various IoT and IIoT sensor telemetry systems. The data was gathered from Windows 7 and 10 operating systems, as well as Ubuntu 14 and 18 LTS, alongside network traffic datasets. Data collection was conducted over a realistic, large-scale network designed at the Cyber Range and IoT Labs, School of Engineering and Information Technology (SEIT), UNSW Canberra at the Australian Defence Force Academy (ADFA).

For this purpose, a new Industry 4.0 testbed network, incorporating both IoT and IIoT networks, was created. The testbed was deployed using multiple virtual machines and hosts running Windows, Linux, and Kali operating systems to

manage the interconnection between the three layers of IoT, Cloud, and Edge/-Fog systems. Various attack techniques, such as DoS, DDoS, and ransomware, were employed to collect attack data.

Once the dataset for model training was determined, data preprocessing techniques were applied. The goal was to retain the maximum number of features, removing only those with very low variance. Following the feature selection methodology presented by Abdallah et al. [9], IP addresses, timestamps, and port numbers were excluded from both source and destination fields to prevent overfitting issues. Non-numeric attributes, such as protocol, service, and connection status fields, were encoded using the *one-hot encoding* technique, converting the resulting boolean values to 1 and 0.

3.1 Explanation of Techniques and Research Motivation

Supervised learning was used, applying ensemble and boosting techniques with decision trees as “weak” classifiers due to their low computational cost for making predictions. It was possible to classify 1,000 data records from the validation set with the models generated, achieving prediction times of around 90 milliseconds using an Intel i7-13700K CPU and an Nvidia GeForce RTX 3080Ti GPU. The prediction accuracy rates obtained were very similar to those from the validation set, and the predictions were made within acceptable deadlines to consider its deployment in real-time scenarios.

Given the nature of the problem, an important aspect to consider is the prediction capability of the models. This is crucial because IoT networks typically have a large and growing number of connected devices. Therefore, it is essential that the models can rapidly predict the class of different network traffic captures to detect attacks at the earliest possible stage and minimize the resulting damage. This allows for the correct functioning of the devices to be restored promptly by applying appropriate countermeasures.

In related works, these same techniques are applied with very similar results, but the predictive capacity of the models is often not discussed or considered. This aspect is crucial for addressing this problem as it minimizes the hardware requirements and energy consumption needed to classify all the traffic of an IoT network. Additionally, it increases the probability of detecting attacks in their earliest stages. Due to the nature of these devices and networks, it is not considered viable to run this type of intrusion detection system directly on the IoT devices themselves.

3.2 Software Description

This subsection presents the software tools used for detecting attacks on IoT devices through supervised learning techniques.

Two datasets were obtained from the original dataset: an optimistic dataset and a dataset with correctly normalized values, which were used to perform different experiments. Both datasets were balanced by undersampling before any further processing operation were performed on them, randomly removing

records from the majority class, which in this case was the positive class that indicates that an attack had occurred. The same selection of attributes was performed on both sets, discarding the attributes of IP addresses, port numbers, and timestamps to avoid overfitting the models. Additionally, attributes related to DNS, which have null or unknown values in most of the data records, were discarded.

Data cleaning was performed by removing a few instances with anomalous values, such as detecting “0.0.0.0” in the ‘src_bytes’ attribute. The one-hot encoding technique was applied to encode the categorical features, and all values were converted to float type.

At this point, different processing was performed for both datasets. For the optimistic dataset, the values of all features were normalized and the dataset was split into 80% for training and 20% for validation, stratified by the ‘label’ attribute, which indicates whether the traffic is normal or malicious. On the other hand, for the correctly normalized dataset, the same dataset split was performed and the values of both sets were then normalized using the Scikit-Learn normalization function. Finally, the training sets contain more than 1 million data records and the validation sets contain more than 300,000 records.

Given the inherent complexity of these attacks, it is necessary to integrate various supervised learning techniques to achieve effective detection. There are different types of ensembles, such as those based on decision trees (e.g., random forest) and those based on neural networks. The individual classifiers that make up the model are referred to as weak classifiers, while the combined set of classifiers is known as a strong classifier. The main idea is to combine multiple weak classifiers to construct a strong classifier, thereby improving the model’s accuracy and robustness.

Random Forest (RF) ensemble and XGBoost (XGB) approaches were chosen. This model was tested with both the balanced and original datasets, yielding good results in both scenarios. By weighting the classes based on the number of instances, slightly better results were achieved.

The Random Forest model, as it was not highly sensitive to hyperparameter tuning and performed well with default parameters, was initially kept unchanged until a second model was implemented for comparison. After implementing and comparing multiple models, hyperparameters were adjusted through a trial-and-error process. This approach involved adjusting each parameter individually and observing its impact on the model’s performance. Additional modifications were made to the *Random Forest* model to incorporate a pipeline that includes feature selection using the SelectKBest function with chi-squared score function. This pipeline was combined with a grid search to determine the best parameters within a predefined set, including the optimal number of features to select from the dataset.

The *Grid Search* process involves implicit cross-validation, evaluating each parameter combination to find the most suitable configuration. However, due to the computational complexity involved, the execution time was significantly long, exceeding 24 h even with a moderate parameter grid and using 5-fold cross-

validation. To address this resource limitation, a randomized search technique was chosen. This method randomly explores parameter combinations within a specified grid, using a defined number of iterations. The best parameters obtained from this search were finally used to train and optimize the model. This strategy allowed for a more efficient and effective adjustment of the *Random Forest* model, maximizing its predictive capacity without incurring prolonged computational costs associated with exhaustive parameter searches.

Tables 1 and 2 display the respective hyperparameters for the XGB and RF models.

Table 1. Parameters used in the XGB models

Model	colsample bytree	learning rate	max depth	eval metric
XGB-MOPT	0.6	0.025	20	log_loss
XGB-RSOPT	0.5	0.255	30	log_loss
XGB-RS	0.7	0.1325	10	error
XGB-LOGL	0.6	0.025	20	log_loss
XGB-ERR	0.6	0.025	20	error

Table 2. Parameters used in the RF models

Model	num estimators	max depth	criterion	num atts
RF-MOPT	241	49	gini	47
RF-RSOPT	285	40	log_loss	35
RF-RS	271	10	gini	45
RF-GINI	250	30	gini	47
RF-ENTR	250	30	entropy	47
RF-LOGL	250	30	log_loss	47

4 Results and Comparison

This section summarizes the results obtained from the implemented models using different hyperparameter configurations and slightly different processed datasets.

The results are presented for different datasets using two implemented models (*Random Forest Classifier* and *XGB Classifier*), both with manual parameter tuning and randomized parameter search. The metrics of precision, recall, and F1-score are reported relative to the positive class (attacks). The booster used with XGB is *gbtree*, and the objective is set to *binary* for binary classification.

Table 3 presents the results obtained with each model using various hyperparameter configurations and evaluation or loss functions. The results are shown for the *Random Forest* classifier (RF) with the optimistic dataset using both manual tuning (RF-MOPT) and RandomizedSearchCV from Scikit-Learn (RF-RSOPT). Similarly, for the XGB model, the results are shown for manual tuning with the optimistic dataset (XGB-MOPT) and RandomizedSearchCV (XGB-RSOPT).

Subsequently, the performance results are provided for models trained with the correctly processed dataset. For *Random Forest*, the results are presented for RandomizedSearchCV tuning (RF-RS) and manual tuning with different impurity measurement criteria: Gini index (RF-GINI), entropy (RF-ENTR), and log-loss (RF-LOGL). For the XGB model, the results are shown for tuning with RandomizedSearchCV (XGB-RS) and manual tuning using log-loss (XGB-LOGL) or error rate (XGB-ERR) as evaluation criteria.

Table 3. Results of the different adjustments to the models

Model	Accuracy	Precision	Recall	F1-score
RF-MOPT	0.9851	0.9943	0.9758	0.9850
RF-RSOPT	0.9852	0.9944	0.9759	0.9850
XGB-MOPT	0.9855	0.9950	0.9759	0.9854
XGB-RSOPT	0.9847	0.9941	0.9752	0.9846
RF-RS	0.9378	0.9325	0.9439	0.9382
RF-GINI	0.9573	0.9690	0.9448	0.9568
RF-ENTR	0.9568	0.9684	0.9445	0.9563
RF-LOGL	0.9568	0.9684	0.9445	0.9563
XGB-RS	0.9861	0.9939	0.9782	0.9860
XGB-LOGL	0.9866	0.9943	0.9787	0.9865
XGB-ERR	0.9865	0.9942	0.9786	0.9863

Figure 1 shows confusion matrices for the RF-GINI (Fig. 1a), XGB-RS (Fig. 1b), XGB-LOGL (Fig. 1c) and RF-RS (Fig. 1d) models, respectively.

- | | |
|---|---|
| (a) Confusion matrix of the RF-GINI model. | (b) Confusion matrix of the XGB-RS model. |
| (c) Confusion matrix of the XGB-LOGL model. | (d) Confusion matrix of the RF-RS model. |

Fig. 1. Confusion matrices for each of the models implemented.

The *Random Forest* models exhibit highly promising metrics when trained on the ‘optimistic’ dataset, with an approximate 3% difference in accuracy. In

contrast, the XGB models show the opposite effect. Performance metrics are superior when the dataset is processed correctly, underscoring the importance of proper data preprocessing. Despite the observed performance loss in RF models, they demonstrate reduced generalization capability when confronted with data different from that used in training and testing, despite superior metrics during these phases. Among RF models trained with correctly processed data, a slight performance enhancement is observed when using the Gini index rather than entropy or logarithmic loss. As for XGB models, the difference is minimal, with the best results obtained using error rate as the evaluation metric.

In the paper by Abdullah Alsaedi et al. [3], the authors achieved accuracies ranging from 61% to 88% using a combination of datasets reflecting the status of various devices considered in the creation of the TON_IoT datasets. Missing values were addressed by completing them with the median of each dataset column.

In the studies by Abdalla R. Gad et al. [10, 12], similar results were obtained. By using the balanced dataset created with the SMOTE technique and performing feature selection, they achieved accuracies of 97–98% with the RF model and 98–99% with the XGB model.

In this work, slightly lower accuracies were obtained with the RF model and similar accuracies with the XGB model. Unlike the previous studies, oversampling was avoided to balance the classes to prevent overfitting the models with inferred data, despite the challenge of omitting crucial data records representing the class.

5 Conclusions and Future Work

For future research, enhancing model accuracy through advanced data preprocessing techniques and fine-tuning of hyperparameters will be prioritized. This may involve employing more sophisticated feature selection methods or exploring more intricate models like convolutional neural networks applied to network traffic data. Additionally, we propose implementing the model in real time to enable a swift and efficient intrusion detection. Achieving this goal will require optimizing model performance and employing real-time data processing techniques, ensuring the system operates effectively in dynamic and evolving IoT network environments. These initiatives aim not only to bolster the system's predictive capabilities but also to enhance its adaptability and operational efficiency in increasingly complex cybersecurity landscapes, with practical applications across a wide variety of sectors.

Some examples of relevant sectors, where attacks to the IoT infrastructure are paramount, can be highlighted. For example, in smart cities, where attacks targeting smart traffic infrastructure can happen, detecting potential attacks on smart traffic lights systems can become life-saving. In smart homes, preventing the hijacking of IoT devices, detecting distributed denial-of-service (DDoS) attacks or unauthorized data access is mandatory to ensure a safe and sound home environment for families. In healthcare, thwarting attacks on connected

medical devices, attempting to gain access and control over devices like infusion pumps or patient monitors to alter treatments or collect sensitive information is once again a life-threatening issue. Lastly, in industry, detecting potential attacks on industrial control systems (ICS) to prevent infiltration aimed at manipulating production processes or causing physical damage to facilities, which can cost lots of money and even lives too.

Acknowledgments. This publication is part of the AI4SECIoT project (“Artificial Intelligence for Securing IoT Devices”), funded by the National Cybersecurity Institute (INCIBE) - Spanish Ministry of Economic Affairs and Digital Transformation, and the European Union (NextGenerationEU*/PRTR).

References

1. Al-Sarawi, S., Anbar, M., Abdullah, R., Al Hawari, A.B.: Internet of Things market analysis forecasts, 2020–2030. In: 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), pp. 449–453. IEEE (2020). <https://doi.org/10.1109/WorldS450073.2020.9210375>
2. Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., Anwar, A.: TON_IoT telemetry dataset: a new generation dataset of IoT and IIot for data-driven intrusion detection systems. *IEEE Access* **8**, 165130–165150 (2020). <https://doi.org/10.1109/ACCESS.2020.3022862>
3. Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., Anwar, A.: Ton_iot telemetry dataset: a new generation dataset of iot and iiot for data-driven intrusion detection systems. *Ieee Access* **8**, 165130–165150 (2020)
4. Alsheikh, M., Konieczny, L., Prater, M., Smith, G., Uludag, S.: The state of IoT security: unequivocal appeal to cybercriminals, onerous to defenders. *IEEE Consumer Electr. Mag.* **11**(3), 59–68 (2021). <https://doi.org/10.1109/MCE.2021.3079635>
5. Bashaiwth, A., Binsalleeh, H., AsSadhan, B.: An explanation of the LSTM model used for DDoS attacks classification. *Appl. Sci.* **13**(15) (2023). <https://doi.org/10.3390/app13158820>, <https://www.mdpi.com/2076-3417/13/15/8820>
6. Devendiran, R., Turukmane, A.V.: Dugat-LSTM: deep learning based network intrusion detection system using chaotic optimization strategy. *Expert Syst. Appl.* **245**, 123027 (2024)
7. Elsayed, R., Hamada, R., Hammoudeh, M., Abdalla, M., Elsaied, S.A.: A hierarchical deep learning-based intrusion detection architecture for clustered Internet of Things. *J. Sens. Actuator Netw.* (2022)
8. Elsayed, R.A., Hamada, R.A., Abdalla, M.I., Elsaied, S.A.: Securing IoT and SDN systems using deep-learning based automatic intrusion detection. *Ain Shams Eng. J.* **14**(10), 102211 (2023)
9. Gad, A.R., Haggag, M., Nashat, A.A., Barakat, T.M.: A distributed intrusion detection system using machine learning for IoT based on ToN-IoT dataset. *Inter. J. Adv. Comput. Sci. Appl.* **13**(6) (2022). <https://doi.org/10.14569/IJACSA.2022.0130667>
10. Gad, A.R., Haggag, M., Nashat, A.A., Barakat, T.M.: A distributed intrusion detection system using machine learning for iot based on ton-iot dataset. *Inter. J. Adv. Comput. Sci. Appl.* **13**(6) (2022)

11. Gad, A.R., Nashat, A.A., Barkat, T.M.: Intrusion detection system using machine learning for vehicular AD HOC networks based on ToN-IoT dataset. *IEEE Access* **9**, 142206–142217 (2021). <https://doi.org/10.1109/ACCESS.2021.3120626>
12. Gad, A.R., Nashat, A.A., Barkat, T.M.: Intrusion detection system using machine learning for vehicular ad hoc networks based on ton-iot dataset. *IEEE Access* **9**, 142206–142217 (2021)
13. Khan, M., Ghafoor, L.: Adversarial machine learning in the context of network security: challenges and solutions. *J. Comput. Intell. Robot.* **4**(1), 51-63 (2024). <https://thesciencebrigade.com/jcir/article/view/118>
14. Moustafa, N.: ToN_IoT datasets (2019). <https://doi.org/10.21227/fesz-dm97>
15. Moustafa, N.: A new distributed architecture for evaluating AI-based security systems at the edge: network TON_IoT datasets. *Sustainable Cities Soc.* **72**, 102994 (2021). <https://doi.org/10.1016/j.scs.2021.102994>, <https://www.sciencedirect.com/science/article/pii/S2210670721002808>
16. Moustafa, N.: A new distributed architecture for evaluating ai-based security systems at the edge: Network ton_iot datasets. *Sustain. Urban Areas* **72**, 102994 (2021)
17. Shahin, M., Chen, F.F., Bouzary, H., Hosseinzadeh, A., Rashidifar, R.: A novel fully convolutional neural network approach for detection and classification of attacks on industrial IoT devices in smart manufacturing systems. *Inter. J. Adv. Manufact. Technol.* **123**(5), 2017–2029 (2022). <https://doi.org/10.1007/s00170-022-10259-3>



Understanding Malware Dynamics in IoT Networks: Dataset Construction Using Mathematical Epidemiology and Complex Networks

Leticia Sainz-Villegas¹, Roberto Casado-Vara¹✉, Nuño Basurto², Carlos Cambra², Daniel Urda², and Alvaro Herrero²

¹ Grupo de Inteligencia Computacional Aplicada (GICAP), Departamento de Matemáticas y Computación, Escuela Politécnica Superior, Universidad de Burgos, Av. Cantabria s/n, 09006 Burgos, Spain
✉ {lsainz,rccasado}@ubu.es

² Grupo de Inteligencia Computacional Aplicada (GICAP), Departamento de Digitalización, Escuela Politécnica Superior, Universidad de Burgos, Av. Cantabria s/n, 09006 Burgos, Spain
{nbasurto,ccbbaseca,durda,ahcosio}@ubu.es

Abstract. The advent of Internet of Things (IoT) devices has given rise to concerns regarding the dissemination of malware. It is of paramount importance to comprehend the behavior of malware in IoT networks in order to devise effective defense strategies. Our study introduces a methodology for constructing datasets that capture the dynamics of malware in IoT networks, combining mathematical epidemiology and complex network theory. The resulting datasets serve as a valuable resource for both researchers and practitioners, facilitating the analysis of malware propagation mechanisms by applying Artificial Intelligence methods, among others. This research contributes to a deeper understanding of malware dynamics in IoT networks, assisting in the design of cybersecurity measures to be applied in this dynamic field.

Keywords: Mathematical epidemiology · graph theory · malware propagation · data science · IoT network

1 Introduction

The Internet of Things (IoT) is considered as a transformative shift in connectivity, enabling the interconnection of billions of intelligent devices across diverse domains [7]. This connectivity enhances our ability to monitor and manage real-world phenomena, thereby enabling intelligent decision-making [8, 14]. However, the IoT possesses a vast potential, which is accompanied by significant security risks. These risks are primarily due to vulnerabilities in device software and limited security measures [9]. Malicious software, including self-replicating malicious

code, represents a significant threat to IoT systems, facilitated by the proliferation of lightweight devices with constrained resources. Malicious software can be disseminated through infected devices and unmanned aerial vehicles, forming botnets that can be exploited for malicious activities such as distributed denial-of-service attacks [10]. The complexity of IoT networks, characterized by the use of diverse communication technologies, serves to further exacerbate the challenges associated with security [11]. While these networks offer seamless communication, they also present opportunities for malware propagation across heterogeneous environments. Despite the critical need to address malware threats in IoT, there is a lack of datasets on propagation, particularly in IoT networks. It is of paramount importance to gain an understanding of the critical threshold for malware propagation if effective defense strategies are to be developed, including the patching of devices and the determination of patching probabilities. The development of optimal patching strategies represents a crucial step in the effective and economic mitigation of malware outbreaks in heterogeneous IoT networks [12, 15].

Given the impracticality of conducting experiments on malware propagation and device patching in large-scale IoT systems and the lack of sufficient high-quality data, mathematical modeling emerges as a promising approach to uncover the mechanisms of malware propagation and devise efficient patching strategies across diverse IoT networks. Classical epidemic models are frequently employed for the study of malware propagation [6, 13]. In these models, nodes are categorized into various compartments based on their spreading states, including susceptible (S), infected (I), and recovered (R) compartments. In order to study the problem of malware propagation in IoT networks and to identify potential mitigation measures, this research has defined a methodology for the construction of datasets on malware propagation in IoT networks. Such datasets are aimed at providing subsequent models (Artificial Intelligence mainly) with the required inputs for appropriate learning. The principal contributions of this paper are outlined below:

- A methodology is proposed for the simulation of IoT networks with the Python networkX library and the creation of contact topology malware propagation models for complex networks.
- A dataset has been created as a case study, with three types of associated visualizations for researchers to have the most information about malware propagation processes in IoT networks.

The rest of this paper is organized as follows. Section 2 describes the tools of our proposal: networkX to build graphs and mathematical epidemiology to model malware propagation. Section 3 introduces the generated dataset and the presented visualizations. Finally, Sect. 4 draws the conclusion of this paper.

2 Methods

This section outlines the two principal tools that are essential for the creation of the datasets under the frame of the present research. Firstly, a brief tutorial is

provided on the Python NetworkX library, which will be employed to simulate the IoT network. Secondly, a brief introduction to epidemiological models is provided, along with a description of the individual malware propagation model with which the dataset will be created. The individual model has been selected for its prevalence and adaptability. Modifications to the code can be made to create a dataset aligned with the desired investigation.

2.1 NetworkX: Modelling IoT Networks with Graphs

NetworkX is a free open-source Python package for modeling, analyzing, and visualizing networks. It provides classes to represent several types of networks and implementations of many of the algorithms used in network science [1–3]. Excellent documentation can be found on the project website <https://networkx.github.io/>. At the time of writing, NetworkX is in version 3.3 but many things are the same as version 2.x. However, documentation and papers related to older versions may no longer be accurate. In order to start working with NetworkX, it is necessary to have Python installed on the operating system, along with the NetworkX package. In some of the subsequent examples, we illustrate how NetworkX interacts with other optional Python packages, including NumPy, SciPy, and Matplotlib. We recommend considering the installation of these packages as well, as NetworkX will automatically utilize them if they are accessible. To begin, import NetworkX into your Python environment using the abbreviation “nx” to streamline your coding process.

```
>>> import networkx as nx
```

The Graph class represents the fundamental container for storing network data. The incorporation of nodes into the graph is achieved through the following method:

```
>>> G=nx.Graph()
>>> G.add_node('device1')
>>> G.add_node('device2')
```

Edges or links between nodes are represented as tuples of nodes. They can be added as follows

```
>>> G.add_edge('device1','device2')
```

If the nodes do not already exist they are automatically added to the graph. NetworkX offers a comprehensive range of functionalities for computing network statistics and metrics, including, but not limited to, diameter, degree distribution, the number of connected components, clustering coefficients, and betweenness centrality. Moreover, it provides generators for various classic graphs and random graph models. These resources are invaluable for modeling and analyzing network data, as well as for evaluating novel algorithms or network metrics.

It is possible to visualize the graph in order to gain a better understanding of its structure. This can be achieved by utilizing the Matplotlib library in conjunction with the NetworkX ‘draw’ function.

```
>>> import matplotlib.pyplot as plt
>>> nx.draw(G, with_labels=True)
>>> plt.show()
```

This will result in the generation of a visual representation of the network, with labels on the nodes, which will facilitate the comprehension of the topology of the network.

2.2 Mathematical Epidemiology and Its Application to Model the Spread of Malware

The SIR model, developed by Kermack and McKendrick in 1927 [4], represents a fundamental framework for understanding the spread of communicable diseases. It categorizes individuals into three compartments: susceptible S , infectious I , and recovered R . The dynamics of these groups are governed by differential equations, where $S(t)$ represents the susceptible population, $I(t)$ the infectious population, and $R(t)$ the recovered population. In the conventional SIR model, it is assumed that the total population N remains constant, with $N = S(t) + I(t) + R(t)$. The transition of individuals between these groups is determined by parameters such as the transmission rate β and the recovery rate γ . In the context of malware propagation in IoT networks, factors such as network connectivity restrictions and security measures can influence transmission rates, thereby underscoring the necessity for ongoing research in this area [5]. Malware propagation through an IoT network can be modeled using a cellular automaton (CA) on a graph $G = (V, E)$, where V is the set of nodes (vertices) and E is the set of edges connecting the nodes. Each node v_i in V represents an entity, such as a computer or an IoT device, and the edges e_{ij} in E represent the relationships between entities. The set S represents the allowed states for a node, such as susceptible (S), infected (I), or recovered (R). The state of each node v_i evolves according to a transition function $f : S^{|N(v_i)|} \rightarrow S$ that depends on the states of the neighboring nodes $N(v_i)$. This iterative process is applied to all nodes at each time step. Formally, for a given node v_i at time $t+1$ the transition rule is as follows:

$$s_i(t+1) = f(s_{j_1}(t), s_{j_2}(t), \dots, s_{j_k}(t)) \quad (1)$$

where j_1, j_2, \dots, j_k denotes the neighbor of v_i . This process is repeated across all nodes at each time step.

To simulate the spread of malware, the transition function f can include parameters that define infection probabilities, recovery rates, and other relevant factors. This allows the cellular automaton implemented on the graph to simulate the spread of malware accurately. Martin del Rey et al. presented transition rules in their paper, which we have implemented to simulate the propagation of malware [6]. Several assumptions need to be considered before outlining the transmission functions:

- A susceptible device can become infected depending on the infection rate, β_i , of each IoT node, where $0 \leq \beta_i \leq 1$. Furthermore, a node in a susceptible

state can only change to an infected state if one of the surrounding nodes is infected.

- An infected device can be restored or remain infected, depending on the recovery rate, γ_i , of each IoT node, where $0 \leq \gamma_i \leq 1$. In this case, nodes which are in an infected or recovered state depend only on themselves, not on the state of nodes in their immediate environment.
- This model considers permanent immunity, preventing the malware from infecting recovered devices.

Consequently:

If $s_i^t = (S)$:

$$s_i^{t+1} = \begin{cases} I, & \text{with probability } \beta_i \\ S, & \text{with probability } 1 - \beta_i \end{cases} \quad (2)$$

If $s_i^t = (I)$:

$$s_i^{t+1} = \begin{cases} R, & \text{with probability } \gamma_i \\ I, & \text{with probability } 1 - \gamma_i \end{cases} \quad (3)$$

If $s_i^t = R$, then $s_i^{t+1} = R$

3 Simulation

This section outlines the simulation setup employed to conduct the simulations, the parameters utilized in the design of the individual model simulating the propagation of malware, as well as the parameters employed for the design of the graph simulating the IoT network. Furthermore, the output (the dataset itself) is ready for exporting it to different data formats, and three visualizations are provided to assist in the comprehension of the malware's propagation.

3.1 Simulation Setup

The simulation to create the dataset was conducted in Python and NetworkX to code the complex networks on a desktop computer (CPU: Intel (R) Core (TM) i7-8700 CPU @ 3.20 GHz, Memory: 16 GB, OS: 64-bit Microsoft Windows 10). The following parameters have been utilized in all simulations. The number of nodes in the graph representing the IoT network is 30, the probability of reconnections when creating a new node is 0.5, and the number of connections when adding a new node is 2. The IoT network was simulated as a random graph utilizing NetworkX's powerlaw_cluster_graph function. This approach facilitates the generation of networks featuring a heavy-tailed degree distribution, akin to what's observed in genuine networks like those in IoT systems [18]. Consequently, certain nodes exhibit numerous connections, contrasting sharply with others

possessing only some of them. Such a degree of distribution mirrors patterns found in various real-world networks, such as those in social and communication contexts. Conversely, the malware infection rate is 0.6, and the recovery rate is 0.2 (these parameters have been selected from among the most common in numerical examples in the literature [16, 17]). The malware has been spreading for 10 steps. The number of time steps was limited to 15 because, being a SIR model that does not contemplate reinfection, the model tends to have all nodes in a ‘recovered’ state with the passage of a few time steps. A total of 50 simulations of malware propagation in the IoT network were conducted over a period of 10 steps, with the results averaged for each IoT network node in each step. This approach serves to mitigate the randomness introduced by the individual malware propagation model, with the infection and recovery probabilities for each IoT network node at each time step.

3.2 Dataset Results and Visualization

This section details the outcomes of the average of all simulations to create the dataset. The dataset, as shown in Table 1, is initially created in a format that can be easily exported to different file-formats. It is coded in letters according to the selected malware model. In this case, an SIR model is being used, which results in the state of the devices being encoded with the letters ‘S’, ‘I’, and, ‘R’. This allows researchers who want to exploit the dataset to make the necessary changes for their study.

Table 1. Sample data from the generated dataset. In the header of the table, ‘TS’ means ‘time step’.

TS/Nodes	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	S	I	I	S	S	S	I	S	S	S	S	S	S	S	S	I	S	I	S	S	S	S	S	S	S	S	S	S		
1	I	I	I	S	I	S	I	S	S	S	S	S	S	S	S	I	I	I	I	S	S	I	I	S	I	I	S	S		
2	R	I	I	I	I	I	I	S	I	I	S	S	S	S	S	I	I	I	R	I	I	I	S	S	I	I	R	R	S	S
3	R	I	I	R	I	I	I	I	I	I	S	I	I	S	S	I	I	R	R	I	I	R	R	R	R	R	R	R	I	I
4	R	I	I	R	I	I	R	I	R	R	I	I	S	S	I	I	R	R	I	I	R	R	R	R	R	R	R	R	I	I
5	R	I	I	R	I	I	R	R	R	R	R	I	I	R	R	I	I	R	R	R	R	R	R	R	R	R	R	R	I	I
6	R	R	I	R	I	R	R	R	R	R	R	I	I	R	R	I	I	R	R	R	R	R	R	R	R	R	R	R	I	I
7	R	R	I	R	R	R	R	R	R	R	R	R	R	R	R	R	R	I	I	R	R	R	R	R	R	R	R	R	I	R
8	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	I	I	I	R	R	R	R	I	S
9	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	S

In order to facilitate the visual analysis of the dataset, three visualizations have been designed, each of them depicts the spread of malware from a different perspective. In all these visualizations, the color scheme is consistently applied

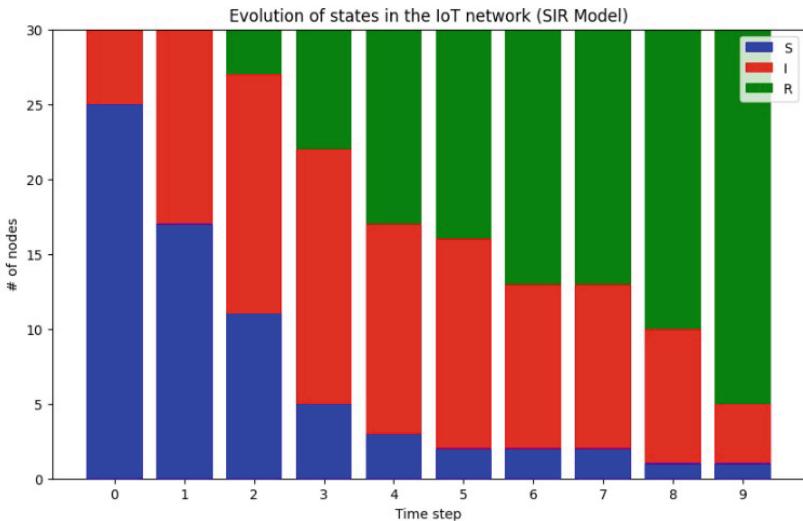


Fig. 1. Overview of the total number of susceptible, infected, and recovered nodes in each time step. Blue is for nodes in a susceptible state (S), red is for infected nodes (I), and green is for recovered nodes. (Color figure online)

throughout the entire network, using the same colors to indicate the following states: blue represents nodes in the susceptible state, red represents infected nodes, and green represents recovered nodes.

In mathematical epidemiology, where compartmental and deterministic propagation models are employed, the contact topology is not considered, and only the quantities present in each compartment are visualized. Figure 1 illustrates the number of IoT devices in each compartment from the initial step with the label ‘0’ to the conclusion of the simulation. As the SIR model employed does not permit reinfection, the simulation can be run multiple times and the results averaged to demonstrate that the majority of devices are in the recovered state, while the infected and susceptible devices represent a minority of all devices.

However, a more suitable visualisation when considering contact topology and therefore using individual propagation models is one in which the state of each individual IoT device (nodes in the graph) can be visualised for the duration of the simulation. Figure 2 shows such visualization, depicting the evolution of the propagation in each of the IoT devices.

For the third visualization, it is proposed to use the NetworkX library to create a succession of graphs, in which each of the nodes is assigned a color legend as indicated above. Furthermore, a layout is employed to ensure that the network remains in a fixed position throughout the simulation, thereby facilitating the identification of each node. In this manner, a panel has been constructed to illustrate the evolution of the propagation of each IoT network during the simulation. As a result, the propagation of the malware can be observed with the naked eye and in an intuitive way (Fig. 3).

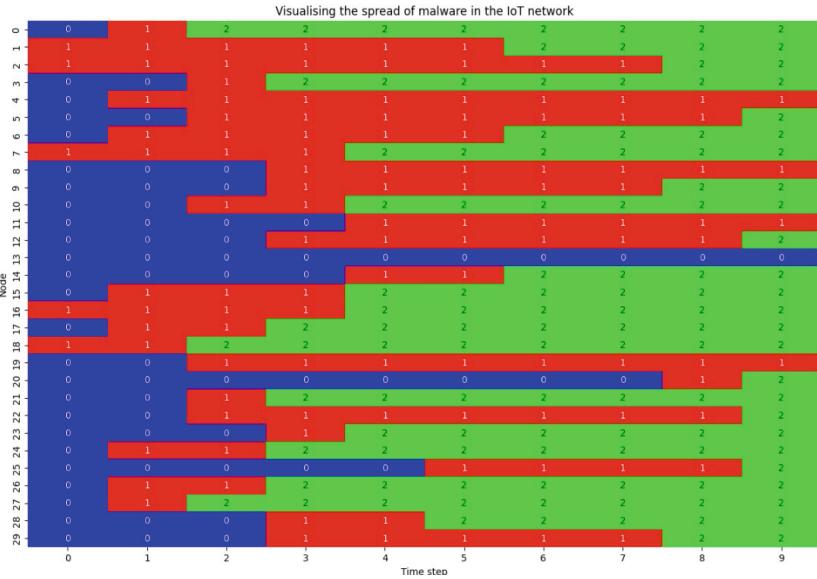


Fig. 2. Global overview of the malware propagation over the IoT network per device (node). Blue is for nodes in a susceptible state, red is for infected nodes, and green is for recovered nodes. (Color figure online)

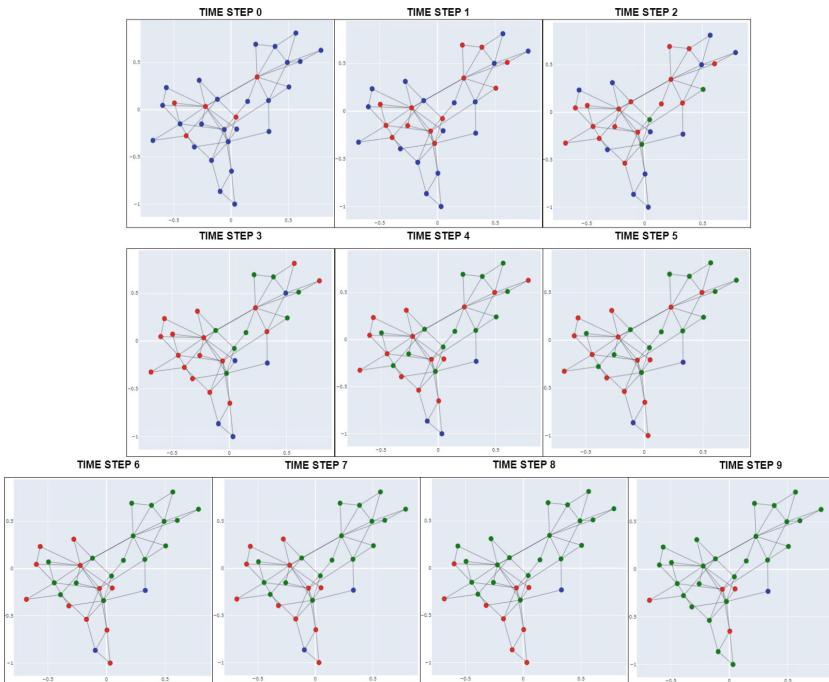


Fig. 3. General overview of the IoT network devices states during the malware propagation.

4 Conclusions and Future Work

A novel approach for the creation of open-access datasets aimed at investigating malware propagation is put forth. This methodology enables the customisation of IoT networks through the utilisation of complex network theory and the NetworkX library. Moreover, for the modelling of malware propagation, the use of a single model based on a SIR model is proposed. However, this model is easily customisable by changing a module of the code. As a consequence of the fact that the contact topology (i.e., the complex network) and the parameters of the epidemiological model depend on each iteration of the model, future work will investigate in depth how to minimise this fact in order to produce more reliable datasets.

Acknowledgements. This research is part of the AI4SECIoT Strategic Project (“Artificial Intelligence for Securing IoT Devices”), under the collaboration agreement signed between the National Cybersecurity Institute (INCIBE) and the University of Burgos. This initiative is carried out within the framework of the Recovery, Transformation and Resilience Plan funds, financed by the European Union (Next Generation).

References

1. Scellato, S.: NetworkX: Network analysis with Python (2013)
2. Hagberg, A., Swart, P., Schult, D.: Exploring network structure, dynamics, and function using NetworkX (No. LA-UR-08-05495; LA-UR-08-5495). Los Alamos National Lab. LANL, Los Alamos, NM (United States) (2008)
3. Platt, E.L.: Network science with Python and NetworkX quick start guide: explore and visualize network data effectively. Packt Publishing Ltd (2019)
4. Kermack, W.O., McKendrick, A.G.: A contribution to the mathematical theory of epidemics. Proc. R. Soc. London Ser. A Contain. Papers Math. Phys. Character **115**(772), 700–721 (1927)
5. Shafiq, M., Gu, Z., Cheikhrouhou, O., Alhakami, W., Hamam, H.: The rise of “internet of things”: review and open research issues related to detection and prevention of IoT-based security attacks. Wirel. Commun. Mob. Comput. **2022**, 1–12 (2022)
6. del Rey, A.M., Vara, R.C., González, S.R.: A computational propagation model for malware based on the SIR classic model. Neurocomputing **484**, 161–171 (2022)
7. Stoyanova, M., Nikoloudakis, Y., Panagiotakis, S., Pallis, E., Markakis, E.K.: A survey on the internet of things (IoT) forensics: challenges, approaches, and open issues. IEEE Commun. Surv. Tutor. **22**(2), 1191–1221 (2020)
8. Chettri, L., Bera, R.: A comprehensive survey on internet of things (IoT) toward 5G wireless systems. IEEE Internet Things J. **7**(1), 16–32 (2019)
9. Yaacoub, J.P.A., Noura, H.N., Salman, O., Chehab, A.: Ethical hacking for IoT: security issues, challenges, solutions and recommendations. Internet Things Cyber-Phys. Syst. **3**, 280–308 (2023)
10. Shirvani, M.H., Masdari, M.: A survey study on trust-based security in internet of things: challenges and issues. Internet Things **21**, 100640 (2023)
11. Qays, M.O., Ahmad, I., Abu-Siada, A., Hossain, M.L., Yasmin, F.: Key communication technologies, applications, protocols and future guides for IoT-assisted smart grid systems: a review. Energy Rep. **9**, 2440–2452 (2023)

12. Wang, X., Zhang, X., Wang, S., Xiao, J., Tao, X.: Modeling, critical threshold, and lowest-cost patching strategy of malware propagation in heterogeneous IoT networks. *IEEE Trans. Inf. Forensics Secur.* (2023)
13. Silva, D.H., Anteneodo, C., Ferreira, S.C.: Epidemic outbreaks with adaptive prevention on complex networks. *Commun. Nonlinear Sci. Numer. Simul.* **116**, 106877 (2023)
14. Casado-Vara, R., Canal-Alonso, A., Martin-del Rey, A., De la Prieta, F., Prieto, J.: Smart buildings IoT networks accuracy evolution prediction to improve their reliability using a lotka-volterra ecosystem model. *Sensors* **19**(21), 4642 (2019)
15. Casado-Vara, R., Severt, M., Díaz-Longueira, A., Rey, Á.M.D., Calvo-Rolle, J.L.: Dynamic malware mitigation strategies for IoT networks: a mathematical epidemiology approach. *Mathematics* **12**(2), 250 (2024)
16. Cangiotti, N., Capolli, M., Sensi, M., Sottile, S.: A survey on Lyapunov functions for epidemic compartmental models. *Bollettino dell'Unione Matematica Italiana* **17**(2), 241–257 (2024)
17. Naik, P.A., Eskandari, Z., Madzvamuse, A., Avazzadeh, Z., Zu, J.: Complex dynamics of a discrete-time seasonally forced SIR epidemic model. *Math. Methods Appl. Sci.* **46**(6), 7045–7059 (2023)
18. Pei, H., Yan, G., Huang, Y.: Impact of contact rate on epidemic spreading in complex networks. *Eur. Phys. J. B* **96**(4), 1–7 (2023). <https://doi.org/10.1140/epjb/s10051-023-00513-2>



Quality of Service Enhancement for IoT-Based Smart Office System Using Ad Hoc On-Demand Distance Vector-Smart Control Ration Control Algorithm (AODV-SRCA)

Lawrence Nforh CheSuh¹, Ramón Ángel Fernández-Díaz¹ , Jose Aveleria-Mata¹ , Martín Bayón-Gutiérrez¹ , Sergio Rubio Martín¹ , and Héctor Alaiz-Moretón²

¹ Department of Electric, Systems and Automatics Engineering, Universidad de León, Campus de Vegazana, S/N, 24071 León, Spain

{lnforc00, raferd, jose.aveleira, martin.bayon, srubm}@unileon.es

² RIASC, Research Institute of Applied Sciences in Cybersecurity, Universidad de León. MIC, Campus de Vegazana, S/N, 24071 León, Spain
hector.moreton@unileon.es

Abstract. The performance of an application in IoT settings heavily relies on the quality of service (QoS) parameters, which play a crucial role. Due to their diverse nature, attaining optimal QoS parameters presents a significant challenge. This study presents an innovative approach to enhance QoS parameters in IoT by leveraging machine learning (ML) and Blockchain methodologies. A simulation of the IoT network integrated with Blockchain technology is conducted using an NS2 simulator to improve QoS. Various QoS parameters, including delay, throughput, packet delivery ratio, and packet drop, are thoroughly examined. Subsequently, the QoS data is analysed using different ML algorithms such as Naive Bayes (NB), Decision Tree (DT), and Ensemble learning techniques. The findings indicate that the Ensemble classifier achieves the highest classification accuracy of 83.74% compared to NB and DT classifiers.

Keywords: Blockchain · Classification accuracy · Decision Tree · Ensemble Learning First Section · Internet of Things (IoT) · Machine Learning · Naive Bayes · Quality of Service (QoS)

1 Introduction

Digital network systems have matured enormously since the advent of the Internet of Things (IoT) network, which revolutionized data transmission by making it more intelligent and autonomous [1]. With the aid of this technology, diverse machines and devices can be integrated into a single network, generating a significant quantity of data that could provide insightful information in a variety of fields. IoT has an impact on the development of smart applications, including intelligent transportation systems [4], smart cities

[3] intelligent transportation systems [4], and smart healthcare applications [5], among others. One notable application of IoT is the smart office concept, which encompasses features aimed at automating and enhancing the work environment's functionality [6]. In a smart office network, a multitude of sensors, actuators, and advanced devices create an intelligent setting by regulating factors like temperature, humidity, lighting intensity, and noise levels to cater to the needs of office occupants [7]. These sensors and actuators use IoT principles to transmit data through network protocols. The smart office concept is tailored to enable employees to efficiently carry out their daily tasks within a predetermined time frame. Smart office solutions interface with various network systems, databases, and project management systems, overseeing office operations from end to end [8]. The primary objective of most smart offices is to enhance employee comfort while achieving business goals such as increased performance, energy efficiency, and improved work quality. Additionally, smart offices must meet predefined Quality of Service (QoS) metrics to optimize overall performance. However, most current studies have concentrated on various facets of IoT, with little emphasis placed on enhancing QoS within IoT applications. There are limited resources available that underscore the significance of QoS in applications like smart cities, smart healthcare systems, and smart offices. It is crucial to manage every facet of QoS metrics, like optimal resource allocation, scalability, security, and reliability, to be able to guarantee effective QoS management in Internet of Things applications. This paper proposes a novel system design which takes into consideration a variety of performance measures covered in the following sections to improve QoS in IoT-based smart offices. Several IoT devices that make up the recommended IoT architecture provide an extensive range of functions, like data monitoring, sensor analysis, and smart device interconnection. However, the paragraphs that follow are indented. Digital network systems have changed because of the Internet of Things (IoT), which has made data transmission more intelligent and autonomous.

This technology enables the integration of diverse machines and devices into a unified network, generating extensive data with the potential for valuable insights across different domains. IoT applications range from smart cities to intelligent transportation systems and healthcare solutions. One notable application is the smart office, which aims to automate and enhance workplace functionality by regulating factors like temperature, humidity, lighting, and noise levels. Smart office solutions utilize IoT principles, leveraging sensors and actuators to facilitate data transmission through network protocols. These solutions are designed to enable employees to efficiently carry out their tasks within a predetermined timeframe, interfacing with various network systems and databases to oversee office operations comprehensively. Furthermore, it is essential for smart offices to prioritize meeting predefined Quality of Service (QoS) standards to optimize overall performance. However, existing studies have predominantly focused on various aspects of IoT without giving adequate attention to enhancing QoS within IoT applications. Limited resources emphasize the importance of QoS in applications like smart cities, healthcare systems, and offices. To ensure effective QoS management in IoT applications, it's crucial to address all facets of QoS metrics, including reliability, scalability, security, and optimal resource allocation. This research proposes a novel system architecture to enhance QoS in IoT-based smart offices, considering various performance metrics detailed in subsequent sections. The proposed IoT architecture

comprises diverse IoT devices offering varied services such as data monitoring, sensor data analysis, and interconnection of various smart devices.

The rise of the Internet of Things (IoT) network represents a significant advancement in digital network systems, revolutionizing how data is transmitted, making it more intelligent and autonomous [1]. This technology enables the integration of diverse machines and devices into a unified network, fostering extensive data generation with the potential to offer valuable insights across various domains (Rathore et al., 2021) [2]. The impact of IoT extends to the development of smart applications, such as smart cities [3], intelligent transportation systems [4], and smart healthcare applications [5], among others. One notable application of IoT is the concept of the smart office, which encompasses features aimed at automating and enhancing the work environment's functionality [6]. In a smart office network, a multitude of sensors, actuators, and advanced devices create an intelligent setting by regulating factors like temperature, humidity, lighting intensity, and noise levels to cater to the needs of office occupants [7]. Employing IoT principles, these sensors and actuators facilitate data transmission through network protocols. The smart office concept is tailored to enable employees to efficiently carry out their daily tasks within a predetermined time frame. Smart office solutions interface with various network systems, databases, and project management systems, overseeing office operations from end to end [8]. The primary objective of most smart offices is to enhance employee comfort while achieving business goals such as increased performance, energy efficiency, and improved work quality. Additionally, it is crucial for smart offices to meet predefined Quality of Service (QoS) metrics to optimize overall performance. However, most of the current studies have concentrated on various facets of IoT, with little emphasis placed on enhancing QoS within IoT applications.

There aren't many sites that highlight how important QoS is for applications like smart offices, smart cities, and smart healthcare systems. All the QoS metrics—reliability, scalability, security, and optimal resource allocation—must be taken into consideration for effective QoS management in Internet of Things applications. Considering the different performance indicators covered in the following sections, this study proposes a new system architecture for enhancing QoS in IoT-based smart offices. The proposed Internet of Things architecture is comprised of numerous IoT devices that provide a range of services, such as data monitoring, sensor data analysis, and smart device interconnection. However, the subsequent paragraphs are indented. The paper is structured as follows: The Abstract summarizes the Research; the Introduction outlines the significance, and the Literature Review discusses previous studies. The Method section describes our algorithm and implementation, while the images give a figurative explanation and finally the references provide all cited sources.

2 Literature Review

The aim of many research projects has been to enhance the quality of service (QoS) in various IoT applications. This project focuses on integrating blockchain and machine learning to improve QoS in smart office environments. The authors in [9] introduced a ML-driven model aimed at predicting QoE in small cell networks, which is sensitive to QoS, content, and device variations. This method utilizes a Multilayer Perceptron

(MLP) neural network to forecast QoE particularly in medical diagnostic scenarios. Additionally, it incorporates an adaptive video streaming technique that adapts to device specifications to uphold and enhance diagnostic quality. The model is trained across diverse datasets encompassing various input parameters including QoS metrics, user attributes, and QoE indicators. The work presented in [10] is dedicated to enhancing QoS and QoE within communication networks, particularly in smart city applications. It delves into the influence of network data on automating the management of QoS and QoE performance, presenting a data-driven approach for this purpose. The proposed method integrates features of training performance prediction and utilizes acquired data to regulate both QoS and QoE performance in smart applications. An efficient approach is presented in [11] focused on improving QoS and QoE performance by analyzing the communication networks in smart applications such as smart cities. The study explores the impact of using network data to automate the controlling process of QoS and QoE performance by developing a data driven QoS and QoE approach. The proposed approach incorporates the characteristics of the training performance prediction and based on the obtained data, it manages the performance of both QoS and QoE for smart applications. A similar approach is discussed in [12] wherein a concise summary of IoT-driven smart applications like smart health, smart home, and smart offices is provided. It examined how IoT contributes to enhancing the QoS in these applications. Various performance evaluation metrics such as innovative solutions, enhanced data rates, expanded bandwidth, increased capacity, reduced latency, and improved throughput were scrutinized. Moreover, the role of artificial intelligence and machine learning in bolstering QoS performance within IoT-driven smart applications was also discussed. The work presented in [13] explored enhancing QoS parameters in wireless sensor networks through machine learning methods. It examined diverse facets of machine learning driven QoS, including research patterns, emerging protocols, and associated challenges. Various QoS aspects such as bandwidth, scalability, latency, and error rates were scrutinized. Additionally, the study outlined potential avenues for future research and highlighted the advantages of employing machine learning techniques to enhance QoS parameters. In addition to these parameters there is a need for enhancing the communication between multiple IoT devices which can be achieved by deploying various routing protocols. Several routing algorithms were examined by the authors in [14], including source location privacy-based ring loop routing (SLPRR), stochastic and diffusive routing algorithm (SDR), energy efficient secure route adjustment (ESRA), double level unequal clustering algorithm (DLUC), and sector-based random routing scheme (SRR). In addition to complying with other QoS metrics, these techniques have been scientifically demonstrated to be effective.

The majority of QoS-oriented research has primarily overlooked intrusion detection within smart office network systems. Given that IoT-based applications are highly susceptible to threats and attacks due to their extensive integration with diverse devices, unauthorized entities find it relatively easier to initiate intrusions into these systems. Such intrusions pose a significant risk of impairing the operational integrity of office environments. In this context this study intends to address this issue and strengthen the performance of IoT-based smart offices.

3 Methods Used for Enhancing QoS in Smart Offices

This study put into practice a blockchain-integrated, IoT-based intelligent system. The many IoT components, sensors, and connected devices make up the envisioned smart system. It is suggested to use a Blockchain-based authentication system to protect IoT data from unidentified IoT nodes. Since a number of IoT functions, including data gathering, data transfer, network activity monitoring, and storage, need high quality of service, these functions are carried out by delegated agents in the IoT network in order to satisfy the QoS requirements. In addition, various sensors are used in conjunction with a secure hash code authentication system based on blockchain technology to create an intelligent path inside the Internet of Things. Within the proposed IoT-based smart network, this strategy tackles problems with packet delivery ratio, packet loss ratio, throughput, delay time, and network congestion. Furthermore, several machine learning techniques for identifying QoS parameters, include ensemble learning model, decision tree, and Naive Bayes (NB).

3.1 Dataset Used

The QoS parameters are assessed using an ML dataset. Six characteristics and 413 records that were taken from the examination of ns2 trace files make up the dataset. Two class labels—QoS (packet transported successfully) and QoS (packet transfer failed)—make up the dataset. This data was used to train machine learning algorithms for categorization.

3.2 QoS Analysis Using AODV-SRCA Algorithm

This study implements a blockchain-based Internet of Things smart system using various IoT devices, sensors, and connecting devices. High QoS is required for certain IoT functions, such as data collection, transmission, storage, and network activity monitoring. To satisfy predetermined QoS requirements, these tasks must be carried out by delegated agents operating within the IoT network. These agents are in charge of keeping previously stored data as well as gathering data in real-time. The IoT network is made up of a collection of overlay zones, or subnetworks, formed by a set of agents inside the network. The QoS is determined by the scalability of parameters, and the IoT system agents also function as Blockchain nodes. Blockchain ensures the security of data stored and sent by these nodes by authenticating nodes inside each subnetwork. An intelligent pathway is established inside the IoT network by implementing a secure hash code authentication technique based on Blockchain in conjunction with several sensors. In the proposed IoT-based intelligent network, this method successfully handles issues such as packet delivery ratio, packet loss ratio, throughput, delay time, and network congestion. Figure 1 depicts the QoS analysis's flow.

The proposed approach employs an AODV protocol along with a Secure Smart Route Control Algorithm (AODV - sRCA) for identifying the optimal route for the nodes in the network. The steps involved in the execution of the AODV-sRCA algorithm are discussed as follows:

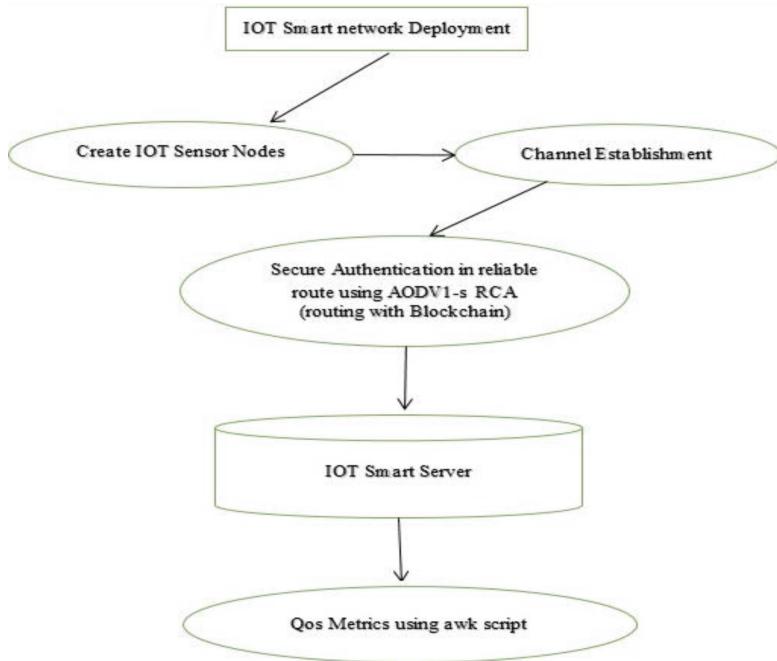


Fig. 1. Flowchart of the QoS analysis process

The AODV Protocol is initialized with different elements such as routing table, sequence number, and other parameters.

A node in the network is selected as an active node. A node broadcasts a path Request (RREQ) packet to a target node in order to find the best path for the current node to transfer data to. The addresses of the source and destination nodes as well as a unique identifier are contained in the RREQ. After receiving the RREQ, every intermediate node checks to see if it has a route that leads to the destination node. The RREQ is replayed in order to discover a valid route if it isn't. A legitimate route intermediate node or the destination node sends back an RREP packet after receiving the RREQ. The addresses of the source, destination, and intermediate nodes are included in the RREP. Furthermore, the intermediate nodes update their routing tables accordingly. For route maintenance, the nodes are checked periodically, and the validity of the routes is ensured and mentioned in their routing tables. If a route becomes invalid, the node initiates a new route discovery process. A Secure Smart Route Control Algorithm with Blockchain IoT network is initialized among the participating IoT nodes. A genesis block is created and is distributed to all the nodes. A copy of the Blockchain ledger is maintained by each node here. For registering the route, when a node wants to communicate with another node, it registers its intention on the Blockchain network.

3.3 ML Models for QoS Classification

The most efficient machine learning classifiers, including ensemble learning, decision trees, and naive bayes, are chosen in this study to categorize QoS parameters.

Naive Bayes Classifier: The NB classifier is selected because of its high scalability and excellent learning capacity. These classifiers represent one of the most straightforward network models, founded on the assumption of independence. This feature makes it possible for the model to be trained without the need for previous data. The accuracy of the classification is further improved by combining these classifiers with kernel density estimation.

Decision Tree (DT) Classifier: Classification and regression tasks are both capable of being completed using the DT classifier. In a decision tree, every leaf node has a class label assigned to it, and the non-terminal nodes (internal and root nodes) have feature test restrictions to identify different features using different features. The learnt function is obtained using decision tree techniques, and decision tree methodology is used to approximate target functions in discrete systems. Numerous applications, especially in image classification, make heavy use of the inductive approach that decision tree techniques require. Data mining techniques like decision trees are very relevant to IoT systems because they can reliably identify.

Ensemble Learning Classifier: Ensemble learning involves amalgamating two or more machine learning classifiers. This approach aids in enhancing the performance of machine learning models by integrating multiple individual models. Ensemble models typically outperform single machine learning models in terms of predictive and classification accuracy. In the context of IoT, ensemble learning effectively addresses various challenges including statistical, computational, and representational issues.

Sample Heading (Third Level). Only two levels of headings should be numbered. Lower level headings remain unnumbered; they are formatted as run-in headings.

4 Methods Used for Enhancing QoS in Smart Offices

The experimental analysis's findings are given in this section. With the NS2 simulator, an IoT-based smart network is simulated. The IoT receives its input data from the NS2 trace file, which was created during NS2 simulation. Table 1 provides information about the simulation parameters and network architecture used in the experimental investigation.

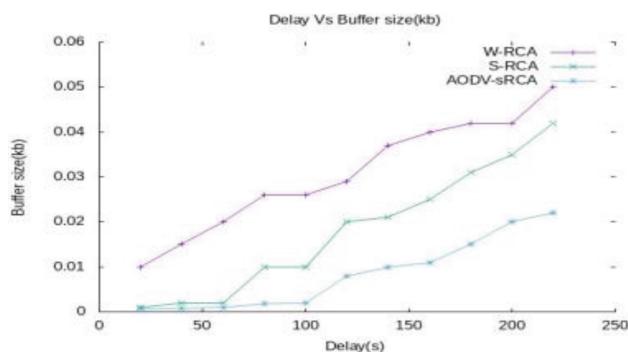
Results of the experimental analysis are set forth in this section. An IoT-based smart network is simulated using the NS2 simulator. The NS2 trace file, which was generated during NS2 simulation, provides the IoT with its input data. The network architecture and simulation parameters used in the experimental investigation are outlined in Table 1.

Every user is required to undergo authentication and verification using this scheme before accessing the data, ensuring data access only upon validation of user requests. Furthermore, all blocks undergo verification using hash codes. The simulation demonstrates an improvement in QoS as illustrated in below graphs (Figs. 2 and 3).

Correspondingly, the output of the ML classifiers are discussed in Table 2 (Fig. 4).

Table 1. Network Configuration and Simulation Parameters

S.No	Parameter	Value	Explanation
1	Network Area (m^2)	1821 * 500	Defines the physical space for node deployment
2	Number of Sensor Nodes	50	Determines the network's complexity and data generation
3	Packet Size (bytes)	1024	Affects network efficiency and latency
4	Propagation Model	Two-Ray Ground	Simulates signal propagation
5	Number of Channels	16	Provides bandwidth for communication
6	MAC Standard	IEEE 802.11	Defines wireless communication protocol
7	Antenna	Uni-directional	Directs signal transmission and reception
8	Scheduler	MAC Scheduler	Manages access to the shared wireless medium
9	Mobility Model	Random Waypoint	Simulates node movement patterns
10	Internet Protocol	UDP	Connectionless datagram protocol
11	IoT Protocol	MQTT	Lightweight messaging protocol
12	Traffic Agent	CBR (Constant Bit Rate)	Generates steady data traffic

**Fig. 2.** Simulation of Delay Vs Buffer Size

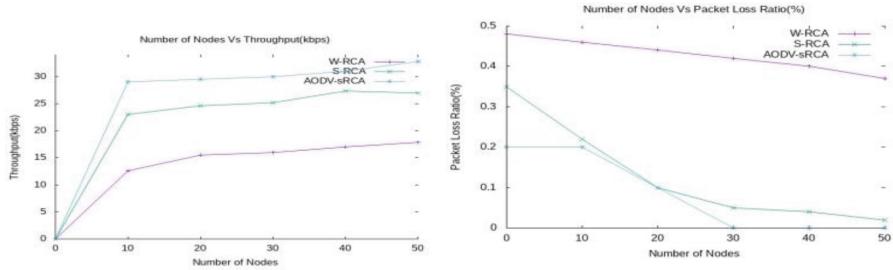


Fig. 3. Simulation of Number of nodes Vs Packet Loss Ratio, And Simulation of Throughput analysis

Table 2. Network Configuration and Simulation Parameters

	Ensemble Learning	Decision Tree	Naive Bayes
Accuracy	83.74%	77.83%	82.75%
Precision	86%	87%	84.34%
Recall	97%	87.13%	98%

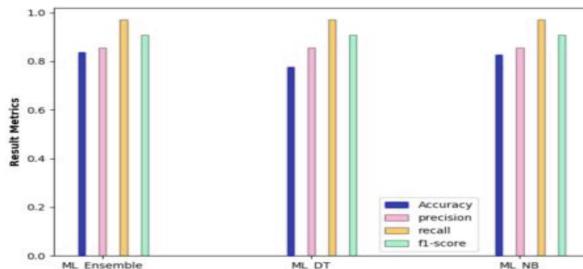


Fig. 4. Simulation of Number of nodes Vs Packet Loss Ratio, And Simulation of Throughput analysis

5 Conclusion

In this study, a smart approach based on Blockchain IoT is given. Resilient network performance, enhanced service quality, accelerated data transfer, and IoT data security were all examined in detail. An authentication method based on blockchain was implemented to protect data from unknown IoT nodes in the network. The performance of the classifier was improved with the addition of the innovative AODV-sRCA method. The ensemble classifier outperformed the decision tree at 77.83% and the NB classifier at 82.75%, with an accuracy of 83.74%, according to the results. Future studies could examine how much energy QoS settings consume and verify categorization accuracy with neural networks.

References

1. Kimbugwe, N., Pei, T., Kyebambe, M.N.: Application of deep learning for quality-of-service enhancement in internet of things: a review. *Energies* **14**(19), 6384 (2021)
2. Rathore, S., Park, J.H., Chang, H.: Deep learning and blockchain-empowered security framework for intelligent 5G-enabled IoT. *IEEE Access* **9**, 90075–90083 (2021)
3. Alnazir, A., Mokhtar, R.A., Alhumyani, H., Ali, E.S., Saeed, R.A., Abdel-Khalek, S.: Quality of services based on intelligent IoT WLAN MAC protocol dynamic real-time applications in smart cities. *Comput. Intell. Neurosci.* **2021**, 2287531 (2021)
4. Sodhro, A.H., et al.: Quality of service optimization in an IoT-driven intelligent transportation system. *IEEE Wirel. Commun.* **26**(6), 10–17 (2019)
5. Patan, R., Ghantasala, G.P., Sekaran, R., Gupta, D., Ramachandran, M.: Smart healthcare and quality of service in IoT using grey filter convolutional based cyber physical system. *Sustain. Cities Soc.* **59**, 102141 (2020)
6. Ryu, M., Kim, J., Yun, J.: Integrated semantics service platform for the internet of things: a case study of a smart office. *Sensors* **15**(1), 2137–2160 (2015)
7. Alhajri, K., AlGhamdi, M., Alrashidi, M., Balharith, T., Tabdei, R.: Smart office model based on internet of things. In: The International Conference on Artificial Intelligence and Computer Vision, pp. 174–183. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-76346-6_16
8. Akribopoulos, O., et al.: Making p-space smart: Integrating IoT technologies in a multi-office environment. In: Mobile Wireless Middleware, Operating Systems, and Applications: 5th International Conference, Mobilware 2012, Berlin, 13–14 November 2012, Revised Selected Papers 5, pp. 31–44. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36660-4_3
9. Tanganelli, G., Vallati, C., Mingozzi, E.: Ensuring quality of service in the internet of things. *New Adv. Internet Things* 139–163 (2018)
10. Tu, W.: Data-driven QoS and QoE management in smart cities: a tutorial study. *IEEE Commun. Mag.* **56**(12), 126–133 (2018)
11. Sheikh, A., Kumar, S., Ambhaikar, A.: Improvement of QoS parameters of IoT networks using artificial intelligence. In: Ubiquitous Intelligent Systems: Proceedings of ICUIS 2021, pp. 1–13. Springer, Singapore (2022). https://doi.org/10.1007/978-981-16-3675-2_1
12. Vidhya, D., Shibu, S.: Survey on IoT for Future Smart Systems: Review of Present Challenges, Next Generation Trends and Perspective (2021)
13. Submitter, I.C.S.M.D.I., KB, S., Basavaraju, T.G.: QoS Aware Energy Efficient WSN Protocols using Machine Learning Algorithms (ICSMDI 2021) (2021)
14. Sheikh, A., Ambhaikar, A., Kumar, S.: Quality of services improvement for secure IoT networks. *Int. J. Eng. Adv. Technol.* **9**(2), 5127–5135 (2019)



Cyberattack Detector for Real-Time IoT Devices and Networks

Antonio Díaz-Longueira^(✉), Álvaro Michelena, Francisco Zayas-Gato,
Marta-María Álvarez-Crespo, Óscar Fontenla-Romero,
and José Luis Calvo-Rolle

CTC, CITIC, Department of Industrial Engineering, University of A Coruña, Rúa Mendizábal s/n, Campus de Esteiro, 15403 Ferrol, A Coruña, Spain
{a.diazl,alvaro.michelena,f.zayas.gato,marta.maría.alvarez.crespo,oscar.fontenla,jose.rolle}@udc.es

Abstract. The Internet of Things is one of the great technological revolutions of recent times. With interconnected devices and, in some cases, with sensitive information, it is essential to have a defense mechanism capable of guaranteeing the security of devices and networks. In this article, we propose a binary classifier based on simple machine learning techniques with a low computational impact that allows the detection of cyberattacks. These techniques will facilitate implementation on devices with low resources, allowing constant status monitoring. This classifier is validated with a dataset of cyberattacks from a real-time IoT infrastructure through a 10K-Fold cross-validation and a subsequent validation stage. The results are promising, indicating a robust and reliable detection of cyberattacks.

Keywords: Internet of Things · Cybersecurity · Cyberattack · Classification · Principal Component Analysis · Synthetic Minority Over-sampling Technique

1 Introduction

The Internet of Things (IoT) is revolutionizing the digital world. It allows everyday objects, such as household appliances, and more specific objects, such as industrial sensors, to connect and exchange data. This interconnection can be beneficial in various scenarios, from smart homes to industrial cases.

However, it also presents significant security challenges, as these devices are often the target of cyberattacks due to the private and dangerous data they exchange and their vulnerabilities. These attacks can range from poisoning the address resolution protocol to distributed denial of service attacks, posing a significant risk to any user [6]. For this reason, correctly and accurately detecting cyberattacks is of capital importance. IoT devices are usually characterized by having limited resources, so the complexity of the algorithms they can execute is restricted.

The development of this work took place thanks to the use of a public dataset consisting of real-time communications information from different IoT devices in regular operation and under attack. In [9], the authors, in addition to generating the dataset used in this article, propose the use of a quantized autoencoder to configure an anomaly detection system. On the other hand, in [2], they propose using Deep Learning techniques for the real-time detection of cyberattacks in IoT networks. [1] presents a slightly modified approach based on Big Data to reduce resource use and machine learning techniques and algorithms to detect Distributed Denial-of-Service (DDoS) attacks.

This article uses standard machine learning techniques and algorithms to develop a classifier that detects whether the IoT device or network is under attack. The techniques and algorithms are computationally simple to facilitate the implementation of the classifier on devices with more resource limitations. The aim is to continuously monitor network traffic and device behavior, allowing attacks to be detected instantly and guaranteeing the security and integrity of Internet of Things systems.

The rest of the paper is structured as follows. Section 2 details the dataset used for the research as well as the different tools, techniques, and metrics used. The configuration and results of the experiments developed are shown in Sect. 3. Finally, in Sect. 4, the document closes with the conclusions drawn and different lines of future jobs.

2 Materials and Methods

This section presents all materials and methods employed in the research's development.

2.1 Materials

In this subsection, the dataset used for the development of this research is detailed.

Dataset. The dataset used to develop this research is obtained from a real-time IoT infrastructure, where there were different connected devices. Those devices are:

- LED bulb with connection to ThingSpeak, LED_TS
- WiFi-enabled smart LED bulb, LED_WiFi
- Temperature sensor communicated by MQTT, T_Sensor

The dataset is composed of 112117 bidirectional Attributes of Network Traffic records, captured through the Zeek tool and the Flowmeter plugin. It has 82 variables related to the parameters and characteristics of network traffic, plus a final categorical variable where the type of behaviour is collected. This behavior can be normal behavior for any of the three devices or tampered behavior due to a cyberattack. A summary of this feature is shown in Table 1.

Table 1. Breakdown of behavior patterns and their quantity in the dataset

Pattern	Behaviour	Quantity
DOS_SYN_Hping	Attack	94659
ARP poisoning	Attack	7750
NMAP_UDP_SCAN	Attack	2590
NMAP_XMAS_TREE_SCAN	Attack	2010
NMAP_OS_DETECTION	Attack	2000
NMAP_TCP_scan	Attack	1002
DDOS_Slowloris	Attack	534
Metasploit_Brute_Force_SSH	Attack	37
NMAP_FIN_SCAN	Attack	28
T_Sensor	Normal	8108
LED_TS	Normal	4146
LED_WiFi	Normal	253

As the previous Table shows, this is a dataset where normal behavior patterns and attack patterns are unbalanced, with 12507 records of normal behavior against 110610 records under attack. This represents a imbalance close to 9 to 1. The dataset was obtained from the UC Irvine Machine Learning Repository [8].

2.2 Methods

This subsection briefly describes the machine learning techniques used to develop the classifier to detect cyberattacks on IoT devices and networks.

Synthetic Minority Over-Sampling Technique. The Synthetic Minority Over-sampling Technique (SMOTE) [3] is a technique used to perform data over-sampling. New data, from real samples, are generated from the minority class to convert an unbalanced classification problem into a balanced one. The generated points are samples similar to the existing ones, the K Nearest Neighbors.

Principal Component Analysis. Principal Component Analysis (PCA) is a dimensional reduction technique with the objective of finding the directions of greatest variability in a data set, known as principal components. For this, the eigenvalues of the correlation matrix are calculated. Then, using these eigenvectors, the initial set can be linearly transformed into a lower dimensional space [4].

Logistic Regression. Logistic Regression (LR) uses a sigmoidal function to calculate the probability of membership in a class. This sigmoidal function, also called logistic function, can take any real number and map it to a value between 0 and 1. The probability values obtained are adjusted following a gradient descent criterion. This technique can be implemented with different solvers.

K Nearest Neighbors. The K-Nearest Neighbors (KNN) algorithm is a supervised learning technique used for classification and regression tasks. It uses the premise that data points of the same class are close in space. During prediction, it calculates the distance between the input point and all training points, identifying the K nearest neighbors, assigning the most common label among the K nearest neighbors [10].

Linear Discriminant Analysis. The Linear Discriminant Analysis (LDA) method fit conditional class densities to the data and uses Bayes' rule to create a linear decision frontier. The model fits a Gaussian density to each class, assuming that all classes share the same covariance matrix. Different solvers can be applied [12].

Naive Bayes. In this case, we are really talking about a family of methods, the Naive Bayes (NB) methods. They are a group of algorithms based on Bayes' theorem and that assume, in a naive way, that all characteristics are independent of each other, given the value of the class variable. The different methods differ depending on the implementation carried out [11].

3 Experiments Setup and Results

This section presents a detailed overview of the experimental process and the subsequent results obtained in the cyberattack detection.

3.1 Experiments Setup

This section explains the steps and procedures used to develop the work. Subsequently, the subsections corresponding to each section detail the configurations and adjustments of the methods used and the metrics used to evaluate the classifier's performance.

First, a preprocessing operation is performed on the original dataset, adapting the data for the experiments. This operation only codes the behavior variable of the collected communication, coding it as 0 if it is a normal behavior and 1 if it is a behavior adulterated by a cyberattack. Then, a dimensional reduction was applied to reduce the detector input variables and thus try to reduce the computational cost. After selecting the appropriate number of components, depending on the explained variance, a cross-validation of 10K-Fold is carried out with 70% of the data set on the tested models. Several configurations are tested for each technique and algorithm, looking for the best performance model. Due to the existing imbalance between both classes, close to 9:1, an oversampling technique was applied for the cross-validation phase. The data set used for training was augmented with this technique, while validation was performed without oversampling. Afterward, and depending on the metrics obtained, the best configuration of each technique will be selected that will finally be validated with 30% of the remaining dataset.

All experiments were conducted on a Windows 10 Enterprise machine with an Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20 GHz and 64 GB of RAM. The work was developed using the Python 3.11 programming language and the scikit-learn [7] machine learning library.

Feature Reduction. In order to reduce the number of input variables used by the classifier to detect a cyberattack, a features reduction will be carried out by applying PCA to the dataset and selecting those components that have a higher percentage of variance explained.

Classification Techniques. Different techniques are tested to find the best cyberattack detector. In addition, each of the techniques presented and evaluated is adjusted through its hyperparameters to obtain the best performance of each technique. The different configurations are mentioned below.

Logistic Regression. Ten Logistic Regression models were deployed, with regularization strength values of: 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, and 1.

K Nearest Neighbors. Eighteen KNN models were tested, modifying the number K of neighbors used for classification. The K values tested are: 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 75, 100, 125, 150, 250, 500, 750, and 1000.

Linear Discriminant Analysis. The search for the best Linear Discriminant Analysis configuration led to the deployment of two models with different solvers: Singular Value Decomposition (SVD) and Least Squares Solution (LSQR).

Naive Bayes. Two implementations of the Naive Bayes method were carried out: the first assumes the presence of a multivariate Bernoulli distribution, while the second uses a Gaussian classification.

Model Evaluation. Different metrics were used to evaluate and compare the model's performance. The metrics used to analyze the behavior of the techniques are accuracy (see Eq. 1), area under the Receiver Operating Characteristic curve [5] (AUC), F1-score (see Eq. 2), and specificity (see Eq. 3). In turn, training time will be considered as data to evaluate resource consumption.

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}} \quad (1)$$

$$\text{F1-score} = \frac{2 \times \text{True Positive}}{2 \times \text{True Positive} + \text{True Positive} + \text{False Negative}} \quad (2)$$

$$\text{Specificity} = \frac{\text{True Negative}}{\text{True Negative} + \text{False Negative}} \quad (3)$$

3.2 Results

This section will present the several results obtained in the different experiments carried out during the development of a cyberattack detector.

Feature Reduction. The Fig. 1 shows, in a bar chart, the amount of variance explained by the components. It can be seen that a single principal component explains more than 80% of the variance. From there, the explained variance is drastically reduced until, starting with the fifth principal component, the explained variance is close to 0%.

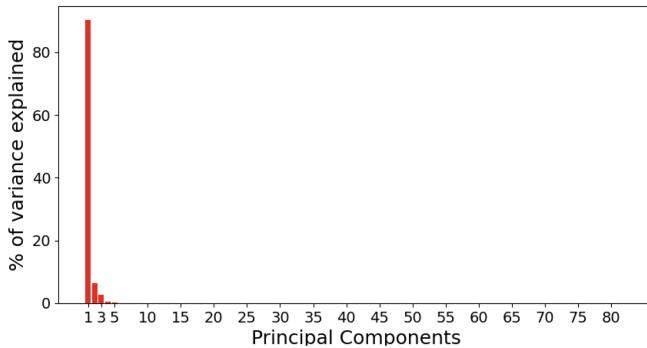


Fig. 1. Results of principal component analysis of the dataset

As a commitment between the reduction of the number of variables to be used by the model, reducing the complexity, size and computational cost of it; and a good performance of this, five components will be used for the next experiments.

Classification Techniques Results. Table 2 specifies the value of the hyperparameters for each technique that obtained the best performance in the cross-validation phase. For LR the regularization strength is 0.0001. In the case of the KNN technique, the optimal number of neighbors is 5. For LDA, the solver used is SVD. Finally, for the NB method, the implementation used is Gaussian.

Table 2. Model hyperparameters with the best performance

Technique	Hyperparameter	Value
LR	Regularization strength	0.0001
KNN	Number of neighbors	5
LDA	Solver	SVD
NB	Implementation	Gaussian

Using these configurations, Table 3 indicates the mean value obtained in the different metrics during the cross-validation process. The LR shows a high accuracy of 0.9070, although its training time is relatively long at 0.3086 s. On the other hand, the KNN model stands out in several metrics, with the highest precision of 0.9947, the highest AUC of 0.9925, and the highest F1_score of 0.9971, suggesting a solid balance between precision and recall. However, it also has the longest training time at 0.8672 s. The LDA has the highest specificity of 0.9982, indicating that it effectively avoids false negatives. Although its training time is relatively short at 0.2297 s, its performance on other metrics is not as high as the other models. Finally, the Naive Bayes model has the highest recall of 0.9748, which indicates that it is very effective in detecting all attacks. It also has the shortest training time at 0.1031 s. However, AUC is the lowest of the four models at 0.8168.

Table 3. Metrics's values in the cross-validation phase

Technique	Accuracy	AUC	F1_score	Specificity	Training time (s)
LR	0.9070	0.9471	0.9454	0.9974	0.3086
KNN	0.9947	0.9925	0.9971	0.9898	0.8672
LDA	0.8660	0.9246	0.9194	0.9982	0.2297
NB	0.9426	0.8168	0.9683	0.6588	0.1031

Figures 2, 3, 4 and 5 shows, in a boxplot, the value of the metrics for the four techniques during this cross-validation process.

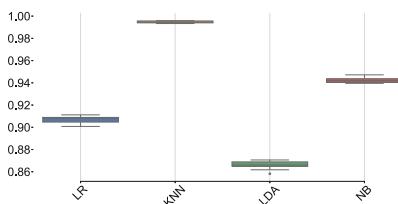


Fig. 2. Accuracy boxplot

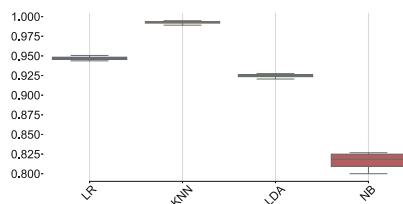
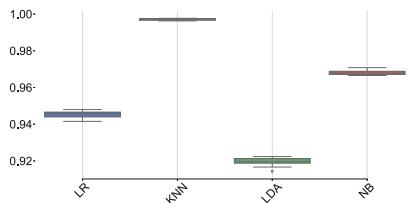
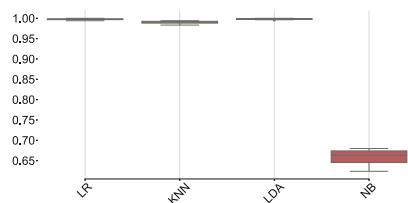


Fig. 3. AUC boxplot

Through these Figures, it is verified that the variability of the results of the four metrics is relatively low throughout the entire cross-validation process since the height of the boxes is low. It is observed that the NB method obtains the worst results compared to the other techniques, presenting more significant

**Fig. 4.** F1_score boxplot**Fig. 5.** Specificity boxplot

variability in its performance. It is also seen how the KNN method obtains the best results with minimal variability, which indicates that the model fits the data well and can correctly classify any scenario.

The Table 4 shows the results obtained by each of the techniques, adjusted with the best configuration, in the validation phase. The best value of each metric stands out in bold.

Table 4. Metrics' values in the validation phase

Technique	Accuracy	AUC	F1_score	Specificity	Training time (s)
LR	0.9092	0.9485	0.9468	0.9978	0.5156
KNN	0.9949	0.9955	0.9972	0.9900	2.4375
LDA	0.8673	0.9251	0.9202	0.9978	0.4375
NB	0.9440	0.8196	0.9691	0.6633	0.3125

In the same way as occurred in the cross-validation, the KNN technique achieves the best results in three of the four metrics, with high values of accuracy, AUC and F1_score: 0.9949, 0.9955, 0.9972, respectively. However, it has the highest training time of the four techniques used, although it is low. From the previous Table, it is clear that the technique with the worst performance is NB, although its training time was the fastest. The other two techniques, LR and LDA, although they stand out the most in the specificity metric, with a value of 0.9978 for both, in the rest of the metrics they do not match the performance of KNN.

Figure 6 shows the confusion matrix obtained by the KNN technique in the validation phase. It is confirmed that the error rate of the classifier is very low, an order of magnitude below the successes in normal behavior, and two orders of magnitude below in cyberattack detection.

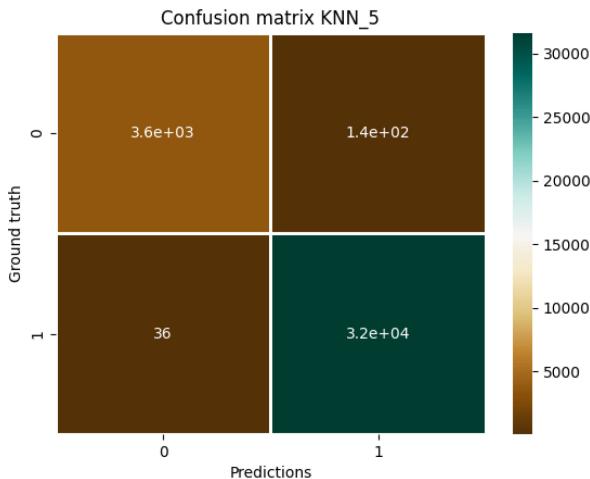


Fig. 6. Confusion matrix in the validation phase

4 Conclusions and Future Works

This article presents different classification techniques, adjusted for the detection of cyberattack on IoT devices. Low complexity techniques were used, thus allowing its execution on low-resource devices. In turn, a dimensional reduction and oversampling technique were applied to improve the performance of the models. After adjusting the techniques, promising results were achieved, with the KNN technique ranking as the technique with the best performance, despite having the highest training time, 0.86 s. This model achieves a precision of greater than 0.99, an AUC greater than 0.99 with an F1 score of 0.9982.

In order to improve the classifier's performance, other techniques can be applied, always taking into account the low resources of the most common devices in IoT networks. Also, to facilitate the response before the detection of a cyberattack, a classification of what attack is taking place would be of great help.

Acknowledgments. Xunta de Galicia. Grants for the consolidation and structuring of competitive research units, GPC (ED431B 2023/49).

Álvaro Michelena's research was supported by the Spanish Ministry of Universities (<https://www.universidades.gob.es/>), under the “Formación de Profesorado Universitario” grant with reference FPU21/00932.

This research is the result of the Strategic Project “Critical infrastructures cyber-secure through intelligent modeling of attacks, vulnerabilities and increased security of their IoT devices for the water supply sector” (C061/23), as a result of the collaboration agreement signed between the National Institute of Cybersecurity (INCIBE) and the University of A Coruña. This initiative is carried out within the framework of the funds of the Recovery Plan, Transformation and Resilience Plan funds, financed by the European Union (Next Generation).

CITIC is funded by the Xunta de Galicia through the collaboration agreement between the Consellería de Cultura, Educación, Formación Profesional e Universidades and the Galician universities for the reinforcement of the research centres of the Galician University System (CIGUS).

Antonio Díaz-Longueira's research was supported by the Xunta de Galicia (Regional Government of Galicia) through grants to Ph.D (<http://gain.xunta.gal>), under the "Axudas á etapa predoutoral" grant with reference: ED481A-2023-072.

References

1. Awan, M.J., et al.: Real-time DDoS attack detection system using big data approach. *Sustainability* **13**(19), 10743 (2021)
2. Callegari, C., Buccianeri, E., Giordano, S., Pagano, M.: Real time attack detection with deep learning. In: 2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), pp. 1–5 (2019). <https://doi.org/10.1109/SAHCN.2019.8824811>
3. 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)
4. Jenatton, R., Obozinski, G., Bach, F.: Structured sparse principal component analysis. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 366–373. JMLR Workshop and Conference Proceedings (2010)
5. McNeil, B.J., Hanley, J.A.: Statistical approaches to the analysis of receiver operating characteristic (ROC) curves. *Med. Decis. Making* **4**(2), 137–150 (1984)
6. Mosenia, A., Jha, N.K.: A comprehensive study of security of internet-of-things. *IEEE Trans. Emerg. Top. Comput.* **5**(4), 586–602 (2017). <https://doi.org/10.1109/TETC.2016.2606384>
7. Pedregosa, F., et al.: Scikit-learn: machine learning in python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
8. RT-IoT2022. UCI Machine Learning Repository (2024). <https://doi.org/10.24432/C5P338>
9. Sharmila, B.S., Nagapadma, R.: Quantized autoencoder (QAE) intrusion detection system for anomaly detection in resource-constrained IoT devices using rt-iot2022 dataset. *Cybersecurity* **6**, 1–15 (2023). <https://api.semanticscholar.org/CorpusID:261516162>
10. Syriopoulos, P.K., Kalampakakis, N.G., Kotsiantis, S.B., Vrahatis, M.N.: KNN classification: a review. *Ann. Math. Artif. Intell.* 1–33 (2023)
11. Webb, G.I., Keogh, E., Miikkulainen, R.: Naïve Bayes. *Encyclopedia Mach. Learn.* **15**(1), 713–714 (2010)
12. Xanthopoulos, P., Pardalos, P.M., Trafalis, T.B., Xanthopoulos, P., Pardalos, P.M., Trafalis, T.B.: Linear discriminant analysis. *Robust Data Mining*, pp. 27–33 (2013)

ICEUTE: General Track



Conducting Final Programming Exams with Auto-Grading and Code Evaluation Tools

José Luis Subirats¹, Gabriel Luque-Polo², and Rafael M. Luque-Baena^{2,3(✉)}

¹ Department of Computer Languages and Computer Science, University of Málaga, Málaga, Spain

² ITIS Software, University of Málaga, C/ Arquitecto Francisco Peñalosa, 18, 29010 Málaga, Spain
rmluque@uma.es

³ Biomedical Research Institute of Málaga (IBIMA), C/ Doctor Miguel Díaz Recio, 28, 29010 Málaga, Spain

Abstract. Programming is a fundamental subject in the majority of technology-oriented degrees, particularly in Computer Engineering programs. Mastering programming requires continuous practice to understand the syntax, semantics, and execution of the chosen programming language, as well as developing the problem-solving mindset necessary for effective coding. Rapid feedback on errors made during exercises is essential for skill improvement. However, manual grading of practices and exams is a challenging and time-consuming task in continuous assessment systems. This work aims to demonstrate the application of an existing tool, VPL (Virtual Programming Lab), integrated with JUnit tests, for auto-grading final programming exams. The proposal has been successfully implemented in the Object-Oriented Programming (OOP) course, yielding highly satisfactory results.

Keywords: code evaluation tools · auto-grading exams · virtual programming laboratory

1 Introduction

In today's society, programming has become an essential skill across various fields and industries. The ability to code is crucial for software development and data analysis, automation, and even decision-making processes in business and research environments. The growing reliance on technology and data-driven approaches has significantly increased professionals' demand for programming proficiency. According to the World Economic Forum's Future of Jobs Report¹, programming and related technical skills are among the top competencies required in the future job market.

With the advent and success of Large Language Models (LLMs) such as OpenAI's GPT-3 and GPT-4 [1], there has been a surge in the popularity of AI-assisted tools for programming support [2]. These tools can help by generating

¹ http://www3.weforum.org/docs/WEF_Future_of_Jobs_2020.pdf.

code snippets, suggesting corrections, and automating mundane tasks. However, more is needed for programmers to possess a solid understanding of coding principles. Often, the code provided by these AI tools can be incorrect [3] or fail to meet stringent quality requirements, necessitating thorough knowledge and critical assessment skills from the users.

Learning to program is inherently challenging and demands extensive practice [4]. For students, having reliable methods to validate their code is essential to their learning process. Test cases and automated testing tools are particularly valuable for novice programmers, as they provide immediate feedback and identify errors that need correction. Automating these evaluations can significantly aid students by ensuring consistency and efficiency in the learning process. Research has demonstrated that automated feedback systems can improve learning outcomes by providing timely and precise feedback, which is critical for skill development in programming [5].

This document details the methodology employed in an educational innovation program that utilizes the “Virtual Programming Laboratory” (VPL) tool in Moodle [6] for conducting exams and practical exercises in the Object-Oriented Programming (OOP) course in Java. The primary objective of this program is to improve the evaluation and self-assessment process for students, providing a more dynamic and efficient environment for both students and instructors. Teaching OOP is fundamental in computer science and engineering programs. However, traditional evaluation methods for this course can be labor-intensive and inefficient. Traditional exams require considerable time for grading and do not always offer immediate feedback to aid student learning. Therefore, it was decided to implement VPL in Moodle, a tool that automates code evaluation through unit tests, to enhance the efficiency and quality of the assessment process.

By integrating VPL, the program aims to provide a structured and scalable approach to evaluating student code, ensuring that assessments are fair and informative. This approach facilitates immediate feedback and allows students to engage in self-assessment and iterative improvement of their coding skills. The innovative use of VPL in the OOP course is expected to set a new standard for programming education, aligning with modern educational practices emphasizing active learning and continuous assessment. However, in this environment, the generation and automation of test cases is quite basic, only allowing string comparisons. This significantly limits the ability to verify code that does not produce text output, requiring the code to be rewritten to generate text strings that can be used to check its validity using VPL test cases. To provide students with a more accessible solution via Moodle, and a robust, automated method for code assessment, JUnit-generated test cases have been integrated into the VPL environment. This integration renders VPL an optimal setting for the instruction of programming courses within Engineering degree programs.

The rest of this document is organized as follows: Sect. 2 reviews related works and sets the context for this study. Then, in Sect. 3, the methodology used to integrate VPL with JUnit-generated test cases in the OOP course is described. Later, Sect. 4 presents the analysis of the results, discussing the impact of the VPL tool

on the evaluation process and student learning outcomes. Finally, Sect. 5 provides conclusions and outlines future works, suggesting potential improvements and expansions of this educational innovation program.

2 Related Works

Regarding the automatic evaluation of code for programming assignments, there are numerous references and studies that list and describe the most commonly used tools [7–9]. More recently, new stand-alone frameworks have emerged, incorporating modern designs. Notable examples include code evaluation tools such as Codio [10], HackerRank [11], and Codewars [12], which are instrumental in learning to program in various languages, including Java. These platforms provide a structured environment where students and aspiring developers can practice coding skills and receive immediate feedback. Codio, for instance, offers an integrated development environment (IDE) tailored for educational purposes, supporting numerous programming languages and features like real-time collaboration, making it an excellent tool for classroom settings and remote learning.

HackerRank focuses on competitive programming and skills assessment, offering coding challenges that cater to different proficiency levels. For students learning Java, HackerRank provides specific challenges to reinforce core concepts such as object-oriented programming, data structures, and algorithms. Codewars gamifies coding practice with kata (coding challenges) that users can solve to gain points and progress through levels, making learning engaging and motivating. It covers problems from basic syntax to complex concepts and allows users to see multiple solutions to the same problem, fostering a deeper understanding of coding techniques. Collectively, these tools contribute to a comprehensive and interactive learning experience, enabling students to master programming languages effectively.

Several tools integrate seamlessly with Moodle to facilitate code evaluation and programming practice, offering the initial advantage that students are already familiar with the environment, thereby reducing the learning curve for the new tool. In addition to VPL, we could mention Coderunner, which is also a Moodle plugin designed for assessing programming assignments. Coderunner supports a wide range of programming languages and allows instructors to create custom question types that automatically compile and test student code. It provides detailed feedback on code correctness, performance, and style, helping students to learn from their mistakes and improve their coding skills. Nevertheless, while Coderunner focuses on automated evaluation of programming questions with customizable test cases and immediate feedback, ideal for quizzes and exams, VPL offers a complete development environment with execution and debugging capabilities, suitable for more complex assignments requiring interactive programming. Blockly [13], although not specifically a code evaluation tool, is another useful plugin adapted from a Google implementation that integrates with Moodle to teach programming concepts. It employs a visual programming approach where students can drag and drop code blocks to create programs. This

method can be particularly beneficial for beginners who are learning the fundamentals of programming before progressing to text-based coding in languages such as Java.

3 Methodology

This section describes the process followed to create an exam and configure it for evaluation using VPL and a series of test cases defined in JUnit. The Subsect. 3.1 provides a more detailed explanation of the utility and functionality of the VPL tool, while the subsequent subsections outline the specific steps for managing the exam. These steps include preparation (Subsect. 3.2), configuration in the VPL environment (Subsect. 3.3), execution of the exam, which can be launched directly from the Moodle tool due to its versatility (Subsect. 3.4), and finally, evaluation and feedback to students for improving and better understanding the implemented code (Subsect. 3.5).

3.1 Description of the Tool: Virtual Programming Laboratory (VPL)

VPL is a Moodle module that allows students to upload and execute programs directly within the Moodle environment. This module is particularly useful for programming courses, as it enables instructors to set up practical exercises and exams that students can solve in real time, receiving immediate feedback on their solutions. At the operational level, the VPL architecture is composed of three modules:

- Moodle Plugin: Integrated as a new assignment within this platform. Within the activity, you can configure aspects such as the delivery date, the number of files, the minimum or maximum score, execution or evaluation options, as well as the black box battery.
- Execution Server (jail-cell): Responsible for actually executing the tests. For each evaluation, an execution order is created on the server, independent of the rest of the system's executions, ensuring the isolation of the various executions.
- Source Code Editor: Allows files to be edited for submission directly from the browser. The editor enables typical editing operations without having access to the execution machine at any time.

Thus, the execution server and the submission server are on separate machines. Access to the Moodle system is done through the browser. When the user requests an execution or evaluation, the jail server receives requests from the module and responds to it via the XML-RPC protocol.

3.2 Exam Preparation

The first step in preparing the exam was defining the content to be assessed. It was decided that the exam would consist of several Java programming exercises,

each of which would be evaluated using unit tests. The exercises were designed to cover fundamental OOP concepts, including classes, objects, inheritance, polymorphism, and exception handling.

For each exam exercise, test classes were prepared using JUnit 4.0. These classes contain the test cases that would validate the solutions provided by the students. In total, 11 test classes were designed, each with approximately 10 test cases, totaling 111 test cases. The test cases were designed to cover a wide variety of scenarios and ensure that the solutions were correct and robust.

A detailed exam statement was drafted, clearly explaining the requirements for each exercise. The statement included examples and clarifications to avoid any ambiguity. This document was made available to the students at the beginning of the exam.

3.3 Exam Configuration in VPL

Once the content and tests were ready, the next step was to configure the exam in the VPL environment of Moodle. This involved creating a new VPL activity for each exam exercise. Within each activity, the corresponding JUnit test classes were uploaded, and the execution and evaluation parameters were configured.

Before releasing the exam to the students, several course instructors took the exam as if they were students. This step was crucial to ensure that all aspects of the exam worked correctly and that there were no errors in the test cases or VPL configuration. Any issues identified during this phase were corrected before proceeding. In addition to internal validation, it was decided to pass the exam to an independent entity for resolution. In this case, ChatGPT was used to solve the exam. This step provided an additional layer of verification, ensuring that the exam was well-structured and that the test cases adequately covered all possible scenarios.

3.4 Exam Execution

The exam was taken by the students in a controlled environment, using the Moodle platform with the VPL module configured in advance. During the exam, students could write and execute their code, receiving immediate feedback on their test results. This allowed them to correct errors and improve their solutions in real time. Technical and academic support was provided during the exam to assist students in case they encountered issues with the platform or with understanding the exercises.

3.5 Evaluation and Feedback

Thanks to the VPL configuration and JUnit test classes, the exam grading was automatic. The test results provided immediate and accurate evaluation of the students' solutions. The failed test cases offered a clear view of the errors committed, facilitating quick and efficient correction. After the exam, each student

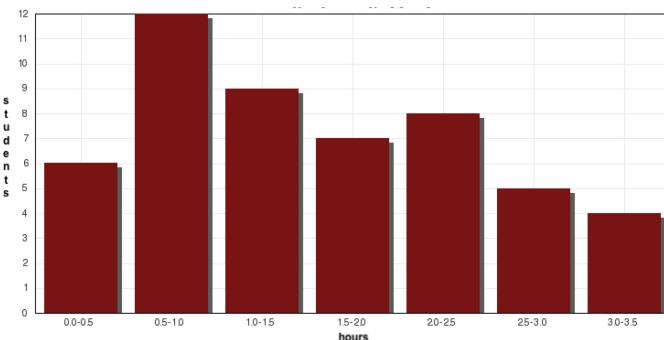


Fig. 1. Final exam duration which show the complexity of the task.

received a detailed report with the results of their tests. This report included a description of the failed test cases and suggestions on how they could improve their solutions. This feedback was very valuable for the students, as it allowed them to identify and correct their errors.

4 Experimental Results

This methodology has been applied in the course “Object-Oriented Programming,” which is part of the first year of the four-year “Bachelor’s Degree in Computer Engineering.” It is the second introductory programming course in the degree program, with Java as the primary language of instruction. In the academic year in question, 77 students were enrolled, although class and lab attendance was significantly lower, around 50–60% of the total students.

Specifically, eight practical assignments and three exams (two midterms and one final exam) were implemented using this approach. It is noteworthy that all these tasks were of high complexity, not just simple programming exercises. They included paradigms such as inheritance, polymorphism, dynamic binding, and including Java classes for collections, I/O operations, and string manipulation. This course does not cover graphical user interface development, so any algorithm outputs, if present, are text-based. Across all tasks and considering all students who used this approach, more than 4500 code evaluations were conducted, a significant number that demonstrates the tool’s utility throughout the course.

Following the final exam conducted using this proposal, we can observe some results obtained by the platform and aggregated to draw conclusions. To highlight the complexity of the test taken by the students, Fig. 1 shows the time taken by the students to complete it. Although some students finished the test in less than an hour, it is observed that the average time ranges between 1.5 and 2 h for the duration of the exam.

Another interesting result is the analysis of the number of attempts made in relation to the final grade obtained. Figure 2 analyzes the clusters to understand different student behaviors and performance patterns, taking into account that

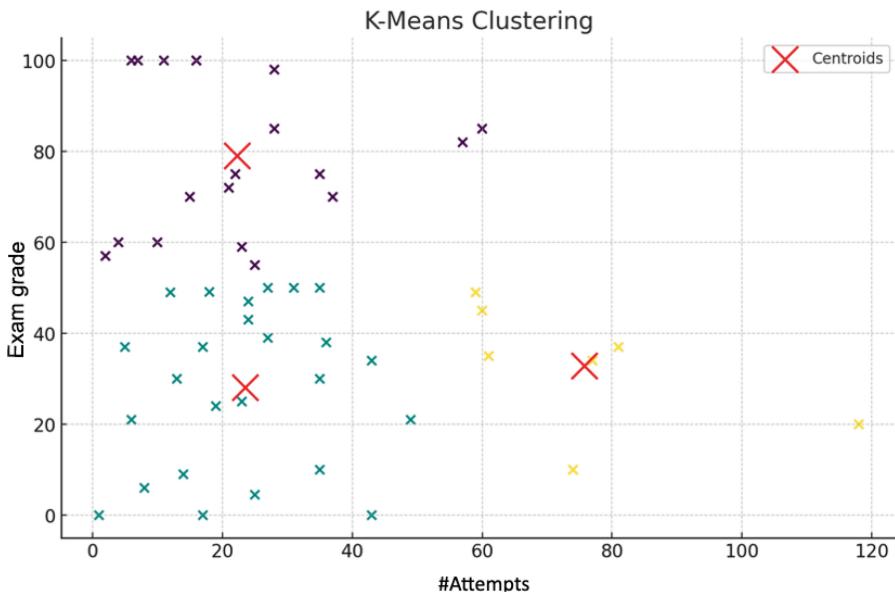


Fig. 2. Cluster representation of the relation between the exam grade of the students and the number of attempts performed to evaluate the code of the exam.

the x-axis represents the number of attempts to evaluate the code and the y-axis represents the final grade. Analyzing this, we can highlight:

- Cluster 0 (Centroid: (22.26, 79.11)) consists of students who made a relatively high number of attempts (around 22 on average) and achieved high final grades (around 79 on average). These students likely exhibit persistence and diligence. The high number of attempts suggests that they frequently revisited and re-evaluated their code to improve their performance, which is reflected in their high final grades.
- Cluster 1 (Centroid: (23.48, 28.14)) consists of students who made a similar number of attempts as Cluster 0 (around 23 on average) but achieved much lower final grades (around 28 on average). These students might be struggling with the course material. Despite their numerous attempts, their low final grades indicate they might not be effectively addressing the issues in their code or could be having conceptual misunderstandings.
- Cluster 2 (Centroid: (75.71, 32.86)) includes students who made a high number of attempts (around 75 on average) and achieved moderate final grades (around 33 on average). These students are making many attempts but still achieving relatively low grades, suggesting significant difficulties in mastering the material. The high number of attempts shows effort, but the low grades indicate persistent challenges.

Finally, Fig. 3 shows the average grades obtained after the teacher's evaluation, with a value of 47.78, compared to the average grades obtained after the

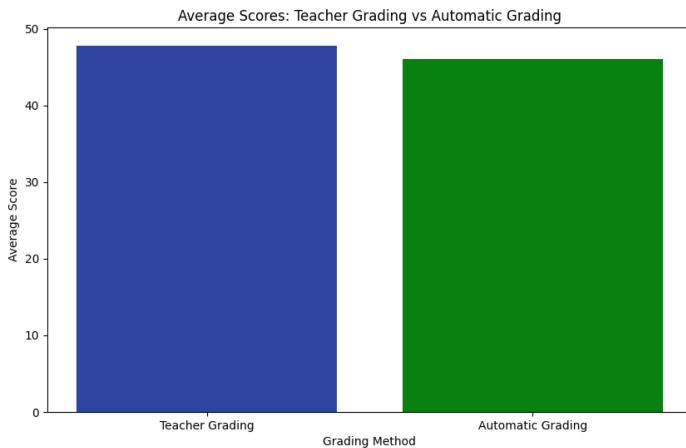


Fig. 3. Average of the exam grades.

automatic evaluation, with a value of 46.01. Both grades are on a 0–100 scale. It is observed that the error is very small, which demonstrates the suitability of this type of tool for the tasks at hand.

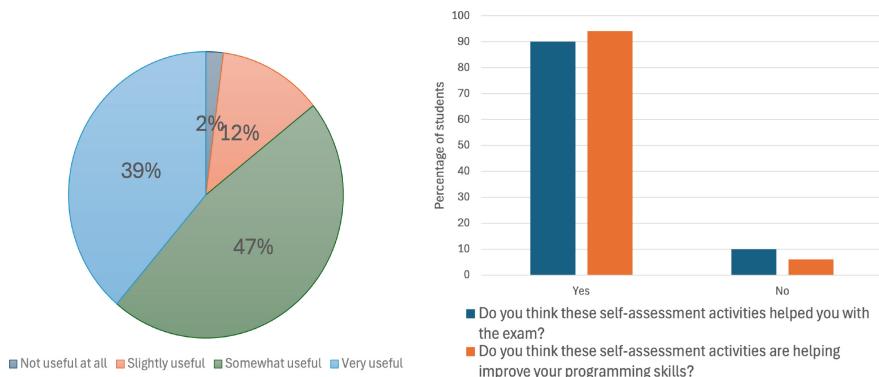


Fig. 4. On the right: usefulness of feedback after the first attempt. On the left: usefulness of self-assessment tasks for helping with the exam and for improving programming skills.

Additionally, to assess the effectiveness of the new evaluation method, a survey was conducted among the students after the exam. The survey results (Fig. 4) indicated that most students found the immediate feedback provided by VPL useful and that the overall exam experience was positive. Students appreciated the opportunity to correct their errors in real time and valued the clarity and precision of the evaluation.

5 Conclusions

The implementation of “Virtual Programming Laboratory” in Moodle for evaluating the OOP course in Java was a success. Although the exam preparation process was long and costly compared to traditional methods, the automatic grading and immediate feedback provided significant benefits for both students and instructors. Students were able to improve their programming skills in real time, and instructors could spend less time grading and more time on teaching and academic support.

This educational innovation program demonstrates the potential of technological tools to improve the quality and efficiency of education. As educational technologies continue to advance, we are likely to see more initiatives like this that transform the way teaching and assessment are conducted in educational institutions.

For future implementations, the following improvements could be considered:

- Expansion of Test Cases: Include more test cases to cover a broader range of scenarios and ensure even more comprehensive evaluation.
- Automated Statement Generation: Develop a tool to automate the generation of the exam statement based on the defined exercises and tests.
- Comparative Evaluation: Conduct comparative studies between different cohorts of students to measure the long-term impact of using VPL on learning and academic performance.
- Advanced Support: Implement a more robust technical and academic support system during the exam, possibly using AI tools to assist students in real time.

The experience gained from this project will serve as a foundation for the implementation of future educational innovations, contributing to the advancement of programming education and other technical disciplines in higher education.

References

1. Brown, T., et al.: Language models are few-shot learners. *Adv. Neural. Inf. Process. Syst.* **33**, 1877–1901 (2020)
2. Barke, S., James, M.B., Polikarpova, N.: Grounded copilot: how programmers interact with code-generating models. In: *Proceedings of the ACM on Programming Languages*, vol. 7, pp. 85–111 (2023)
3. Liu, J., et al.: Is your code generated by ChatGPT really correct? Rigorous evaluation of large language models for code generation. In: *Advances in Neural Information Processing Systems*, vol. 36 (2024)
4. Tan, P.-H., Ting, C.-Y., Ling, S.-W.: Learning difficulties in programming courses: undergraduates’ perspective and perception. In: *2009 International Conference on Computer Technology and Development*, vol. 1, pp. 42–46. IEEE (2009)
5. Bey, A., Jermann, P., Dillenbourg, P.: A comparison between two automatic assessment approaches for programming: an empirical study on MOOCs. *J. Educ. Technol. Soc.* **21**(2), 259–272 (2018)

6. Thiébaut, D.: Automatic evaluation of computer programs using moodle's virtual programming lab (VPL) plug-in. *J. Comput. Sci. Coll.* **30**(6), 145–151 (2015)
7. Aldriye, H., Alkhalfaf, A., Alkhalfaf, M.: Automated grading systems for programming assignments: a literature review. *Int. J. Adv. Comput. Sci. Appl.* **10**(3) (2019). <https://doi.org/10.14569/IJACSA.2019.0100328>
8. Caiza, J.C., Del Alamo, J.M.: Programming assignments automatic grading: review of tools and implementations. In: INTED Proceedings. 7th International Technology, Education and Development Conference. IATED, pp. 5691–5700 (2013). ISBN 978-84-616-2661-8
9. Keuning, H., Jeuring, J., Heeren, B.: Towards a systematic review of automated feedback generation for programming exercises. In: Proceedings of the ACM Conference on Innovation and Technology in Computer Science Education, pp. 41–46 (2016). <https://doi.org/10.1145/2899415.2899422>
10. Codio Ltd. Codio: The Integrated Development Environment for Education (2024). <https://www.codio.com/>
11. HackerRank. HackerRank: Technical Hiring Platform (2024). <https://www.hackerrank.com/>
12. Codewars. Codewars: Achieve Mastery Through Coding Practice and Challenges (2024). <https://www.codewars.com/>
13. Seraj, M., et al.: Scratch and google Blockly: how girls' programming skills and attitudes are influenced. In: Proceedings of the 19th Koli Calling International Conference on Computing Education Research (2019). <https://doi.org/10.1145/3364510.3364515>



Educational Metaverse

Alberto Martínez-Gutiérrez^{1,3} , Javier Díez-González^{1,3} ,
Rubén Ferrero-Guillén^{1,3} , Iván Sánchez-Calleja^{1,3} , Pablo Alonso-Diez² ,
Madalena Araújo³ , and Hilde Perez^{1,3}

¹ Department of Mechanical, Computer and Aerospace Engineering,
Universidad de León, 24071 León, Spain
amartg@unileon.es

² Faculty of Geagraphic and History, Universidad de Salamanca,
37071 Salamanca, Spain

³ ALGORITMI, Universidade do Minho, Escola de Engenharia,
Campus Azurem, Guimaraes, 4800-058 Braga, Portugal

Abstract. Education is a process that has been key to social and technological development through the acquisition of ideas, skills, and knowledge. For this reason, innovations in this field are essential to improve and share this knowledge and experience of people. In this context, digitization has been a major breakthrough, providing flexibility and global access never seen before in education. However, one of the limitations of digital educational platforms is the immersive interaction between the students and the teacher, which is key in training and the generation of crucial experiences in the learning process. For this reason, the authors have developed a virtualized environment called metaverse where humans can see each other and share experiences and knowledge in a completely virtual environment. This novel platform has been tested on engineering students who have performed an internship in both the virtual and the real world in order to compare them. The perception of the students has been collected through a survey where very promising results have been obtained to improve the educational metaverse as a space for academic and human growth.

Keywords: Industrial metaverse · Digital Training · Digital Twin · Virtual Reality · Education 5.0

1 Introduction

The transfer of knowledge and experience across generations has led to unprecedented progress in human evolution [1]. From the earliest teachings to the most current innovations, education has been a driving force in the generation of knowledge for the benefit of society. This exchange is not only limited to the direct transmission of information [2], but also encompasses the ability to express ideas, concepts, and impressions about nature through various media such as literature, art, and cultural expression. Therefore, we are facing a continuous process where humans acquire knowledge, skills, values, and attitudes that make up education [3].

Education has been undergoing a transformation process for years, where the digitalization of methodologies is being the protagonist in the acquisition of knowledge and recently of skills [4]. This process is transforming the way knowledge is accessed, shared, and assimilated, democratizing learning with greater accessibility to information. In addition, digitization allows delocalization and flexibility in both time and learning methods according to individual needs. For this purpose, digital platforms are used to facilitate educational management and the transmission of multimedia resources [5].

Furthermore, educational platforms have also been developed for practice-oriented virtual laboratories [6] where the experience is simulated remotely using exclusively computer resources. For this purpose, they use interactive simulations based on software to recreate experiments whose implementation in reality would have to involve significant human and material resources, no longer needed. Therefore, they are particularly suited to applications, where they have been mostly used.

To improve the experience and therefore the acquisition of knowledge, it is necessary a more immersive experience where humans perceive reality with more senses than traditional interfaces (e.g., computer, tablet, ...) [7]. For this reason, new technologies such as Virtual Reality (VR) or Digital Twins (DT) [8] have recently been used to virtually recreate an environment where physical elements can interact with humans [9]. This has required the development of specific and complex platforms based on hyperconnectivity [10] for the integration of these new technologies for the transfer of skills needed in the industrial sector [11],[?].

However, this environment only takes into consideration the interaction between humans and the rest of the assets or elements of the virtualized environment. This is a problem because the human being is isolated from others, preventing a natural interaction that allows not only the transfer of skills [12] or knowledge but also attitudes and values that are an essential part of education [13].

The creation of virtual human beings (such as avatars) thus plays an active role in the development of digital educational platforms. They can be beneficial not only for virtual laboratories, but for various fields of knowledge [14]. In this digitized environment, we not only have the advantages of digitization, but we favor integration and interaction between individuals allowing creativity and inspiration, key to development [15].

To achieve this proposal, the authors developed and implemented a new digital platform, called educational metaverse, in order to evaluate the impact on a population that has been the object of study, obtaining highly promising results.

Therefore, the main contributions to the scientific literature of the work are:

- Development of an educational digital platform where humans and elements of the virtual environment can interact.
- Evaluation of students' perception when using the educational metaverse platform.

The paper is organized as follows: Sect. 2 presents the case study, Sect. 3 deals with the technological development of the educational metaverse, Sect. 4 describes the methodology used to obtain the results, which are shown in Sect. 5, and Sect. 6 contains the conclusions.

2 Case Study

This work has been carried out in a mechanical engineering degree course at the University of León. The objective is the development of a practice using enabling technologies for the training of industrial operators. In this way the students get to know and test these environments in order to perform collaborative tasks in the industrial field.

To establish a cooperative environment and provide a more immersive and valuable educational experience, we need a challenge that involves interactions between humans and machines. This challenge involves collecting and placing 16 geometric prisms on a table as specified on a panel inside the workstation. Participants will have to move and bend down to pick up the objects from a shelf, simulating real-life conditions, as illustrated in Fig. 1.



Fig. 1. Real environment of experimentation where there are two workstations with shelves with colored geometric figures for the realization of the challenge.

However, for the challenge to be truly collaborative, users should exchange some objects according to each other's goals, fostering interaction within the virtual environment. Additionally, an autonomous vehicle has been incorporated to facilitate the transfer of objects between workstations. This enhances the interaction between humans and key machines in virtual laboratories.

3 Technological Implementation

The objective of the educational metaverse is to recreate not only the interaction with machines or equipment but also with the rest of the humans as realistically as possible. This requires the convergence of various technologies (e.g., DT, VR, Information and Communication Technologies (ICT)) through a digital platform in order to create a virtualized environment as can be seen in Fig. 2.

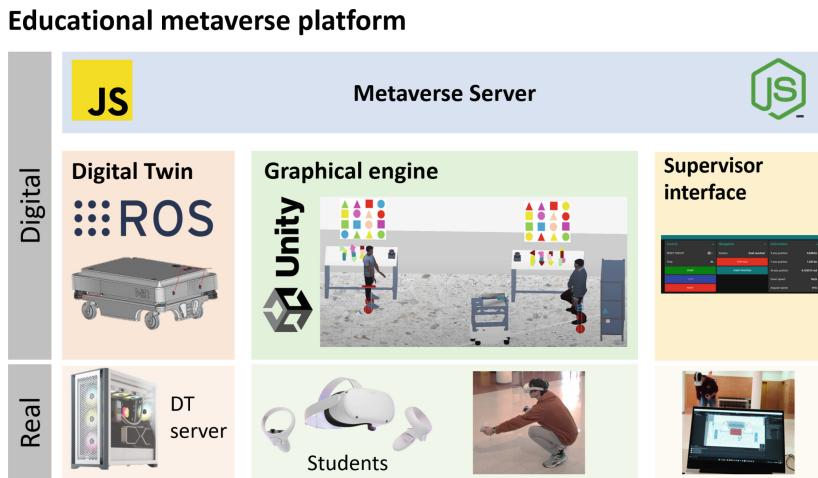


Fig. 2. Representative diagram of the technologies and communications used for the implementation of the educational metaverse.

The platform is implemented on a server based on Node.JS where JavaScript functions have been implemented for the communication of all assets. In addition, this server has a web interface where the platform can be monitored and controlled, as well as the experimentation environment.

DTs are used to recreate the operation of machines or assets within a virtual environment. The DTs enable the reliable simulation of elements' behavior according to their surrounding environment. This allows for simpler working hypotheses compared to real-world functioning. In our case study, an industrial autonomous vehicle has been simulated using the Robot Operating System (ROS) ecosystem.

This ecosystem allows to simulate both the sensors and the actuators of the vehicle being able to establish the same navigation algorithms that the real equipment has [16]. This approach creates a replication of behavior in response to the same stimuli (e.g., obstacles) as would occur in reality. In addition, communication has been established through the platform to receive the conditions of the environment, as well as to send the pose [17] (i.e., location and orientation) of the vehicle because ROS lacks a realistic interface to the environment.

For this reason, the Unity graphics engine has been used to digitally recreate all the elements of the experimental environment, including the vehicle. Additionally, to create a more human-like and immersive experience, people have been virtualized, called avatars in the metaverse world. However, the movement of the avatars' joints has been challenging since only the wrist and neck poses are available. To address this, direct kinematics algorithms have been implemented to establish the position of each link in the kinematic chain.

Thus, the avatar can interact with the machine, but not with other avatars, which prevents a more realistic training. In this sense, a series of scripts have been programmed in order to communicate with the platform server to know not only the state of the machines but also the interactions of humans in the virtual environment. In this way, when an avatar moves or modifies the environment, it notifies the server, which forwards this information to the rest of the participants.

Therefore, with this platform, students and teacher will be able to interact naturally in a virtual environment. Given the nascent state of technological implementations in the literature, it is important to understand the students' perceptions of these technologies. To achieve this, a clear methodology must be established.

4 Methodology

In order to obtain the results, an experimentation has been carried out with the students of the machine design course of the fourth year of mechanical engineering at the University of León. The subject has a total of 60 students who have formed the population under study, who have been divided into two groups of 30 students, giving rise to 15 pairs per group. One group did the metaverse activity first and the other group did the reality activity first. This population was randomly divided to assess collaborative task performance in an industrial setting where all individuals were then surveyed on their perceptions.

For this purpose, the same scenario of the real environment was recreated in the virtual one so that the students could assess the potential of the technologies. However, depending on the activity they performed first, they may be biased, which is why the authors decided to divide the pairs into two homogeneous groups. Despite this, all pairs independently of the group, performed the two activities (i.e., real experimentation and educational metaverse) in order to compare both of them.

For the evaluation, an anonymous survey was conducted digitally (i.e., by computer) to all individuals. The survey consisted of a series of mandatory questions that individuals had to answer using a 1 to 5 Likert scale or with preset responses.

Two Oculus Quest 2 virtual reality goggles were used to carry out the experimentation, which have two 1920×1832 pixel resolution screens with a refresh rate of 72 Hz, as well as WiFi connectivity. For the servers and the DT, we have opted for an Intel Core i9 processor and Ethernet connectivity in order to reduce response times and thus improve the experience of the students from which the following results have been obtained.

5 Results

Students' perceptions were collected by conducting a survey on their personal experience of the educational metaverse just after they had experienced both the real and the virtual experiences. The aim of the questionnaire was to determine the differences between the two worlds on subjective issues such as dimensions of space and ease of interaction with objects. For this purpose, the following questions were asked to the population:

- In general, indicate the degree of similarity between the experience in the real and virtual world.
- How intuitive have you found the interaction with the objects?

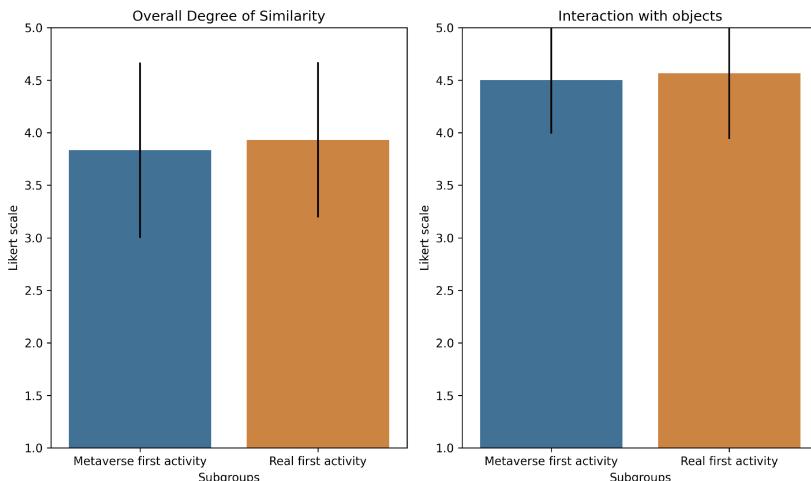


Fig. 3. Graphs representing on the Likert scale the similarity between the real and virtual world (left) and the ease of interaction with objects (right).

According to the graphs in Fig. 3, it can be seen that there is no significant evidence of bias according to which activity was performed first, obtaining very similar and positive results in both groups on the similarity of the environment. In addition, the ease with which students interact with objects has been analyzed, resulting in very intuitive for the majority of the population.

We have also analyzed the perception of space between the immersive experience of VR goggles and reality. In addition, we have addressed in a novel way whether students with VR glasses limit the speed of their movements due to the fear of colliding with objects or people in the real world. In order to obtain these results, we have asked the following questions:

- In terms of spatial perception of the environment

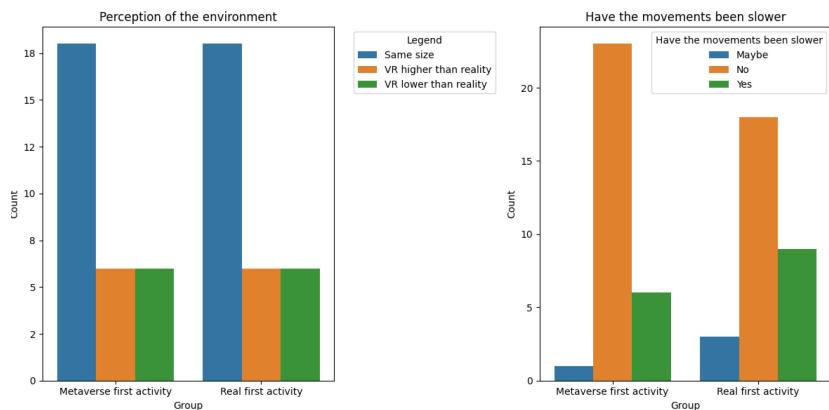


Fig. 4. Graphs depicting students' perception of real and virtual world dimensions (left) and limitation of movement when wearing VR glasses (right).

- Have you slowed down your body movements with virtual reality goggles for fear of hitting yourself in reality?

According to the results shown in the graphs in Fig. 4, most people, regardless of the group, indicate that the virtual world is the same size as the real world. However, 6 people per group think that the real world is larger and another 6 people who think the opposite, having designed the virtual environment on a real scale. However, for the group whose first activity was the metaverse, they indicated that the majority did not limit their movements, while those who did it first in the real world limited their movements more with the VR goggles. These observations highlight the complexity of perception and interaction in virtual environments, raising the need to further explore initial experiences and their impact on adaptation to virtual reality in educational and social contexts.

6 Conclusions

Education is a cornerstone for the progress of society. Digitalization in education offers flexibility and global access but has not yet solved human-to-human interaction in a natural way. In this context the educational metaverse represents a significant milestone by offering an immersive and collaborative environment that transforms the way we learn and teach. Faced with this challenge, the authors have developed a virtual space for the practice of an engineering subject where students have interacted between them and the machines. However, the implementation of this technology in the classroom has been a technological challenge because it has required the integration of numerous technologies such as DT, ICT or VR. To integrate these technologies a digital platform has been developed where these technologies converge to have an immersive and collaborative educational environment. To demonstrate these qualities, a survey has

been conducted to students to know their perception before these virtual environments where very promising results have been obtained to continue working in the educational metaverse as an integral space for academic and human development.

This work was partially supported by the Spanish Research Agency (AEI) under grant number PID2023-153047OB100 and The teaching innovation group MECACOM of the Universidad de León.

References

1. Woodfield, P.J., Husted, K.: How does knowledge sharing across generations impact innovation? *Int. J. Innov. Manag.* **23**(08), 1940004 (2019)
2. Nada, C.I., Legutko, J.: “Maybe we did not learn that much academically, but we learn more from experience”—erasmus mobility and its potential for transformative learning. *Int. J. Intercult. Relat.* **87**, 183–192 (2022)
3. Yarychev, N., Mentsiev, A.: Impact of digital education on traditional education. *J. Phys. Conf. Ser.* **1691**, 012132 (2020)
4. Schmidt, J.T., Tang, M.: Digitalization in education: challenges, trends and transformative potential. *Führen und Managen in der digitalen Transformation: Trends, Best Practices und Herausforderungen*, pp. 287–312 (2020)
5. Davronovich, A.D., Mansurjonovich, J.M.: Important advantages of organizing the educational process in a digital technology environment. *Galaxy Int. Interdisc. Res. J.* **11**(2), 149–154 (2023)
6. Potkonjak, V., et al.: Virtual laboratories for education in science, technology, and engineering: a review. *Comput. Educ.* **95**, 309–327 (2016)
7. Checa, D., Gatto, C., Cisternino, D., De Paolis, L.T., Bustillo, A.: A framework for educational and training immersive virtual reality experiences. In: De Paolis, L.T., Bourdot, P. (eds.) *AVR 2020. LNCS*, vol. 12243, pp. 220–228. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58468-9_17
8. Martínez-Gutiérrez, A., Díez-González, J., Ferrero-Guillén, R., Verde, P., Álvarez, R., Perez, H.: Digital twin for automatic transportation in industry 4.0. *Sensors* **21**(10), 3344 (2021)
9. Martínez-Gutiérrez, A., Díez-González, J., Verde, P., Perez, H.: Convergence of virtual reality and digital twin technologies to enhance digital operators’ training in industry 4.0. *Int. J. Hum. Comput. Stud.* **180**, 103136 (2023)
10. Martínez-Gutiérrez, A., Díez-González, J., Verde, P., Ferrero-Guillén, R., Perez, H.: Hyperconnectivity proposal for smart manufacturing. *IEEE Access* (2023)
11. Verde, P., Díez-González, J., Álvarez, R., Perez, H.: Characterization of AGV localization system in industrial scenarios using UWB technology. *IEEE Trans. Instrum. Meas.* **72**, 1–13 (2023)
12. Díez-González, J., et al.: Evaluation of the skills’ transfer through digital teaching methodologies. In: García Bringas, P., et al. (eds.) *CISIS ICEUTE 2023*, pp. 340–349. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-42519-6_32
13. Summers, R., Higson, H., Moores, E.: The impact of disadvantage on higher education engagement during different delivery modes: a pre-versus peri-pandemic comparison of learning analytics data. *Assess. Eval. High. Educ.* **48**(1), 56–66 (2023)
14. Zhang, R., Wu, Q.: Impact of using virtual avatars in educational videos on user experience. *Sci. Rep.* **14**(1), 6592 (2024)

15. Hayes, S., Jopling, M., Connor, S., Johnson, M.: Human Data Interaction, Disadvantage and Skills in the Community: Enabling Cross-Sector Environments for Postdigital Inclusion. Springer, Cham (2023). <https://doi.org/10.1007/978-3-031-31875-7>
16. Martínez-Gutiérrez, A., Díez-González, J., Verde, P., Ferrero-Guillén, R., Perez, H.: Analysis of navigation algorithms for a fleet of mobile robots by means of digital twins. In: Vizán Idoipe, A., García Prada, J.C. (eds.) IACME 2022, pp. 412–417. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-38563-6_60
17. Sánchez-Calleja, I., Martínez-Gutiérrez, A., Ferrero-Guillén, R., Díez-González, J., Perez, H.: Contact system method for the precise interaction between cobots and mobile robots in smart manufacturing. Int. J. Precis. Eng. Manuf. 1–16 (2023)



Cybersecurity and Cyber Defense in the Classroom

Jorge Barbosa^(✉)

Polytechnic University of Coimbra, ISEC/IPC, Coimbra, Portugal
jorge.barbosa@isec.pt

Abstract. The inclusion of soft skills linked to cybersecurity and cyber defense in the academic curricula of higher education courses is a very current issue. Several studies have been carried out in this regard given the increasing importance of creating awareness about good practices in accessing computer networks, particularly the Internet.

However, the conclusions of these studies are generally limited to determining a set of recommendations on good practices to build the academic curriculum of the course.

The approach we take here is from the perspective of the students who may be the target of these cybersecurity skills. In this sense, we are developing a study in several Portuguese higher education institutions, to understand what higher education students think about this issue and their perception of the need to have training in the area, where we will question students from various areas of education on this topic.

We will then present the preliminary results of a pilot test study where we questioned 423 students from the Informatics Engineering Degree course.

Keywords: Cybersecurity · Cybersecurity Curricula · Higher Education · Soft Skills

1 Introduction

1.1 The Current Reality

Due to the exponential increase in the use of ICT in all economic areas and even in all homes, as well as the diversification and sophistication of computer equipment and the large number of applications to be used in this equipment, it creates constant pressure on professionals who work in ICT. And the need for this type of professionals has also grown substantially in recent years.

In particular, a sub-sector of these professionals, those specialized in cybersecurity, presents a shortage in the number of these professionals available, aggravated by the constant evolution of cyber-attack techniques, which requires these professionals to have extensive and constant training to be up-to-date and up to date of these new techniques so that they can protect the organizations to which they belong. ISACA's annual State of Cybersecurity Report (ISACA) identified several reasons for this shortage, including the complexity and breadth of the skill set that these professionals must master.

1.2 Cybersecurity and Distance Learning Technologies

The recent COVID-19 pandemic brought the use of distance learning technologies to the present day. Suddenly students, at all levels of education, as well as teachers found themselves faced with the need to use these technologies and, preferably, in a safe way. However, what came to be verified is that most students and teachers, especially those associated with more basic levels of education and in less technical areas, did not have the technical knowledge, even if basic, of how to safely use technological means. At the same time, the use of technological resources associated with the teaching-learning process was carried out from each one where the security of home networks was non-existent or very basic.

If there was already a perception of the need to train the general population and in particular younger people in cybersecurity, with the pandemic this then became an absolute necessity and in the current post-pandemic period, initiatives in this regard have multiplied, especially because the use of remote means of work, called teleworking, became commonplace and came to be at the mercy of the experience and habits acquired at that time.

1.3 How to Do This Training

Many publications have been made on the issue of including cybersecurity training in school curricula, particularly higher education courses, and how to do so. In (Ramezanian and Niemi 2023), we have an excellent guide on how to build a curriculum including cybersecurity. Also (Ricci and Janout 2021), describe from a PESTLE perspective how to introduce and teach cybersecurity in curricula. As it is based on the PESTLE analysis, an analysis studies the key external factors (Political, Economic, Sociological, Technological, Legal and Environmental) that influence the application scenario, in this case of cybersecurity training, it is an excellent guide for decision-makers and professionals of education in making strategic decisions involved in the construction of higher education course curricula. There are many other excellent references for building a cybersecurity curriculum, (Swire 2018), (Fischer 2022) (Daimi and Francia III 2020), (Buckley & Zalewski 2019), (Sikos and Haskell-Dowland 2023) and (Ramezanian and Niemi 2023).

1.4 Fundamentals for the Current Study

In this paper, we will present the preliminary results of a pilot study carried out with an intentional and representative sample of higher education students, namely undergraduate students in computer engineering at a higher education institution.

The choice of these degree courses in this pilot project is deliberate because considering the context and profile of the students to whom the survey questionnaire was applied, we consider that these courses would be appropriate to assess and improve the project itself at this stage.

In the following part, we will present the results of the treatment of the survey that we carried out and that corresponds to the first phase of the project that we will later implement.

2 Survey Developed

2.1 Application Environment

The survey was distributed to approximately 490 students from three different undergraduate courses in Informatics Engineering, LEI. These courses were the Bachelor's Degree in Informatics Engineering running during daytime hours, the course in Informatics Engineering running after working hours, and the course in Informatics Engineering – European Course, European Computer Science Engineering Course, a transnational course and multi-grade. In addition to regular students, some Erasmus students were also present. Finally, to carry out the survey in a more restricted and responsible environment, the surveys were distributed during written exam tests in which students were present. Three curricular units were selected, one from each academic year, and students were invited to fill out the surveys, with these responses being anonymous and optional.

For a better characterization of the sample and to consider correlations between the types of students and their responses, in addition to indicating in the survey the course in which they were enrolled, they were also asked to indicate their gender so that it could be analyzed whether there was some tendency in the answers depending on the gender of the respondents.

Likewise, and considering that any of these three courses have a curricular plan in accordance with Bologna, that is, each course has a curricular plan spread over three academic years, they were asked to indicate how many enrollments they had already completed on the course. This distinction between the number of enrollments is due to the fact that we want to correlate the time in Higher Education with possible experience and contact with situations of exposure or knowledge of cyber-attack actions and their possible predisposition and knowledge of how to face them and treat them.

2.2 Survey Structure

To gather the desired information, we built a survey composed of two groups of questions. In the first group, we intend to characterize the sample and in the second group, we intend to evaluate the students' perception of the concept of cybersecurity.

It should be noted that the second group presents simple and compound questions. The Likert scale was used to classify the compound questions with values from «1» (which means less concern about the issue) to «5» (which reflects greater concern about the issue).

2.3 Data Correlation

Through this set of questions, we will correlate the data provided in order to obtain information and extract knowledge considering different combinations of answers given depending on the type of respondents to the survey, such as gender, course, number of enrollments, whether or not you have already had training in the area and other considerations.

From this pilot test presented here, some lessons were drawn regarding the correlations that could and should be considered. Therefore, in the next phases of the study,

this aspect of correlations will be improved. The objective will be to obtain, based on other combinations of data, the possibility of considering more correlations to obtain more information and extract other knowledge.

In this previous analysis of results carried out here, we will not analyze the correlations, although these can already be done and will be presented in future articles.

2.4 Results Analysis

- **Answer Chart for Question 1:** “Understand the concepts of Cybersecurity and Cyber defense?”

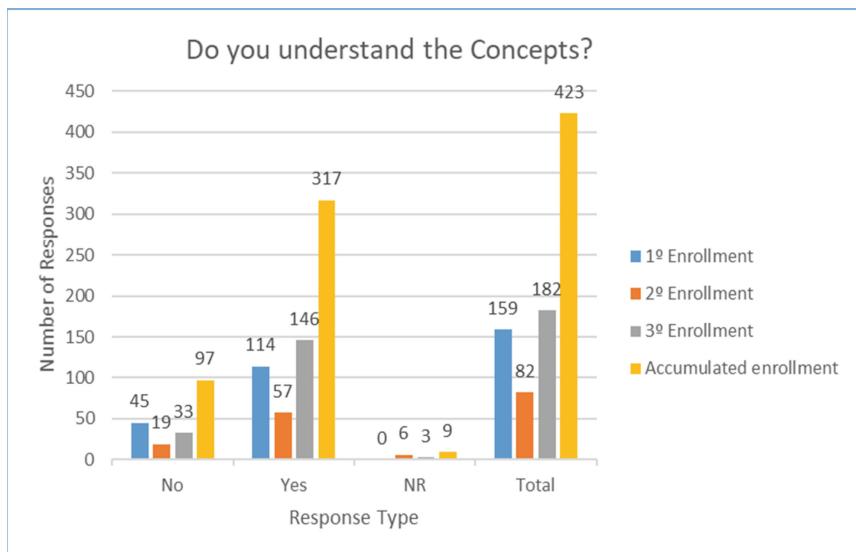


Fig. 1. Understand the concepts of Cybersecurity and Cyber defense?

- **Answer Chart for Question 2:** “Have you had any training (any) in this area?”
- **Answer Chart for Question 3:** “Do you believe that this subject should be part of the curricula of Higher Education courses?”
- **Answer Chart for Question 4:** “Have you ever been asked for help/advice on these matters that you knew how to respond to?”
- **Answers Charts Question 5 (Composite Question):** “As an IT user, do you feel cyber threatened in this use?”
 - Answer Chart 1 – Base Question
 - **Answer Chart 2 – Likert Scale-Based Graduation**

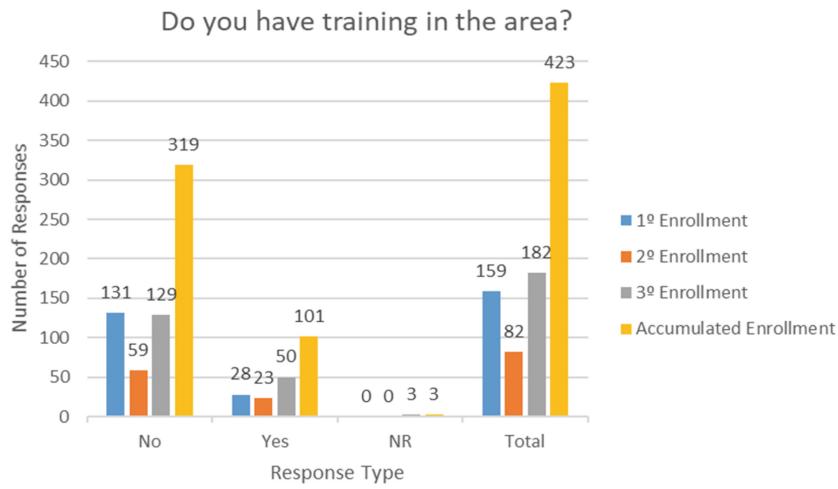


Fig. 2. Have you had any training (any) in this area?"

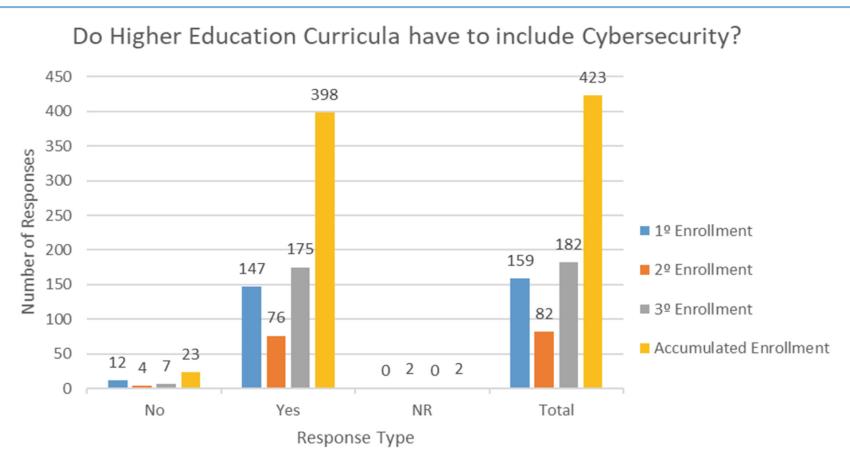


Fig. 3. Do you believe that this subject should be part of the curricula of Higher Education courses?

2.5 Responses and Graph Analysis

- **Question 1:**

From the analysis of this Graph1, Fig. 1, as would be expected given the profile of the responding students, students of a Informatics Engineering course, a large majority of respondents indicate that they know the concepts of Cybersecurity and Cyber defense.

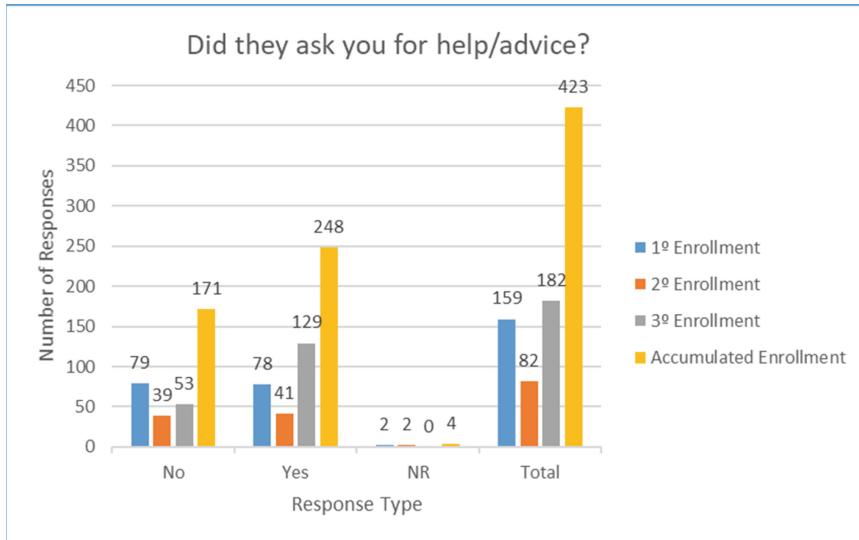


Fig. 4. Have you ever been asked for help/advice on these matters that you knew how to respond to?

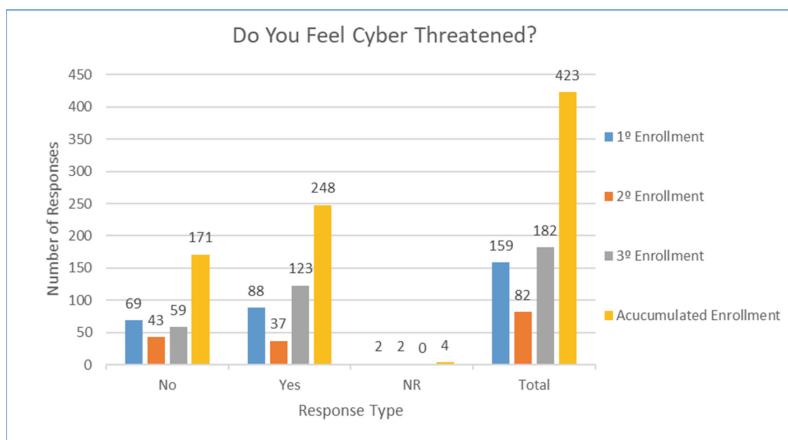


Fig. 5. As an IT user, do you feel cyber threatened in this use? – Base Question

As would also be expected, the predominance of “No” answers are given by students with only one enrollment to date, although the number of students with more enrollments is still relatively large, especially with three or more enrollments for students with this training profile.

- **Question 2:**

The analyses of the second graph reveal that despite the profile of these students, the number of students who respond that they have not attended any training in the area of

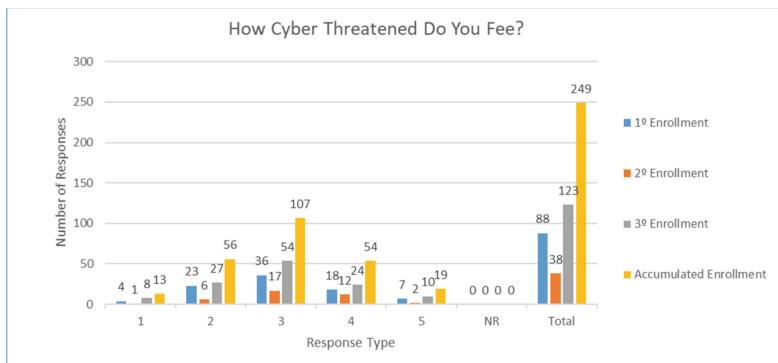


Fig. 6. As an IT user, do you feel cyber threatened in this use? – Likert Scale Based Graduation

cybersecurity is very large, even among students with three or more enrollments, Fig. 2. This number is practically the same when comparing students with fewer enrollments and those with more enrollments who answered “No”.

Considering the positive responses, the numbers relating to the three groups of enrollment numbers are already close to what seems to be expected for this question as the students with the highest number of enrollments are those who respond positively in greater numbers. However, these numbers are still overall and partly very low.

From a global point of view, only approximately a quarter of students respond that they have already had training, which we consider a very low value for students in this area.

• Question 3:

This third question, whose results are in the Graph shown in Fig. 3, is not only about the possible need for training in the area of cybersecurity but is posed as whether training in the area is necessary for all higher education courses.

This question is the most consensual of all those considered and is in line with what was predicted and also with the results obtained in the previous question. In a way, we also think that it can be considered that students themselves not have much training in the area and probably the missing of this training reflects this fact in the high number of positive responses to this question.

Also, the fact that they are students in this area of training probably makes them more aware of this issue and this will also have been reflected in the high number of positive responses that were obtained.

The number of negative responses is residual, representing in global terms only 0.5% of the responses and there are also very few students who did not answer this question.

• Question 4:

In this question, in which we purposely left many aspects open, the distribution between negative and positive answers is approximately one negative to one and a half positive.

It is expected that students in this area will be in high demand by friends and family to solve problems with computer hardware and software, so it is not surprising that they will also be in high demand on issues related to cybersecurity.

It is also worth noting that practically the same numbers of negative and positive responses were obtained in the case of students with one registration, with the same trend being observed in students with two registrations. Practically half of the students in these two subgroups, considering the number of enrollments, answered “No” and the other half answered “Yes”, Fig. 4.

In the case of students with three or more enrollments, almost two-thirds responded affirmatively and only almost one-third responded negatively. Considering this profile of students and their “experience” in higher education in the area, these data are coherent.

- **Question 5 (Composite question):**

This question is one of three composite questions that were asked to students in the survey. In these questions, the student answers an initial basic question, Fig. 5, and if they have answered “Yes” to it, they will grade their answer according to a Likert scale where they indicate an integer value from 1 to 5, with “1” being the value that reflects less concern about the issue and “5” reflects greater concern about the issue.

In the specific case of this question, the data reveals that there are many students, around 58.6%, concerned about cybersecurity issues as they answer “Yes”, that they feel threatened.

There is also a significant number of students, around 40.4%, who appear unconcerned, as they answer “No” they feel afraid. This relatively large number of negative responses is worrying and should be analyzed with more care and detail to try to understand why there are so many negative responses given the current context regarding cybersecurity. In part, and only based on the data obtained, it can be considered that this high number of “No” responses is due to the relatively large number of students with a single enrollment who responded “No”, around 43.4% of these students. Perhaps their relative, incipient, lack of knowledge of this subject leads them to respond negatively, although, however, there is also a large number of students with a single enrollment who answered “Yes”, around 55.3%.

The data relating to students with three or more enrollments are as expected, as although there is still a significant number of students from this sub-profile, around 32.4%, who responded “No”, the remaining 67.5% responded “Yes” which means they feel cyber threatened.

Regarding the grading of “fear” that the students who answered “Yes” did, the graph is in line with what seemed to be expected. The distribution presented in the graph is like a pure, Gaussian Normal Distribution, Fig. 6, as the largest number of occurrences is in the average, central value, “3”, with the distribution being practically equal in the lateral values corresponding to the left and right of the value. Central and decreasing towards the extremes of the graph with very similar values between the occurrences on the left and right.

The average value “3” represents approximately 43% of the total number of responses, which indicates that there is a significant number of students who feel moderately cyber threatened.

The maximum number on the scale, “5”, was indicated by a still significant number of students, around 7.6%, which means that there are students who consider themselves very “cyber threatened”, with this value being slightly higher than the referring to students who do not feel threatened at all, around 5.2%.

3 Conclusion

A first analysis of the data obtained in this pilot project allows us to conclude that there is a great awareness among students regarding issues related to cyber security.

In addition to this first conclusion, we can also say that the vast majority of these students are concerned about the causes and consequences of cyber-attacks and the lack of knowledge that most people have about this problem.

The combination of these first two conclusions leads us to consider as a third conclusion, perhaps the result of the two previous conclusions and their implications, the fact that practically all of these students, around 94%, consider it essential that all people, in particular, students from any area of higher education must have training in cybersecurity!

However, these conclusions may be biased by the fact that all 423 students who responded to this survey are ICT students, in particular informatics engineering students. Hence the need that we feel and that we will implement in later stages to extend this study to students from other areas of higher education in order to draw more general conclusions as they are obtained with less dependence on a certain profile of students.

As mentioned, also that we understand that it is necessary to train more professionals in the area and, without neglecting this need, to train all students and teachers at all levels of education beyond higher education even if it's just in a more basic way but no less effective and necessary way.

These results are preliminary results of a pilot study applied to a very specific and purposefully chosen set of students, computer engineering students, with the aim of measuring the results and testing the questions asked.

As future work and based on this previous study analyzed here, we will extend it to a much larger number of students from other courses and other higher education institutions in order to not only increase the sample size but essentially to dilute the effect that certain type of students, namely ICT students, may have in the study and thus bias it.

References

- Buckley, I.A., Zalewski, J.: Course development in the cybersecurity curriculum. In: 17th LACCEI International Multi-Conference for Engineering, Education, and Technology (2019)
- Daimi, K., Francia III, G.: Innovations in Cybersecurity Education (2020)
- Fischer, P.J.: A cybersecurity skills framework. In: Research Anthology on Advancements in Cybersecurity Education (2022). <https://doi.org/10.4018/978-1-6684-3554-0.ch010>
- ISACA. State of Cybersecurity 2023 - Annual ISACA® Global State of Cybersecurity. ISACA (2023)
- Ramezalian, S., Niemi, V.: Cybersecurity education in universities: a comprehensive guide to curriculum development, vol. 11 (2023)

- Ricci, S., Janout, V.: PESTLE analysis of cybersecurity education. In: The 16th International Conference on Availability, Reliability and Security (ARES 2021), Vienna, Austria (2021). <https://doi.org/10.1145/3465481.3469184>
- Ruiz, R.: A study of the UK undergraduate computer science curriculum: a vision of cybersecurity. In: 12th International Conference on Global Security, Safety & Sustainability, London (2019)
- Sikos, L.F., Haskell-Dowland, P.: Cybersecurity teaching in higher education (2023). <https://doi.org/10.1007/978-3-031-24216-8>
- Swire, P.: A pedagogic cybersecurity framework. Commun. ACM **61**(10) (2018). <https://doi.org/10.1145/3267354>

ICEUTE: Special Session: Innovation in Computer Science Higher Education



Optimization of Transport Routes Through a Social Interaction Algorithm-Based Application

Diego Ismael Valdivia Alcalá¹ , Ángel Miguel García Vico¹ ,
and Cristóbal José Carmona del Jesús^{1,2}

¹ Andalusian Research Institute in Data Science and Computational Intelligence
(DaSCI), University of Jaén, 23071 Jaén, Spain
diva0001@red.ujaen.es

² Leicester School of Pharmacy, De Montfort University, Leicester 9BH, UK

Abstract. In this paper, we present a prototype application designed to solve the Traveling Salesman Problem (TSP). This application allows for obtaining high-quality solutions within reasonable time frames using bio-inspired algorithms, specifically variants of Ant Systems such as Ant Colony System, Max-Min Ant System and Best-Worst Ant System. We will review the operation of each of these algorithms and apply them interactively to solve TSP instances. These algorithms will be compared with the deterministic Lin-Kernighan algorithm to demonstrate their effectiveness.

Keywords: metaheuristics · travelling salesman problem · ant colony

1 Introduction

The Traveling Salesman Problem (TSP) [1–4] presents the critical challenge of determining the shortest route that visits each city exactly once before returning to the starting point. This problem is crucial in fields such as logistics and route planning, where optimizing travel routes directly impacts operational efficiency and costs.

However, the combinatorial complexity of the TSP escalates dramatically as the number of cities increases, with the number of potential routes growing factorially. This renders exact methods [5] like brute force impractical for large instances due to their exponential runtime.

Consequently, research has shifted towards alternative methods, particularly metaheuristics [6, 7], which offer efficient ways to find approximate solutions within reasonable time frames. Constructive metaheuristics, for instance, build solutions incrementally using heuristic rules to guide city selection and route construction. This approach not only enhances computational efficiency but also yields solutions of acceptable quality for many practical applications.

Beyond computational complexity, the TSP exhibits sensitivity to small variations in input data, leading to significantly different solutions. This sensitivity underscores the importance of methods that are not only time-efficient but also adaptable and robust under varying problem conditions.

The use of these metaheuristics, while yielding good results, is not accessible enough for users who are not experts in the field, such as students studying metaheuristics. Therefore, this work introduces a software application that enables students to easily utilize these algorithms, helping them gain a better understanding of the concepts and operation involved. Additionally, the application facilitates their learning about parameter tuning and its impact on the solutions obtained, allowing them to compare different algorithms and generate reports based on the results achieved.

This paper is organized as follows: Sect. 2 presents the background of our work, Sect. 3 shows our proposed software solutions, Sect. 4 discusses the experimental study, and finally, Sect. 5 presents the conclusions.

2 Background

Traditional methods for tackling the TSP include exact algorithms such as brute force [8] and backtracking [9], which meticulously explore all possible route combinations to find the optimal solution. However, their efficiency declines significantly with larger problem sizes due to the inherent computational complexity of the TSP.

In contrast, classic heuristics like the nearest neighbor algorithm [6] aim to efficiently find approximate solutions by employing simplified decision-making strategies. These approaches often yield suboptimal results and do not guarantee global optimization, limiting their effectiveness in diverse scenarios.

Dynamic programming [10] offers another approach to solving the TSP by reducing the number of partial solutions evaluated, although it can be computationally intensive in scenarios with numerous cities.

Deterministic algorithms, characterized by their adherence to fixed rules and predictable outcomes for a given input, face similar challenges. While capable of delivering consistent solutions, they struggle with the intricate nature of the problem and the inability to adapt to variations in dataset characteristics. Noteworthy deterministic algorithms used for TSP include Lin-Kernighan [11], Chained Lin-Kernighan and the Christofides algorithm.

The evolution of approaches to solving the TSP has seen traditional methods aiming for optimal or approximate solutions. However, their inefficiencies, particularly with larger and more complex problem instances, underscore the need for advanced and efficient methodologies. As a result, modern techniques such as Ant System [3] and its variants like Ant Colony System [12], Max-Min Ant System [13] and Best-Worst Ant System [14] have emerged. These approaches have demonstrated superior effectiveness leading to significant improvements in solution quality and computational efficiency.

Currently, there are systems for solving the TSP, such as Concorde TSP Solver¹. However, it relies on classical algorithms like integer linear programming (ILP) [15] and specific heuristics like LKH [11], which may encounter efficiency challenges with a high number of cities. Additionally, it can be complex for inexperienced users to grasp. These limitations will be addressed in the proposed application outlined in this document.

3 Proposed Solution

To solve the TSP, we will develop an application that facilitates interactive exploration of solutions for a symmetric TSP, where travel distances between cities remain consistent in both directions.

We will begin by exploring bio-inspired algorithms, starting with an explanation of Ant System, which forms the basis for subsequent bio-inspired algorithms. Afterward, we will delve into the deterministic Lin-Kernighan algorithm. Concluding this section, we will highlight the user interface that integrates these algorithms.

3.1 Ant System

This Ant System algorithm (AS) [3] starts by generating an initial solution in a greedy manner and evaluates its cost. Then, the initial amount of pheromone is calculated as:

$$\tau_0 = \frac{1}{C(S) \cdot m} \quad (1)$$

where $C(S)$ is the cost of the initial solution and m is the number of ants.

Subsequently, all ants start from the same city, which is selected randomly, and choose the subsequent cities based on a transition rule that calculates probabilities.

$$p_k(r, s) = \begin{cases} \frac{[\tau_{rs}]^\alpha \cdot [\eta_{rs}]^\beta}{\sum_{u \in J_k(r)} [\tau_{ru}]^\alpha \cdot [\eta_{ru}]^\beta}, & \text{if } s \in J_k(r) \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

After constructing paths for each ant, we evaluate and retain the solution with the lowest cost. If this solution improves upon the best one found so far, we update our records.

Subsequently, we update the pheromone matrix. Each ant deposits pheromone on the arcs it has traversed, based on the cost C of its solution S . This contribution is computed as follows:

$$\frac{1}{C(S)} \quad (3)$$

This process allows ants in the next iteration to be more likely to visit the most promising arcs, those with higher pheromone levels.

¹ <https://www.math.uwaterloo.ca/tsp/concorde.html>.

Additionally, pheromone evaporation takes place, decreasing its value on less visited arcs (less promising ones). This final step is summarized in Eq. 4.

$$\tau_{rs} = (1 - \rho) \cdot \tau_{rs}(t - 1) + \rho \cdot \sum_{k=1}^m \Delta\tau_{rs}^k \quad (4)$$

This process is repeated until the termination condition is met.

3.2 Ant Colony System

The Ant Colony System (ACS) [12] introduces three significant modifications in AS:

- Modification of the transition rule: Adjusted to balance exploration and exploitation within the solution search space. A random value between 0 and 1 is generated; if this value is greater than or equal to the parameter q_0 , the most promising arc is selected based on an improved transition rule. Otherwise, the AS transition rule is applied.

$$p'_k(r, s) = \begin{cases} \frac{[\tau_{rs}]^\alpha \cdot [\eta_{rs}]^\beta}{\sum_{u \in J_k(r)} [\tau_{ru}]^\alpha \cdot [\eta_{ru}]^\beta}, & \text{if } s \in J_k(r) \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

$$p_k(r, s) = \begin{cases} \arg \max_{u \in J_k(r)} \{ [\tau_{ru}]^\alpha \cdot [\eta_{ru}]^\beta \}, & \text{if } q_0 \leq q \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

- Global pheromone update (offline): Evaporation of pheromones occurs on all arcs, and only the best ant contributes to the pheromone update. This promotes the selection of more promising arcs.

$$\tau_{rs} = (1 - \rho) \cdot \tau_{rs}(t - 1) + \rho \cdot \Delta\tau_{rs}^{best} \quad (7)$$

- Local pheromone update (online): Pheromone evaporates each time all ants choose a new arc, encouraging exploration of less-visited arcs.

$$\tau_{rs} = (1 - \phi) \cdot \tau_{rs}(t - 1) + \phi \cdot \tau_0 \quad (8)$$

3.3 Max-Min Ant System

The Max-Min Ant System (MMAS) [13] is characterized by greater exploitation of the best solutions and includes an additional mechanism to prevent search stagnation.

It retains the transition rule of the AS and modifies the following:

- The pheromone update mechanism is more aggressive, evaporating all traces and depositing pheromone only on the best solution traces.

$$\tau_{rs}(t) = (1 - \rho) \cdot \tau_{rs}(t - 1) + \rho \cdot \Delta\tau_{rs}^{best} \quad (9)$$

- It defines minimum and maximum limits for the pheromone trails.

$$\tau_{max} = \frac{1}{1 - \rho} \cdot \frac{1}{f(s^{opt})} \quad (10)$$

$$\tau_{min} = \frac{\tau_{max} \cdot (1 - \sqrt[n]{p_{best}})}{(avg - 1) \cdot \sqrt[n]{p_{best}}} \quad (11)$$

These values are updated upon discovering a solution better than the current best found. Each edge's pheromone update ensures it remains within the range $\tau_{min} \leq \tau_{rs}(t) \leq \tau_{max}$. This strategy promotes exploration across the search space by preventing some edges from becoming rarely selected (near zero pheromone) and others from becoming excessively favored (approaching infinite pheromone).

- It restarts the search when it stagnates. This checks if, after a specified number of iterations, no better solution than the current best has been found. If so, all edge pheromones are reset to the maximum value (τ_{max}) to initiate a new search phase.
- Pheromone smoothing. This involves checking algorithm convergence based on whether the pheromone levels on edges in the best solution approach the maximum value (τ_{max}). If so, all edge pheromones are adjusted proportionally to the difference, boosting levels on less-visited paths while minimally affecting well-explored ones.

$$\tau_{rs}(t) = \tau_{rs}(t) + \delta \cdot (\tau_{max}(t) - \tau_{rs}(t)) \quad (12)$$

3.4 Best-Worst Ant System

The Best-Worst Ant System (BWAS) [14] is another extension of the AS that integrates components from Evolutionary Computing to enhance the exploration-exploitation balance.

It retains the transition rule of the AS and introduces the following changes:

- The update mechanism is more exploitative by evaporating all traces, positively reinforcing only those of the best global solution, and negatively reinforcing those of the current worst solution.

$$\tau_{rs}(t) = (1 - \rho) \cdot \tau_{rs}(t - 1) + \Delta\tau_{rs}^{best} \quad (13)$$

$$\tau_{rs}(t) = (1 - \rho) \cdot \tau_{rs}(t), \forall a_{rs} \in S_{current_worst} \text{ and } a_{rs} \notin S_{global_best} \quad (14)$$

This approach promotes exploitation of the best solutions by avoiding traces from the worst arcs found in each iteration.

- It restarts when it stagnates.
- Pheromone smoothing.

$$\tau_{rs}(t) = \tau_{rs}(t) + \delta \cdot (\tau_{max}(t) - \tau_{rs}(t)) \quad (15)$$

- It applies pheromone trail mutation for exploration.

$$\tau'_{rs}(t) = \tau_{rs}(t) + N(0, \tau_{threshold}) \quad (16)$$

$$\tau_{threshold} = \frac{C(S_{global_best})}{n} \quad (17)$$

The mutation function is characterized by:

- o The strength of mutation increases with iterations: initially, $\tau_{threshold}$ is close to τ_0 and the mutation is small. As the trails of S_{global_best} grow, $\tau_{threshold}$ becomes larger.
- o Upon reset, it returns to its initial range.

3.5 Lin-Kernighan Algorithm

The Lin-Kernighan algorithm (LK) [11] is a sophisticated approach to combinatorial optimization designed to solve the TSP and similar problems requiring optimal route solutions. It improves upon the λ -opt concept.

The λ -opt algorithm posits that a route is λ -optimal if no improvement in length is possible by replacing any set of λ links with others. This suggests that a λ -optimal route may also be optimal for smaller λ values, offering an efficient approach to exploring different route configurations under specific conditions.

Lin-Kernighan adapts this principle by dynamically adjusting the value of λ during algorithm execution. This flexibility is crucial as it allows for optimal link exchanges to vary based on the problem's structure, complexity, and the quality of the current solution.

The Lin-Kernighan algorithm's core involves r -opt moves, where r represents the number of links considered for simultaneous exchange. Each iteration identifies two sets of links, X and Y, whose exchange leads to an improved route length. These moves are guided by the following criteria:

1. **Sequential exchange criterion:** Links $x_i = (t_{2i-1}, t_{2i})$ and $y_i = (t_{2i}, t_{2i+1})$ are selected such that x_i and y_i share a node, and y_i and x_{i+1} also share a node. This ensures that the sequence $(x_1, y_1, x_2, y_2, \dots, x_r)$ forms a chain of adjacent links, with $y_r = (t_{2r}, t_1)$ to close the chain.
2. **Feasibility criterion:** x_i connects t_{2i} to t_1 , enabling a valid closed route starting from $i = 3$, thus reducing execution time and simplifying coding.
3. **Positive gain criterion:** y_i is chosen so that the gain G_i of the proposed exchanges is positive, where G_i sums individual gains g_1, g_2, \dots, g_i .
4. **Disjointness criterion:** Sets X and Y must be disjoint to simplify coding, reduce execution time, and provide an effective stopping criterion.

3.6 Graphical User Interface

To facilitate the use of the implemented and previously described models, we have developed the following user interface. Next, we will present its two main screens to explain how it operates in solving the TSP.

The first screen (Fig. 1) of the application offers several key functionalities:

- It allows users to import datasets containing any number of cities, provided they adhere to the TSPLIB format². If the format is not correct, the system identifies it as invalid and prohibits the execution of any algorithms until the issue is resolved.
- It enables the loading of previously generated solutions for visualization purposes. However, it strictly permits the loading of PDF files that were generated by the system itself. This precaution helps prevent potential errors that could arise from incompatible file formats or data inconsistencies.
- Users can select and customize algorithm parameters directly within this interface. The system meticulously verifies the accuracy of each parameter value entered by the user. If any parameter value is found to be invalid, the system promptly notifies the user with an error message. This validation process ensures that only correctly configured parameters are used, thereby safeguarding the execution of the selected algorithm from potential errors or inaccuracies.

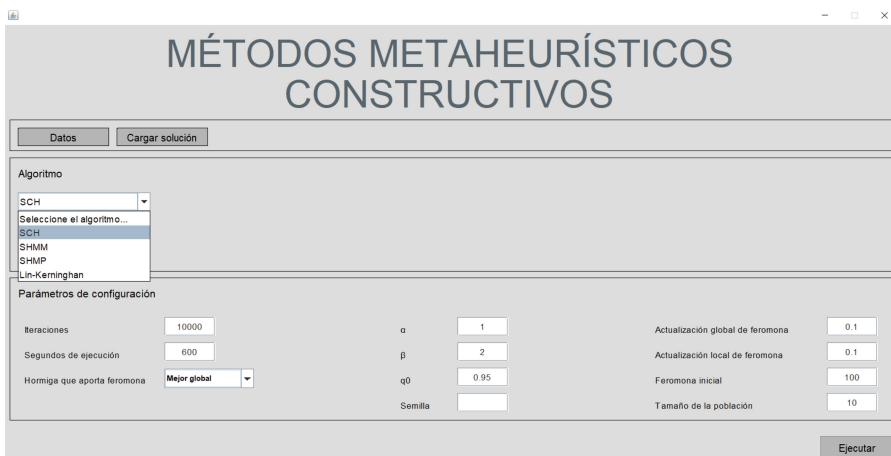


Fig. 1. Initial screen of the proposed user interface.

Once the user initiates execution, the system consolidates all the provided information and transitions to the second screen (Fig. 2). Here, users are presented with several key functionalities:

- Users can observe the algorithm currently in operation, providing insight into its progress and status. They can also visualize the dataset being utilized, allowing for a clearer understanding of the geographical layout or distribution of cities.

² <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>.

- The screen displays the best solution found thus far, presented both numerically and graphically. In the graphical representation, the starting city is highlighted in green, with other cities depicted in blue. This visual differentiation aids in easily identifying the path and structure of the current best solution.
- Users can view the cost of the best solution found in kilometers, offering a quantitative measure of its efficiency or optimality.
- The interface allows users to halt the execution of the algorithm at any point if necessary, providing flexibility and control over the process.
- Users have the option to save the results obtained during the execution as a PDF file. This feature ensures that the findings and visual representations can be preserved for future reference or sharing purposes.

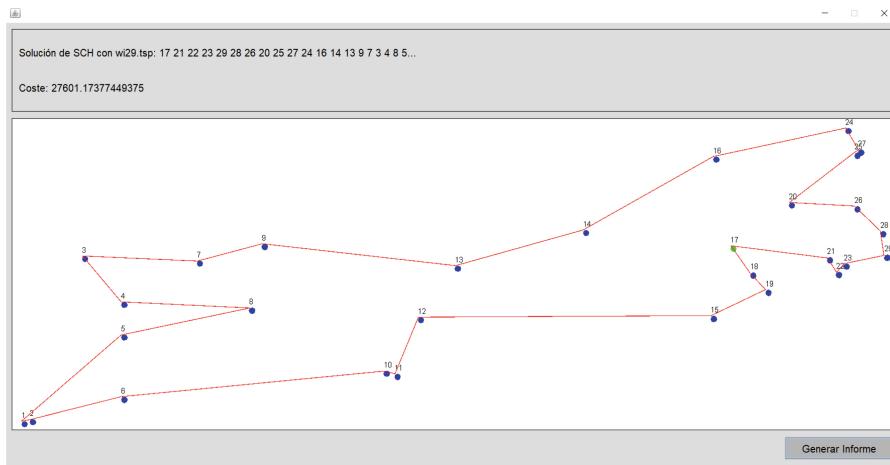


Fig. 2. Second screen of the proposed user interface.

The results report thoroughly documents the executed algorithm, capturing its configuration parameters and values used. It also records the dataset used for analysis and details the best solution found during the algorithm's run, including its associated cost.

Furthermore, the report provides insights into the algorithm's performance metrics. It specifies the total number of iterations performed during the algorithm's execution and records the overall time taken for completion.

Furthermore, the report includes the geographical coordinates of each city involved in the solution, facilitating graphical representation upon loading into the system. It also documents city-to-city distances and heuristic data.

Lastly, the application can also be executed from the command line. This functionality has been incorporated to allow more advanced computer users to concurrently run different algorithms on various datasets.

To conclude the section, a video showcasing the application's functionality³, the application itself⁴, and its source code⁵ are provided.

4 Experimentation

This section evaluates the implemented algorithms by applying them to 9 datasets with varying numbers of cities, running each for up to 600 s or 10,000 iterations, whichever occurs first. For each algorithm and dataset, three seeds are used, presenting the average cost and standard deviation in Table 1. Global results for each algorithm are also provided in Table 2.

Table 1. Results of each algorithm for each dataset.

Dataset	Optimum	ACS length (km)	MMAS length (km)	BWAS length (km)	LK length (km)
29 city	27600	27601,17 ± 0,00	29115,05 ± 479,69	27601,17 ± 0,00	29343,51 ± 397,96
38 city	6656	6659,43 ± 0,00	6869,60 ± 157,82	6659,43 ± 0,00	7139,41 ± 227,36
194 city	9352	12032,36 ± 37,18	11992,47 ± 625,49	16852,46 ± 325,65	9691,12 ± 41,81
734 city	79114	129921,63 ± 1299,24	116042,17 ± 1442,85	204620,70 ± 7802,01	88016,74 ± 2946,95
929 city	95345	156929,61 ± 801,84	141275,98 ± 3629,33	233098,25 ± 9171,69	105650,37 ± 2021,67
1979 city	86891	452286,40 ± 19812,43	172895,29 ± 30818,94	462804,00 ± 2846,78	262694,67 ± 29175,41
7146 city	172387	1440459,32 ± 44898,15	1456142,94 ± 3271,35	1476520,22 ± 11691,16	454517,68 ± 9436,98
8246 city	206171	2522195,43 ± 43151,61	2591233,15 ± 31469,57	2590304,68 ± 20392,09	453941,04 ± 6709,09
14185 city	427377	5864485,55 ± 32839,03	6122171,70 ± 69410,71	6122171,70 ± 69410,71	1107241,04 ± 7745,48

Table 2. Overall results for each algorithm.

Algorithm	Length (km)	Seconds
ACS	1179174,55	563,30
MMAS	1183082,04	587,89
BWAS	1237848,07	520,52
LK	279803,95	14797,78

If we focus exclusively on bio-inspired algorithms, the ACS achieves better results with a similar runtime to the other two methods.

On the other hand, the deterministic algorithm LK consistently delivers superior results. In datasets with many cities, all algorithms perform few iterations due to limited runtime, which affects solution quality. Although LK requires more time per iteration than bio-inspired methods, its ability to find more effective solutions results in better average outcomes in this specific context.

In conclusion, extending the runtime of bio-inspired algorithms could potentially enhance average results compared to LK, enabling more iterations in less

³ <https://acortar.link/KXlpMd>.

⁴ <https://github.com/diva0001/aplicacionTSP>.

⁵ <https://github.com/diva0001/TFG>.

time. Analyzing the results suggests that LK is highly suitable for smaller datasets, delivering rapid and high-quality solutions. However, its efficiency diminishes for larger datasets, reducing its recommendation for such cases.

5 Conclusions

The desktop application developed enables users to easily apply metaheuristics for route optimization, improving existing systems such as the Concorde TSP Solver. It allows visualization of results to compute various metrics and facilitates generating and importing results in PDF format for analysis. Additionally, it supports concurrent execution of algorithms from the command-line interface.

Some potential future improvements to the system aim to integrate new algorithms and expand problem-solving capabilities beyond TSP.

In conclusion, an intuitive software application has been developed to facilitate algorithm learning for users of all levels. This enables them to develop additional competencies such as analytical and reasoning skills, particularly when comparing different algorithms.

References

1. Colorni, A., Dorigo, M., Maniezzo, V.: Distributed optimization by ant colonies. In: Proceedings of ECAL91-European Conference on Artificial Life, pp. 134–142. Elsevier, New York (1991)
2. Colorni, A., Dorigo, M., Maniezzo, V.: An investigation of some properties of an ant algorithm. In: Proceedings of Parallel Problem Solving from Nature Conference (PPSN 92), pp. 509–520. Elsevier, New York (1992)
3. Dorigo, M.: Optimization, learning and natural algorithms. Ph.D. dissertation, DEI, Politecnico di Milano, Italy (1992). (in Italian)
4. Dorigo, M.: The any system optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybern.-Part B **26**(1), 1–13 (1996)
5. Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., Shmoys, D.B.: The traveling salesman problem: a comprehensive survey. Oper. Res. **34**(2), 319–368 (1985)
6. Lin, H., Gao, L., Wang, X.: Solving the traveling salesman problem using multi-agent simulated annealing algorithm. Appl. Soft Comput. **11**(7), 6113–6120 (2011)
7. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J. Global Optim. **39**(3), 459–471 (2006)
8. Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., Shmoys, D.B.: The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization. Wiley, Hoboken (1985)
9. Bitner, J.R., Reingold, E.M.: Backtrack programming techniques. Commun. ACM **18**(11), 651–656 (1975)
10. Bellman, R.: Dynamic Programming. Princeton University Press, Princeton (1957)
11. Liu, C.-C., Kernighan, B.W.: An effective heuristic algorithm for the traveling-salesman problem. Oper. Res. **21**, 498–516 (1973)
12. Gambardella, L.M., Dorigo, M.: Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans. Evol. Comput. **1**(1), 53–66 (1997)

13. Stützle, T., Hoos, H.H.: MAX-MIN Ant System. *Futur. Gener. Comput. Syst.* **16**(8), 889–914 (2000)
14. Cordón, O., Fernández de Viana, I., Herrera, F., Moreno, L.l.: A new ACO model integrating evolutionary computation concepts: the best-worst ant system. In: *Actas de A '2000*, pp. 22–29 (2000)
15. Laporte, G.: The traveling salesman problem: an overview of exact and approximate algorithms. *Eur. J. Oper. Res.* **59**(2), 231–247 (1992)



Analysis of School Dropout Rate in Paraguay Using a Machine Learning Approach

Diego R. Insfrán-Coronel¹, Enzo M. Enrique-Sánchez¹, Federico Beck¹,
Aurelio López-Fernández², and Miguel García-Torres²(✉)

¹ Universidad Nacional de Concepción, Concepción, Paraguay

² Data Science and Big Data Lab, Pablo de Olavide University,
ES-41013 Seville, Spain

mgarciat@upo.es

Abstract. This study investigates the school dropout rates in Paraguay, focusing on the transition from ninth grade to the first year of secondary school in the Concepción department. Using an extract, transform, and load (ETL) process, data from the Paraguayan Ministry of Education and Science and the National Institute of Statistics were analyzed. The research employs clustering techniques, particularly K-means, to identify patterns and risk profiles among students. The findings highlight the significant impact of socio-economic factors, such as poverty and child labor, on school dropout rates. These insights aim to inform targeted interventions to improve educational outcomes and reduce dropout rates in Paraguay.

Keywords: education · clustering · data mining

1 Introduction

Thousands of young people in Paraguay are compelled to drop out of secondary school every year, interrupting their educational path. This widespread problem in the country, especially in the first department, goes beyond just numbers, representing a complex issue with significant effects on the social, economic, and cultural progress of a nation [8].

The phenomenon of school dropout in Paraguay has a greater impact on the most vulnerable segments of society, and the economic condition of the household is a crucial factor in this regard. As socioeconomic circumstances become more unfavourable, the probability increases that students will not gain the essential knowledge needed to attain sufficient salaries in maturity. One of the main reasons why kids drop out of school is poverty, which worsens the educational system's failure to support and meet the needs of children who are facing difficulties or unique situations [2].

Although there have been determined attempts to tackle this problem, the effectiveness of compensatory measures, such as distributing school supplies and

offering scholarships, is still limited. It is crucial to adopt a comprehensive and diversified approach that addresses the underlying reasons for dropout from several perspectives.

This research aims to investigate this problem at the local level by conducting a thorough analysis of the dropout rate from ninth grade to the first year of secondary school in the department of Concepción. More precisely, a procedure known as extract, transform, and load (ETL) is used to acquire several complex datasets from the Paraguayan Ministry of Education and Science and the Paraguayan National Institute of Statistics. The ETL procedure guarantees data integrity and accuracy by meticulously preparing the datasets. This entire procedure facilitates further analysis by researchers to extract useful knowledge and draw significant inferences from that knowledge.

Data mining serves a crucial role in extracting useful knowledge from the data obtained during the ETL process. It serves as an essential tool for analysing large amounts of data and uncovering hidden patterns, correlations, and trends. Hence, the use of data mining techniques can facilitate the examination of these datasets to identify hitherto unrecognised patterns or connections, offering an essential understanding of the complex dynamics of school dropout.

Recent studies on student dropout have emphasised the vital significance of data mining in evaluating and forecasting the factors that impact dropout. An instance of this is research conducted by Abideen, Z.U. et al. [1], which employed data mining methodologies such as classification and regression algorithms to predict potential dropouts and enhance the literacy rate in secondary schools. Another study carried out by Venkatesan, R.G. et al. [9] used clustering methodologies to precisely identify regions exhibiting elevated dropout rates, increased repeat rates, and decreased promotion rates. By specifically focusing on these areas, the study assists education policy in identifying high-dropout districts and devising comprehensive intervention measures.

In this work, we present a comprehensive strategy to tackle the issue of school dropout in Paraguay, acknowledging its substantial influence on the country's social, economic, and cultural advancement. We use the extract, transform, and load (ETL) process to obtain intricate information from trustworthy government sources, enabling us to conduct a thorough investigation of the school dropout rate in the Concepción department. This procedure enables us to guarantee the integrity and correctness of data, thereby facilitating robust and relevant analysis. By using clustering approaches, such as K-means algorithm, we are able to detect concealed patterns within student dropout data. This enables us to classify students into high-risk categories and gain a deeper comprehension of the fundamental issues that impact their choice to discontinue their education. By synthesising these discoveries, our objective is to acquire valuable insights that can be utilised to develop precise interventions that target the underlying factors contributing to student dropout rates. These interventions can foster equal educational opportunities and enhance the overall educational environment nationwide.

2 Methodology

This study investigates the causes of school dropout during the transition from the ninth grade to the first year of secondary school in public schools in Paraguay, specifically focusing on the department of Concepción. The methodological approach relies on a thorough extraction, transformation, and loading (ETL) procedure to guarantee the accuracy and dependability of the collected data. Next, complex data mining techniques are applied to uncover concealed patterns and correlations that may offer more profound insights into the dropout issue.

Firstly, this section offers an elaborate explanation of the dataset and the ETL (Extract, Transform, Load) process employed to acquire the findings of this study. Subsequently, details are provided regarding the use of clustering methodologies on the derived dataset. The purpose of these results, which are obtained using data mining techniques, is to simplify the process of extracting valuable knowledge from the data.

2.1 Dataset and ETL Process

The main data sources for this study consisted of the publicly accessible databases of the Paraguayan Ministry of Education and Science (MEC) and the National Institute of Statistics (INE). The datasets contained extensive information, encompassing family poverty levels, school attendance records, school lagging, inadequate education statistics, and data on child labour among individuals aged 10–17.

The process of collecting data was carefully and systematically planned and carried out. The INE's Encuesta Permanente Continua de Hogares [4] (EPHC) offers a detailed analysis of poverty levels and associated variables within households, encompassing both national and sub-national perspectives. The dataset used (EPHC 2022), consisted of 5018 rows and 1127 columns. It contained important fields such as household identifiers, department, area (urban or rural), and other poverty indicators. The MEC enrolment database [3] contains 2719 rows and 22 columns. It offers detailed information about the number of enrolments in scientific and technical secondary education, categorised by gender, department, and area of residence.

The ETL procedure comprised three essential processes, each specifically developed to guarantee the strength and dependability of the data utilised in this research. Initially, the extraction phase prioritised the collection of pertinent data fields from the datasets given by the Instituto Nacional de Estadísticas (INE) and the Ministerio de Educación y Ciencias (MEC). The extraction process was conducted with great care, encompassing the collection of comprehensive information on household identifiers, geographical locations (department and region), poverty indicators, school attendance records, delayed schooling statistics, incomplete education data, and child labour figures. This stage was crucial in establishing the foundation for further analysis by ensuring that the data obtained was thorough and relevant to the research objectives.

Subsequently, the transformation phase significantly contributed to improving the quality and uniformity of the retrieved data. The initial stage of this process involved data cleaning, which entailed the identification and rectification of problems, such as the presence of missing numbers and inconsistencies. Data cleansing is crucial, especially when working with information from several sources, as inconsistencies can have a substantial impact on the correctness of the research. Another crucial objective during this time was the standardisation of data formats. Standardisation was necessary to assure compatibility and comparability of data points, considering that the datasets came from different institutions with potentially varied data protocols. For example, the numerical values were consistently formatted, and the categorical variables were organised according to a standardised schema. The process of standardisation was crucial in ensuring the accuracy and consistency of data, as well as enabling smooth integration of data.

In addition, the transformation phase involved the crucial work of combining datasets using shared identifiers, such as household IDs and geographical codes. The purpose of this merging method was to combine separate data sources into one unified dataset, enabling a more thorough analysis. By integrating data from the INE (National Institute of Statistics) and MEC (Ministry of Education and Culture), we successfully established a correlation between educational outcomes and socio-economic variables. This analysis has yielded a more comprehensive and detailed comprehension of the various elements that contribute to school dropout rates. The integration of this dataset facilitated the examination of intricate connections and patterns that would have been indistinguishable from separate data sources.

Ultimately, the converted data was imported into a centralised database, a crucial process that guaranteed the data was systematically arranged and readily available for examination. The centralised database provided the basis for later data mining and statistical analysis. Through the process of centralising the data, we have established a system where all team members have the ability to access a singular and reliable source of information. This approach effectively reduces the likelihood of inconsistencies that may arise from the use of numerous versions of data. In addition to importing the data, the process also included the creation of indexes and partitions inside the database. This was done to enhance the performance of queries, allowing for efficient retrieval and processing of enormous amounts of data. This configuration was especially crucial for conducting intricate studies, such as clustering and cross-tabulation, that necessitate significant computer resources.

Consequently, the ETL process involved multiple steps to establish the groundwork for the research by guaranteeing the accuracy, consistency, and comprehensiveness of the data used. The extraction phase systematically collected pertinent data, the transformation phase purified and standardised the data while including many sources; and the loading step guaranteed that the data was arranged and easily accessible for analysis. The ETL method used was strong and allowed for a thorough analysis of school dropout rates. This analysis

helped identify patterns and provide significant insights into the socio-economic aspects that affect educational results in Paraguay.

2.2 Clustering Techniques

The data study utilised various advanced methodologies to extract significant insights into the factors that contribute to the rates of students leaving school prematurely in Paraguay. Descriptive statistics were initially employed to establish a fundamental comprehension of the data distribution and the central patterns of the variables. This stage was essential as it enabled us to comprehend the fundamental indicators of the dataset, such as household income, levels of education, and school attendance rates. The descriptive statistics provided a concise overview of the entire terrain and emphasised initial patterns and irregularities that required additional examination.

In order to reveal more intricate patterns, the K-means clustering algorithm was utilised. This method was highly efficient in identifying cohorts of students with comparable characteristics who are at a heightened risk of discontinuing their education. K-means clustering is a method that divides a dataset into k separate groups, taking into account the similarity between data points. Within this framework, it categorised kids into clusters based on their socio-economic status, school attendance records, and other pertinent variables. Through the analysis of these clusters, we successfully identified unique risk profiles within the student population. For instance, one cluster could indicate pupils from severely deprived households with inconsistent school attendance, while another cluster could consist of students from rather secure economic origins but burdened with substantial family obligations. These clusters offered a subtle comprehension of the various routes that result in school dropout.

The clustering analysis unveiled intricate patterns and relationships that were not readily apparent from the unprocessed data. For example, it emphasised particular combinations of characteristics that were substantially linked to increased percentages of students leaving school before completion. An important discovery was the substantial influence of severe poverty and child labour on the ability of students to continue attending school. Students from economically disadvantaged households and those involved in labour activities were grouped together, suggesting a greater likelihood of dropping out. This revelation was essential in formulating precise therapies targeted at these vulnerable populations.

In addition, cross-tabulation was employed to examine the associations between factors such as poverty rates, school attendance, and participation in scientific and technical education, taking into account gender and geographical location. Through cross-tabulation, we were able to analyse the intersection and mutual influence of many elements. For example, we observed the correlation between poverty levels and school attendance rates in various geographical regions and among different genders. The investigation revealed that rural regions had elevated poverty rates and decreased school attendance, particularly among male students. Moreover, the analysis of cross-tabulation data indicated that girls residing in rural areas exhibited a higher propensity to discontinue

their education as a result of socio-cultural influences, such as early marriage and domestic obligations, in contrast to their urban peers [5].

The implementation of a multi-faceted analytical technique allowed for a comprehensive study of the data from several perspectives, resulting in a thorough comprehension of the dropout phenomenon. By combining descriptive statistics, kMeans clustering, and cross-tabulation, we obtained a comprehensive understanding of the factors that impact school dropout rates. This method not only determined the main causes of students leaving school but also emphasised the intricate interaction among many socio-economic elements. The findings emphasised the significance of simultaneously addressing many aspects of the issue, rather than dealing with them separately.

Furthermore, the utilisation of kMeans clustering and cross-tabulation aided in the identification of precise intervention points. For instance, the clusters discovered by employing the kMeans algorithm yielded distinct objectives for policy interventions, such as implementing financial assistance programmes for students living in severe destitution and launching educational initiatives to mitigate child labour. Furthermore, the utilisation of cross-tabulation outcomes facilitated the identification of regions with the most pressing requirements, hence allowing for a more effective distribution of resources. The comprehensive analytical approach was used to construct strategic interventions with the goal of reducing dropout rates and enhancing educational outcomes in Paraguay.

Ultimately, the utilisation of advanced data analysis methods such as descriptive statistics, kMeans clustering, and cross-tabulation in this study offered a strong foundation for comprehending the elements that contribute to school dropout rates. These tools provided valuable insights into the socio-economic processes and helped identify specific actions. The results of this analysis are crucial for guiding policy decisions and implementing efficient methods to tackle the issue of high school dropout rates in Paraguay.

3 Results and Discussion

This section outlines the results of two data studies carried out to determine the factors that contribute to the rates of students leaving school prematurely and other educational problems in Paraguay. We utilised advanced statistical methods, such as descriptive statistics, K-means clustering, chi-square testing, and ANOVA, to derive meaningful insights on enrollment patterns, socio-economic conditions, and student risk profiles. The first data analysis examined the enrollment rates in scientific, technical, open secondary, and vocational training education. The analysis was conducted by considering gender and geographical location. The second analysis included socio-economic and educational factors such as household poverty, school attendance, school delay, minimum educational achievement, and child labour.

3.1 Enrollment Rates Based on Levels of Education

The preliminary descriptive analysis of the enrollment variables offered a comprehensive perspective on the distribution of the data. The descriptive statistics indicate that the average enrollment in scientific fields for males is 31.34, with a standard deviation of 29.53. For females, the average enrollment is 33.70, with a standard deviation of 32.73. The mean enrollment for males in technical fields is 7.49, while for females it is 8.06. Enrollments in open secondary and vocational training exhibit a significant decline, with average values consistently below 0.5 in the majority of instances.

The K-means clustering technique was employed to detect clusters of students exhibiting similar features, utilising the enrollment factors. By employing the elbow technique, we were able to determine the optimal number of clusters in advance, resulting in the identification of three different clusters. Cluster 0 comprises the largest number of institutions (2580 instances) with reasonable enrollments in scientific and technical fields for both males and females, but very few students are pursuing open secondary and vocational training. Cluster 1 exhibits exceptionally high enrollments in scientific and technical disciplines for both males and females, along with significant enrollments in vocational training. This group comprises institutions that exhibit a high degree of specialisation and a notable emphasis on technical and scientific education. Cluster 2 comprises institutions with minimal student numbers in scientific and technical fields, but significant student numbers in open secondary education. This group likely comprises students that have a preference or requirement for a more adaptable or unrestricted educational approach, potentially as a result of personal or financial factors [6].

A chi-square test was used to assess the significant relationships between clusters and geographical areas (rural/urban). The analysis yielded a p-value of $3.06 * 10^{-18}$, which indicates a highly significant connection. This implies that the distribution of clusters is greatly influenced by the geographic region [7]. In addition, ANOVA tests were performed to compare the means of enrollment factors among the clusters. The ANOVA analysis revealed statistically significant differences in all enrollment factors, as seen by the exceptionally low p-values. These findings indicate that the observed differences between clusters are statistically significant. As an illustration, the scientific enrollment data for males yielded an F-value of 737.08, along with a p-value of 1.94×10^{-256} . This confirms that the disparities across the clusters are statistically significant.

As shown in Fig. 1, examining the distribution of clusters based on gender and geographical region unveiled significant trends. Also, the visualisations indicate that there is a notable disparity in the distribution of clusters between rural and urban areas, as well as within various departments. This emphasises the necessity for customised strategies based on the pupils' geographical region and gender.

By analysing the detected clusters, it is possible to extract distinct risk profiles. Cluster 0 comprises students who have a moderate number of enrollments in scientific and technological fields. This group would greatly profit from ini-

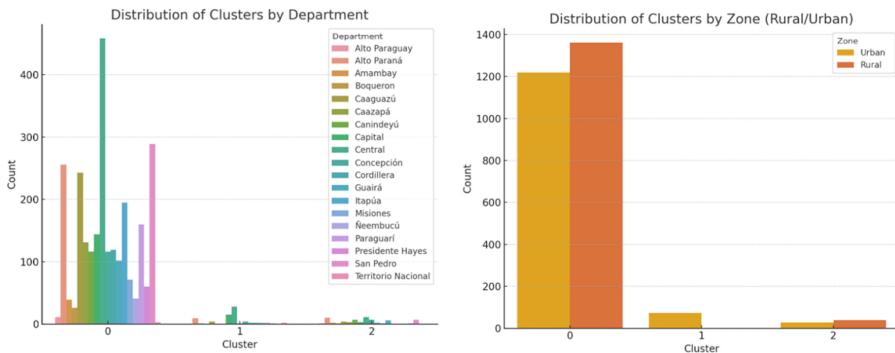


Fig. 1. Distribution of clusters by department and zone.

tiatives aimed at enhancing infrastructural and educational resources in these regions. Cluster 1 consists of students who have exceptionally high enrollments in scientific and technological fields. Interventions in this context might prioritise the preservation and improvement of these sophisticated programmes by offering additional assistance and resources to sustain their academic progress. Cluster 2 exhibits significant enrollments in open secondary education, potentially associated with economic or personal necessities. Possible treatments may involve providing monetary assistance and adaptable educational initiatives to enable these individuals to pursue their studies despite economic or personal obstacles.

3.2 Socio-economic and Educational Factors

The mean poverty index was determined to be 2.73, with the bulk of values concentrated around 3, indicating a significant degree of poverty among the households polled. The statistics also indicated that the average number of children not attending school per family was 0.04, implying that the majority of households reported no children who were not enrolled in school. Likewise, the occurrence of school delays was minimal, with an average of 0.07, suggesting that this problem was not widespread in the majority of homes. In addition, the average value for the lowest level of educational attainment was 0.11, indicating that a tiny yet significant proportion of households had children who did not fulfil the minimal educational requirements. Ultimately, the mean number of children between the ages of 10 and 17 who were engaged in any form of labour, likely child labour, was 0.03. This indicates that the occurrence of this problem within the examined population is extremely uncommon.

Next, the K-means clustering technique was utilised to detect clusters of families that share similar attributes, specifically related to socio-economic and educational factors. The study revealed three separate groups, each exhibiting distinct traits. Cluster 0 emerged as the most populous group, comprising 4334 households. These homes had a significant poverty index, with an average of 2.76. Although facing economic difficulties, they demonstrated minimal rates of

children's non-attendance and school delays, with respective averages of 0.02 and 0.05. The cluster had minimal occurrences of issues pertaining to minimum educational achievement and child labour.

On the other hand, Cluster 1 consisted of 154 households and had a distinct character. The average poverty index of these households was 2.64, which was lower. Nevertheless, they encountered substantial educational obstacles. The prevalence of children not attending school was significantly greater, with an average of 0.32, and there was also a larger frequency of school delays, averaging 0.25. Furthermore, this cluster demonstrated significant problems concerning the minimum level of educational attainment, with an average of 0.16. Additionally, there was a prevalent occurrence of child labour, since all households in this cluster reported at least one kid between the ages of 10 and 17 involved in labour.

Cluster 2, consisting of 529 homes, has the lowest average poverty index among all the clusters, with a value of 2.44. Although the rates of children not attending school and school delays were very low, averaging 0.09 and 0.11, respectively, this group stood out due to considerable problems concerning minimum educational achievement, as reported by all households. Nevertheless, child labour does not pose a substantial problem within this cluster. The graphs provided depict the distribution of socio-economic and educational factors among the clusters. These visualisations elucidate the unique attributes of each cluster and emphasise the diverse levels of educational difficulties and socio-economic circumstances encountered by the households.

When analysing these clusters, it is evident that Cluster 0 contains homes characterised by high poverty levels but minimal occurrences of educational difficulties and child labour. This implies that interventions targeted at this demographic might prioritise poverty reduction initiatives in order to enhance their overall quality of life. On the other hand, Cluster 1, consisting of homes with relatively lower poverty levels but facing significant educational challenges and child labour, is highly vulnerable and would greatly benefit from focused interventions. These initiatives should focus on improving school attendance, reducing school delays, enhancing educational outcomes, and striving to eliminate child labour. Cluster 2, on the other hand, consists of households that have the lowest poverty levels but have moderate educational challenges, especially when it comes to meeting minimal educational requirements. This group would benefit from educational interventions focused on enhancing learning outcomes and ensuring that children attain the requisite educational benchmarks.

4 Conclusions and Future Works

The study provides a comprehensive analysis of school dropout rates in Paraguay, revealing the critical role of socio-economic factors, including poverty and child labor, in influencing educational outcomes. The use of advanced data mining techniques, such as kMeans clustering, has identified distinct risk profiles and patterns within the student population. These findings underscore the necessity for targeted interventions that address the unique needs of vulnerable groups to mitigate dropout rates.

Future research should focus on expanding the dataset to include additional regions of Paraguay, enhancing the generalizability of the findings. Moreover, integrating longitudinal data could provide deeper insights into the long-term effects of socio-economic factors on educational attainment. Further studies could also explore the effectiveness of specific interventions, such as financial aid programs and educational support initiatives, in reducing dropout rates. Finally, other clustering techniques should be applied to overcome the limitations of K-means.

References

1. Abideen, Z.U., et al.: Analysis of enrollment criteria in secondary schools using machine learning and data mining approach. *Electronics* **12**(3), 694 (2023)
2. Ali, K., Yaseen, M.R., Makhdum, M.S.A., Quddoos, A., Sardar, A.: Socioeconomic determinants of primary school children dropout: a case study of Pakistan. *Int. J. Educ. Manag.* **35**(6), 1221–1230 (2021)
3. de Educación y Ciencias, M.: Matriculaciones Educación Media (2024). https://datos.mec.gov.py/data/matriculaciones_educacion_media. Accessed 14 July 2024
4. de Estadística, I.N.: Encuesta Permanente de Hogares Continua (EPHC) (2024). <https://www.ine.gov.py/microdatos/Encuesta-Permanente-de-Hogares-Continua.php>. Accessed 14 July 2024
5. Hossain, T.: Factors leading to school dropout in Bangladesh: an empirical approach. *Glob. J. Hum.-Soc. Sci.* **45**–49 (2021)
6. Naz, U., Ejaz, Z., Khan, N.: Determinants of dropout and child school enrollment: a case study from rural Islamabad. *J. Quant. Methods* **3**(2), 77–89 (2019)
7. Pillai, T.J.: An exploratory study of factors affecting school dropouts of Pune district. *Indian J. Sci. Technol.* **12**(36), 1–7 (2019)
8. Reyes, G.E., Sosa, D.V., Quispe, M.G., Iraola-Real, I.: Family, economic and pedagogical factors involved in dropping out of school and the consequences for students of a private institution in Lima-Peru. In: 2021 IEEE Sciences and Humanities International Research Conference (SHIRCON), pp. 1–4. IEEE (2021)
9. Venkatesan, R.G., Mappillairaju, B.: Detection of hotspots of school dropouts in India: a spatial clustering approach. *PloS One* **18**(1), e0280034 (2023)



Exploring Educational Trends: Specializations in Secondary Education in Paraguay from 2018 to 2021

José A. Torres-Báez¹, Juan B. Torres-Báez^{1,2}, Aurelio Lopez-Fernandez^{2(✉)},
Francisco A. Gomez-Vela², and Federico J. Beck¹

¹ Universidad Nacional de Concepción, Concepción, Paraguay

² Universidad Pablo de Olavide, Seville, Spain

alopfer1@upo.es

Abstract. Paraguay's education system has undergone significant changes, particularly at the secondary level, introducing new specializations and teaching methods. While this diversity offers students unique opportunities, it also presents challenges in selecting a suitable specialty. Examining the variety and demand of specializations provides insights into educational trends and job market needs. Analyzing gender distribution across fields can help address disparities. Additionally, factors like accessibility, curriculum variety, overage students, and indigenous inclusion must be considered. Advanced methods like exploratory data analysis (EDA) are essential for understanding these complexities. This study introduces a tool for EDA and comprehensive investigation of enrollment data, aiming to provide valuable insights for students. The importance of EDA in educational research is emphasized, along with advancements in its implementation.

Keywords: Secondary education · educational trends · exploratory data analysis

1 Introduction

Paraguay's education system has experienced notable transformations in recent years, particularly at the secondary level, where novel specializations and instructional methods have been introduced [9]. This variety, although it enhances the educational options available, can create confusion for students when selecting a specialty due to the absence of clear direction on which one would best align with their interests and future aspirations.

It is crucial to analyze the range of educational specializations that are now available in this particular environment. The multitude of choices provides students with a distinctive chance to delve into their interests and abilities, but it can also be daunting in the absence of explicit supervision. Furthermore, comprehending the progression of demand for academic disciplines across time can offer useful insights on educational patterns and labor market requirements.

Another crucial issue to examine is the gender distribution among students in various disciplines of study. Gender inequalities in education may arise from structural and socio-cultural obstacles that restrict some groups from accessing specific areas of study. Examining these disparities can aid in identifying areas of inequity and formulating solutions to mitigate them.

The accessibility and variety of educational opportunities are other important factors to take into account. The availability of a diverse selection of educational curricula can have an impact on the fairness and excellence of education. Moreover, the existence of specialized alternatives can offer distinct prospects for cultivating particular proficiencies demanded in the contemporary job market.

However, it is crucial to tackle the concerns of students who are older than their expected age for their school level and those who repeat grades. These events may suggest difficulties in the school system, such as a deficiency in academic assistance or inadequacies in the curriculum. To provide a fair and high-quality education for all kids, it is crucial to identify and address the root causes of students being older than their grade level and repeating grades.

Furthermore, the inclusion of indigenous students in Paraguayan secondary education is a significant factor to take into account [4]. These pupils may encounter distinct obstacles associated with culture, language, and the availability of educational resources. Gaining insight into their experiences and requirements is crucial to guaranteeing that they receive sufficient assistance and all-encompassing instruction. Within this particular context, the analysis of educational data necessitates the utilization of sophisticated techniques in order to obtain a comprehensive understanding of all these facets and ultimately enhance the educational community as a whole.

Exploratory data analysis (EDA) is a powerful tool for comprehending the complexity of the education system and enhancing decision-making. For example, MATLAB software has been used in engineering classes to give students a powerful tool for analysing data [3]. Also, PowerBI has become a pertinent tool for exploratory data analysis (EDA) in educational settings. In the university context, the authors A. Khojastehfar et al. [8] used it to transform Moodle data into visual reports for decision-making purposes. These diverse software tools highlight the critical role of EDA in understanding and improving educational systems.

This paper introduces a tool that enables both exploratory data analysis and in-depth research of current educational trends. This integrated method will provide a thorough comprehension of enrollment datasets and offer significant insights for young individuals contemplating their baccalaureate choices. Additionally, this paper will emphasize the significance of EDA analysis in current educational research and examine the latest advancements in its implementation within this particular context.

The structure of this work is as follows. Section 2 provides a comprehensive explanation of the tool employed to gather and analyse the enrollment data, with specific emphasis on conducting exploratory data analysis. Section 3 displays the acquired results, which include the ranking of specializations, the demographic

distribution of students, and other pertinent metrics. Section 4 provides an analysis of the study's findings and proposes potential avenues for future research.

2 Methodology

The primary goal of this work is to provide a visual tool that assists the scientific and educational communities in conducting exploratory data analysis, even without requiring considerable computer proficiency.

To begin with, this section provides a detailed description of the dataset that was used to derive the outcomes of this study. Next, it provides information about the tool that was created and the filtering process that was built to generate graphs. These graphs are intended to facilitate the extraction of useful knowledge from the data.

2.1 Dataset

The data for this study was obtained by a thorough examination of the number of students enrolled in different specializations of the baccalaureate programme throughout all departments of the country from 2018 to 2021. The purpose of this study was to examine and analyse the changes and distribution of student enrollments over a four-year period. The primary data source used for this analysis was the CSV files found in the Microdata section of the Scholar Map [10], an official platform provided by the Ministry of Education and Science of Paraguay.

The significance of this research rests in its capacity to offer a thorough and all-encompassing depiction of the fluctuations in baccalaureate enrollments at the national level. This, in turn, can serve as a foundation for well-informed decision-making by both students and education authorities.

In order to conduct this study, the PowerBI software was utilised as the primary tool for importing, merging, and analysing the CSV files for each year. This allowed for the consolidation of the data into a single table, facilitating its modification and analysis.

The generated datasets are of substantial scale, with a substantial number of rows and a constant size of 34 columns, which accurately represent the extensive and intricate information gathered. In 2018, a total of 1,452,613 rows were gathered, and in 2019, there were 1,480,530 reported enrollments. The number of enrollments registered for the years 2020 and 2021 was 1,509,523 and 1,528,216, respectively. The number of enrollments resulted in a final dataset for further exploratory data analysis with a total of 5,970,882 rows and 34 columns.

Each enrolment is documented with information on the school year, location (including department, district, area, and zone), educational institution, sector, and educational provision. The student information includes data on nationality, age, gender, indigenous status, and if they are repeaters or over-age students. Additional pertinent details, such as the mode, year, and shift, can also be noticed in the final dataset.

Crucially, the data has undergone several pre-processing processes, such as normalisation, standardisation, and consolidation, to guarantee the accuracy and dependability of the acquired results. The pre-processing stage is crucial for ensuring data quality and facilitating future analysis.

The findings of this study will offer a comprehensive and contextualised summary of the patterns in secondary education enrollments within the specified time period. These findings are relevant not only to researchers and academics in the field of education but also have significant practical implications for young students. They can use this information to analyse their options and make well-informed decisions about the specializations offered in the baccalaureate, based on their interests and career goals.

2.2 Education Data Analysis Tool

The tool has been developed to conduct an exploratory data analysis (EDA) within the framework of the educational system in Paraguay. This approach will facilitate decision-making for the scientific and educational community, and provide valuable guidance for students in selecting their desired specialism. However, considering the significant number of enrollments that need to be considered, and anticipating that this number will continue to grow each academic year, it is essential for this tool to have the capacity to handle substantial volumes of data.

The methodology was based on PowerBI, a powerful and flexible platform known for its efficient and effective data visualisation and analysis capabilities. Furthermore, this platform is extensively embraced by the scientific and educational communities. PowerBI enables efficient handling of enormous data volumes, seamless integration with diverse data sources, and the generation of intuitive reports and interactive visualisations.

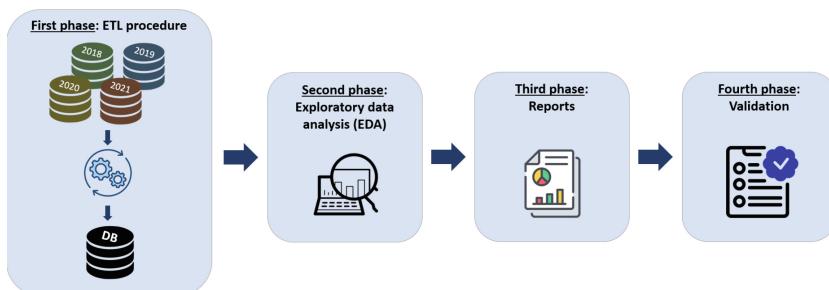


Fig. 1. PowerBI workflow consisting of four phases: ETL procedure, EDA, report generation and subsequent validation.

As shown in Fig. 1, the use of PowerBI for this methodology encompasses four primary stages: an extraction, transformation, and loading (ETL)

procedure, an exploratory data analysis (EDA), the creation of visualisations and reports, and validation.

The ETL procedure involved gathering enrollment data from the Microdata portion of the School Map, which is maintained by the Ministry of Education and Science of Paraguay. The data was acquired for the years spanning from 2018 to 2021. For further details, please refer to Sect. 2.1. The data, which is spread across various databases, is imported into PowerBI to extract the pertinent information from the source database. Subsequently, a data transformation procedure is executed to guarantee the coherence and soundness of the data. This entails the process of cleaning, removing duplicates, and revising the format of data. After completing the data transformation process, a unified database is created containing all pertinent and modified enrolments.

Subsequently, a preliminary examination of the data obtained during the initial stage of the approach is conducted. This analysis has generated various statistics to identify significant patterns and trends in educational data. These include the evolution of Baccalaureate specializations, the distribution of enrollments by gender, and the presence of over-age students, among other important aspects.

During the third phase of this methodology, the findings derived from the exploratory data analysis were utilised to produce comprehensive and interactive reports and visualisations using PowerBI. These reports offered a lucid and comprehensible perspective of the educational data, enabling well-informed decision-making by students, educators, and policy officials. Ultimately, the final stage guarantees that these reports and visualisations undergo validation and scrutiny by education professionals to verify their precision and pertinence.

This methodology offers a potent and easily usable instrument for researchers, educators, and policy makers. It allows for a more profound and significant comprehension of educational data and supports well-informed decision-making in the field of education.

3 Results and Discussion

The primary aim of this section is to showcase the relevance and usefulness of this tool for conducting exploratory data analysis. Here, we focus on a series of precise tasks that can aid in the examination of patterns in secondary education in Paraguay.

As shown in Fig. 2, the first specific task is to provide rankings that provide a comprehensive quantification of the number of students enrolled in each specialty. The quantification can be modified by applying filters to generate findings unique to a certain geographic area, educational institution, or school year.

This figure illustrates that social sciences had the greatest enrollment numbers from 2018 to 2021, totaling 407,954 enrolments. In addition, the number of enrollments in this field is 2.35 times greater than the number of enrollments in basic sciences and 7.50 times greater than the number of enrollments in computer science. This factor is intriguing because there are countries that prioritise

improving disciplines related to science, technology, engineering, and mathematics (STEM) in their education policies. This is done to address the present and future demands of the job market [7].

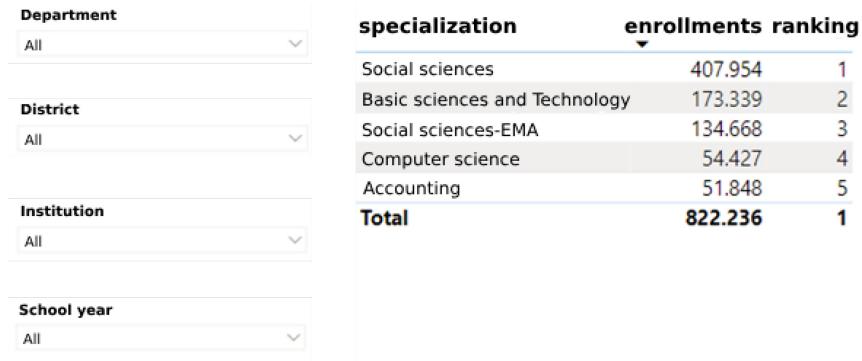


Fig. 2. Example of a ranking design with different filtering.

This tool not only provides ranking structures but also enables us to address specific goals by visualising graphs. Upon closer examination of the aforementioned information, Table 1 illustrates the temporal evolution of enrollment numbers for each discipline.

Table 1. Percentage of enrolments by each of the disciplines with the most students enrolled in secondary education in Paraguay.

	2018	2019	2020	2021
Computer Science	24.3%	24.9%	25.8%	26.9%
Basic Sciences	24.6%	25.4%	25.9%	26.0%
Social Sciences	31.1%	32.5%	33.9%	35.0%
Accounting	25.1%	25.7%	25.9%	25.4%

In this case, it can be observed that the number of enrollments in social sciences and computer science follows a linear trajectory, but disciplines like accounting have experienced a significant decline in the past year. Nevertheless, it is intriguing to observe the stabilisation of the trend in those pursuing basic sciences, despite the fact that the labour market in this field presents significant job prospects in light of the COVID-19 pandemic [1].

Paraguay's approach to indigenous education has been implementing a system of schooling that follows Western standards, with the aim of integrating indigenous people into the broader national society [6]. The future of indigenous

school education in Paraguay is a topic of concern and importance. Several studies conducted from 2013 to 2017 have indicated that 50% of indigenous individuals who successfully finish elementary school proceed to secondary education, and out of these, only 60% successfully complete it [2,11].

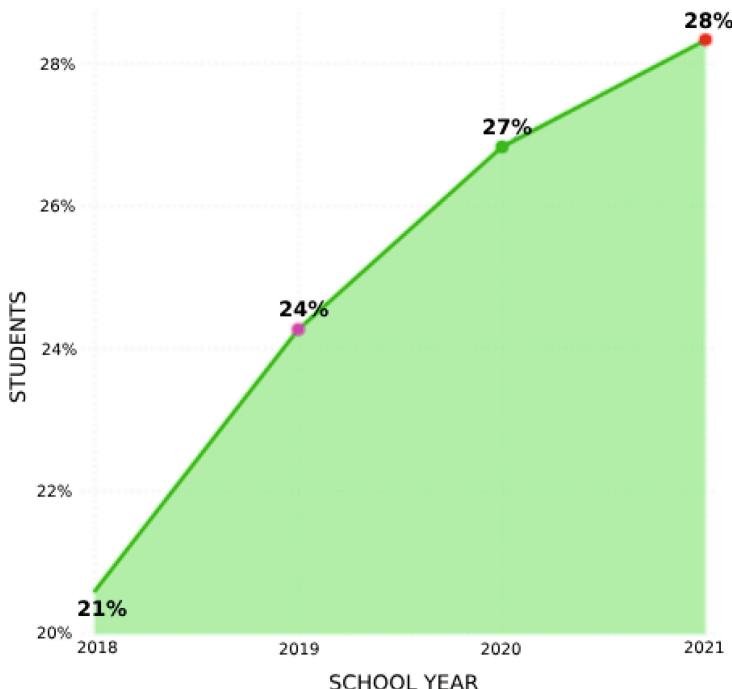


Fig. 3. Percentage of people belonging to indigenous communities who are enrolled in secondary education in Paraguay between 2018 and 2021.

In order to determine if this educational approach maintains its current pattern, Fig. 3 displays the proportion of students from indigenous communities who have registered for secondary education. Here, we observe that the number of enrolments exhibits a nearly linear upward tendency. Hence, we posit that there has been a significant rise in the educational enrollment of individuals from these groups, either starting from the early years of schooling or due to an increase in the transition rate from primary to secondary education. To obtain more definitive results on this matter, it is necessary to conduct a study investigating the conduct of these pupils during their elementary education phase.

The presence of older pupils is also a factor to consider when evaluating the progress of adult education in this country (refer to Table 2). It is noteworthy that the enrollment of adults in secondary education is nearly uniform across all fields in this situation. The population falls between the range of 18,474 to 21,903 individuals. From this perspective, the uniformity for each school year

Table 2. Over-age students enrolled between 2018 and 2022 for the four most representative disciplines.

	2018	2019	2020	2021
Computer Science	0	0	0	0
Basic Sciences	2586	2756	2625	2695
Social Sciences	15263	17181	17288	18470
Accounting	625	825	777	738

is particularly intriguing, considering the number of enrollments for each field exhibits a fluctuation of up to 25.02-fold. This indicates that the proportion of over-age students in different fields varies significantly, depending on the ratio between the total number of registered students and over-age students. Areas within the STEM field, for instance, have a lower percentage of older students compared to areas like social sciences. Furthermore, it is especially relevant how there are no overage students enrolled in the discipline of computer science.

Table 3. Repeater students enrolled between 2018 and 2022 for the four most representative disciplines.

	2018	2019	2020	2021
Computer Science	0	0	9	8
Basic Sciences	327	417	472	612
Social Sciences	1502	1782	2523	3507
Accounting	0	86	149	177

The inclusion of repeaters is a significant aspect to consider while evaluating the advancement of secondary education in this nation (refer to Table 3). It is worth mentioning that the number of students who repeat courses varies greatly throughout the fields studied and remains consistent when examining each subject separately from 2018 to 2021.

Significantly, the field of Computer Science exhibits an exceptionally low incidence of students repeating courses, as seen by the absence of any recorded instances in both 2018 and 2019, and only a few cases in the following years. This phenomena is especially interesting given the difficult difficulty of this course and the evidence suggesting that students repeated this discipline owing to a lack of motivation [12].

In contrast, the fields of Basic Sciences and Social Sciences have had a consistent rise in the number of students who repeat courses over time. The number of students repeating Basic Sciences courses shows a steady growth, going from 327 in 2018 to 612 in 2021. Similarly, in Social Sciences, the number of repeaters climbs from 1502 in 2018 to 3507 in 2021. In the field of Accounting, there is a

fluctuating pattern in the number of individuals who repeat a course. There was a significant rise in 2019, followed by a more gradual increase in the subsequent years.

These statistics indicate that there is considerable variation in the likelihood of students repeating an academic year based on the field of study. Although the field of Computer Science exhibits a decreased occurrence of repetition, fields like Social Sciences and Basic Sciences have difficulties in this aspect. This analysis emphasises the significance of comprehending disciplinary dynamics to efficiently meet the needs of students who are repeating in the secondary education system.

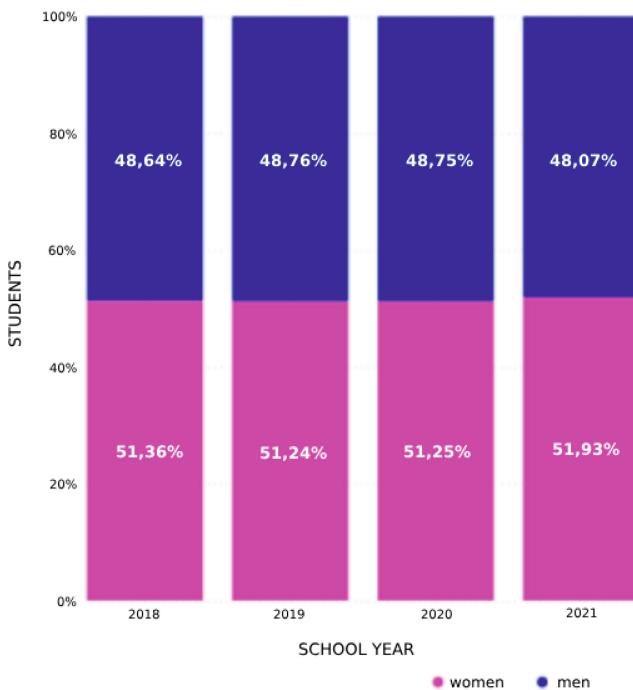


Fig. 4. Percentage of enrolments, categorised by gender, in secondary education in Paraguay between 2018 and 2021.

Additionally, another goal could involve conducting a gender-based examination of enrollment figures. Figure 4 illustrates that the number of women and men in 2018 and 2019 is nearly identical. Despite evidence showing that internationally there are more women than males enrolled in postsecondary and secondary school, the gender disparity in fields like STEM persists to varying degrees in all nations [5]. Hence, it is imperative that we persist in implementing educational initiatives aimed at enhancing these percentages in particular academic subjects.

4 Conclusions and Future Work

This work aims to develop a specialized tool for efficiently searching micro-data provided by Paraguay's Ministry of Education and Science. The tool will use ETL (Extract, Transform, Load) procedures and EDA (Exploratory Data Analysis) techniques to improve the search process. ETL processes consolidate multiple files into a single dataset, ensuring data accuracy through cleaning and filtering. EDA analysis visually represents the data, generating visualizations and tables that can aid the general community and prospective secondary education students in choosing their desired specialization.

References

1. Bachmann, M., Enodien, B., Frey, D.M., Taha-Mehlitz, S., Taha, A.: The development of healthcare jobs in the COVID-pandemic—a new economic market. *Front. Publ. Health* **10**, 848636 (2022)
2. Blanco, C.S.: Inclusión educativa en paraguay: Un análisis de los programas orientados al acceso a la educación de jóvenes en situación de vulnerabilidad y exclusión social. *Población y Desarrollo* **23**(45), 83–94 (2017)
3. Carrasco Fernández, J.J., Soria Olivas, E., Gómez Sanchis, J., Serrano, A.J., Soriano-Asensi, A., Guerrero Martínez, J. F.: Matlab-based educational software for exploratory data analysis (eda toolkit). *Comput. Appl. Eng. Educ.* **20**(2), 313–320 (2012)
4. Demelenne, D.: Paraguayan indigenous peoples and the challenge of citizenship. In: *Global Citizenship Education: Critical and International Perspectives*, pp. 31–42 (2020)
5. García-Holgado, A., et al.: Gender equality in stem programs: a proposal to analyse the situation of a university about the gender gap. In: *2020 IEEE Global Engineering Education Conference (EDUCON)*, pp. 1824–1830. IEEE (2020)
6. Gaska, E.: Futuro de la educación escolar indígena en el paraguay. *Paraguaya* (2013)
7. Kayan-Fadlelmula, F., Sellami, A., Abdelkader, N., Umer, S.: A systematic review of stem education research in the GCC countries: trends, gaps and barriers. *Int. J. STEM Educ.* **9**, 1–24 (2022)
8. Khojastehfar, A., Abarbekouh, H.S., Safai, S.S., Fatemi, O.: Analysis of teaching and learning in Moodle with the help of visualization with power bi: case study of e-learning course of university of Tehran. In: *2023 10th International and the 16th National Conference on E-Learning and E-Teaching (ICeLeT)*, pp. 1–7. IEEE (2023)
9. Naidu, I.: Issues and challenges for mathematics teacher education (2022)
10. M. of education and science of Paraguay. Scholar map (2021). https://mapaescolar.mec.gov.py/mapa_escolar/contenido/microdatos. Accessed 21 Feb 2024
11. Reyes, C.T., López, I.G., Clacso, C.: Políticas públicas y derecho a la educación de los pueblos indígenas en paraguay. evaluación de la implementación y alcance de las becas estatales para el acceso a la universidad (2013)
12. Snead, S.L., Walker, L., Loch, B.: Are we failing the repeating students? Characteristics associated with students who repeat first-year university mathematics. *Int. J. Math. Educ. Sci. Technol.* **53**(1), 227–239 (2022)



Effects of Language of Instruction in Higher Education

Silvia Vázquez-Noguera¹, Fabiola Martínez¹, David Becerra-Alonso²,
Aurelio López-Fernández³, Isabel Lopez-Cobo⁴, Perla Sosa⁵,
and Miguel García-Torres^{3(✉)}

¹ Universidad Americana, Asunción, Paraguay

² Department of Quantitative Methods, Universidad Loyola Andalucía, Sevilla, Spain

³ Data Science and Big Data Lab, Pablo de Olavide University, 41013 Seville, Spain

mgarciat@upo.es

⁴ Department of Didactics and Educational Organization, Faculty of Education
Sciences, University of Seville, 41013 Sevilla, Spain

⁵ Universidad Nacional de Itapúa, Encarnación, Paraguay

Abstract. In higher education, educational centers are paying attention to student mobility and internationalization to increase visibility. This increase in the intake of international students has resulted in not only improved global competitiveness, but also economic benefits. However, in order to develop education in a language different from the mother tongue, it requires high-level proficiency in such a language. Therefore, in this work, we explore the impact of the language of instruction on business administration and management grade from Universidad Loyola (Spain). We analyze and compare the student profile according to the language of instruction using a clustering approach. The results suggest that students who receive instruction in a foreign language achieve better performance than those who receive it in their mother tongue. However, the number of students who decide to study in a foreign language is smaller.

Keywords: Higher education · educational data mining · clustering

1 Introduction

In the contemporary world, mastering only one language is no longer sufficient, as the international mobility of students is increasing [5]. Therefore, learning multiple languages is becoming increasingly essential [15]. In this context, analyzing student performance according to the language of instruction and understanding the impact of the language of instruction are crucial to the success of education in a second language, especially in multilingual societies where multiple languages are used in the educational system.

The influence of instruction language on student performance has been the subject of several studies. Yang and Farley [16] explored how language impacts the performance of international accounting students in Australian universities, revealing

that students with English as a foreign language struggle more in conceptual subjects due to the cognitive load resulting from linguistic and psychological effects. Similarly, Shareef [13, 14] discovered a significant impact of instruction language on the comprehension of questions and subsequent scores. Furthermore, Purpuri et al. [10] delved into the influence of the foreign language on moral decision-making, risk aversion, and causality perception, noting that students often accept the challenge of working in a foreign language for improved scores.

The advent of information technologies in education has revolutionized data storage and analysis [11]. This technological advancement has opened avenues for evaluating the educational system in various domains, including student performance and identification of key competencies in the learning process [4]. However, the analysis of educational data requires sophisticated tools to extract meaningful insights that can be utilized to benefit students, teachers, and administrators effectively.

In this context, SAPIENS emerges as a learning tool developed within the Erasmus+ 2-year project Skill Assessment for PersonalIzed CompEtence NetworkS. The project aimed to create an innovative tool for evaluating students' skills in physics, mathematics, and statistics, enabling the identification of learning deficiencies and gaps among students. In addition, it aimed to help teachers design and implement high-quality study plans customized to their students' learning needs identified by the evaluation tool.

SAPIENS is a multilingual and accessible web tool (also available as a mobile app) that allows the management of courses in Higher Education institutions. It allows us to create competency-oriented material for a specific course, including elements such as activities. In addition, the exercises will be assigned different competencies belonging to the subject and can be corrected automatically. In this way, teachers can take advantage of the power of the tool, not only to correct activities in a more objective and faster way but also to carry out an analysis of the competencies that have been acquired or not by the students. It also allows students to comprehend complex exercises. It generates personalized exercises for each student and has a hands-on automated tutoring system that helps students improve their understanding, especially on topics related to math, science, statistics, and economics. To identify the strengths and weaknesses of the students, SAPIENS generates reports for the teacher with the learning progress of the class or of each student, as well as elaborates network representations of the competencies acquired by solving the exercises. For more information on SAPIENS, visit the project website [8].

2 Methodology

In this study, a cross-sectional, descriptive, correlational, and deductive methodology is employed, framed within the fields of Educational Data Mining (EDM) and Learning Analytics (LA), to investigate the influence of language of instruction on student performance in a Business Mathematics course. Data are collected from two groups of students who received instruction in Spanish and English during a specific academic semester at Universidad Loyola Andalucía. The methodology

focuses on describing and analyzing the characteristics and performance of the students, using basic statistics and comparative analysis between the two groups. In addition, cluster analysis, a common technique in EDM, is utilized to identify student profiles based on their performance. Based on a previous hypothesis about the influence of language of instruction, it is deduced and evaluated whether there are significant differences in student performance between the groups. This methodological approach, which combines techniques from EDM and LA, allows exploration of the relationship between language of instruction and student performance in a specific higher education context.

The use of educational data has been addressed from two different fields: EDM and LA [2]. EDM refers to the application of data mining techniques to educational datasets to address educational questions, while LA is defined as the measurement, collection, analysis, and reporting of data to track learners and their interactions of a learning ecosystem within an organization. EDM and LA are interdisciplinary areas that can be drawn as the combination of three main fields: computer science, education, and statistics. Despite the overlap between both disciplines, EDM is focused on the discovery of new patterns in data and the development of new algorithms and models.

In EDM clustering has been applied in several works [3, 7, 12]. In [1], Akhmedov aims to improve the quality of education using a clustering approach. To do so, he models the streamlining and coordination of the training of teaching staff. In another work, Moubayed et al. [9] analyze the e-learning experience of students to identify the main factors that keep students motivated and engaged. To do so, they propose using k-means to model the students. Globalization has promoted the study of foreign language learning using data mining tools [6].

In this work, we analyze the data collected from a first-year course on mathematics for business at the Universidad Loyola Andalucía. The course was taught in Spanish and the other in English. To develop the different exercises, we used the SAPIENS tool. We compare the performance between both groups and identify the different profiles of the students using a clustering approach.

2.1 Dataset

The data used come from a first-year Mathematics for Business course that was part of the fall semester academic plan for various Business degrees at Universidad Loyola Andalucía. The timeframe was September 2022 to January 2023.

Data were collected from two different groups of students: Group ES (GES) and Group EN (GEN). GES refers to those who studied the course in Spanish, while GEN refers to those who studied in English. The sample size of GES and GEN was 108 and 45 respectively.

The course covers topics related to multiple variable calculus, partial derivatives, concavity of functions, and optimization with and without restrictions. Students are required to respond to questions on these topics, directly as mathematical problems but also related to business and economics. The students used the SAPIENS platform to complete the exercises. The content of the course is divided into four units:

- U1-Var: Multiple variable functions questions aim at students having an understanding of the shape and general properties of functions represented as surfaces. The geometrical meaning of partial derivatives and the concept of gradient are covered in this part of the course.
- U2-Con: Concavity questions cover the properties of the second partial derivatives on surfaces and the detection and meaning of saddle chair points.
- U3-Opt: Optimization exercises use all prior knowledge to find local and global maxima and minima of different functions. One the business side of it, functions of income, profit, utility, etc. are maximized, while cost functions are minimized. The type of restrictions included in these exercises incorporate bounded production or limitations in investment capital.
- U4-OpR: Optimization exercises similar to previous unit but using R software.

2.2 Clustering Techniques

In clustering, let \mathcal{A} be a set of n instances described by d features X_j , $j = 1, 2, \dots, d$. Let a_{ij} denote the value of the feature X_j for instance \mathbf{a}_i . The purpose of clustering is to partition the set \mathcal{A} into p clusters C_1, C_2, \dots, C_p such that instances in the same cluster are more similar to each other than to those in other clusters.

The algorithm used in this work is the Hierarchical Clustering algorithm, which organizes a dataset into a hierarchy of clusters that do not require the number of clusters to be specified in advance. Instead, they construct the cluster hierarchy iteratively, combining or splitting groups successively. The general operation involves initializing each data point as an individual cluster, calculating the distances between pairs of clusters, identifying the closest clusters, and merging or splitting them. This process is repeated until only one cluster remains or a stop criterion is met. Hierarchical clustering algorithms produce a dendrogram, a tree-like diagram that represents the hierarchy of clusters. The height of a branch on the dendrogram indicates the distance between the clusters it merges.

Hierarchical clustering algorithms offer several advantages, such as the ability to discover clusters without specifying a number in advance, generate a hierarchical representation of the data, and be robust to the presence of outliers. In this work, we use an agglomerative approach.

3 Results

This section presents the analysis of business administration and management students and compares the performance between students who received instruction in Spanish and English. We analyze both groups of students -GES and GEN- and study whether each group of students corresponds or not to a clustering group.

3.1 Statistical Analysis

In this section, we present the analysis of the students and compare the performance of the two groups. GES for those who study in Spanish and GEN for those who study in English. Table 1 presents some basic statistics. As we can see, the number of students who study in Spanish outnumber those of GEN. However, the performance of this group is much better than that of GES, since the dropout rate of GES is around 27%, almost 9 points above GEN. In addition, the success rate of GES is around 28% which is around 19 points below the GEN.

Table 1. Statistics of the performance of the students according of the teaching language.

	Spanish	English
#Students	108	45
Dropout rate (%)	26.9	17.8
Success rate (%)	27.8	46.7

The distribution of the average score of the students according to the language of instruction is presented in Fig. 1. We can see that more students from GEN tend to achieve better average scores than students from GES.

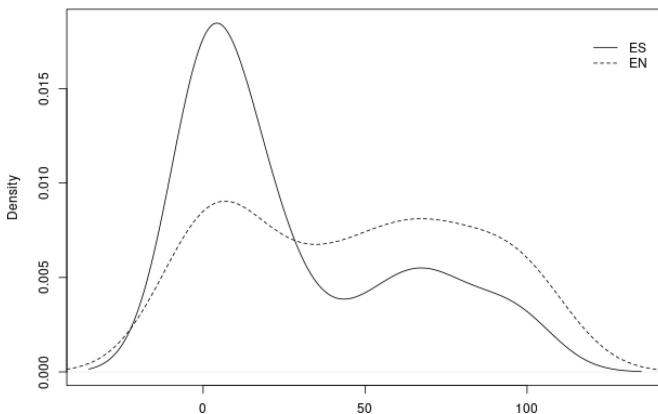


Fig. 1. Distribution of average score of students from GES and GEN.

In Table 2, the average performance of the GES and GEN students is presented in each unit. The results for each group are shown in each row. The last column refers to the average score for each group. The last row gives the average of the score on each unit. In all cases, except for unit 2, GEN outperforms GES.

Unit 2 seems to be the most difficult unit, and so, in both groups, the average is lower than in the rest of units.

Table 2. Average score of each group on each unit.

Group	U1-Var	U2-Con	U3-Opt	U4-OpR	Total
GES	5.56	5.74	6.83	6.00	5.96
GEN	7.01	4.71	7.27	6.48	6.34
Total	6.03	5.34	7.00	6.26	6.11

The trend can be observed more clearly in Fig. 2. As can be seen, the first and second units have the largest differences, on average, between the GES and GEN groups, although the slope is the opposite. For the last two units, despite the differences in the values of the average scores, the slopes are very similar.

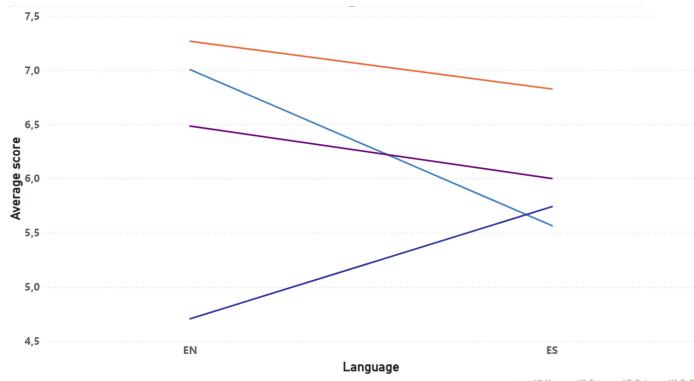


Fig. 2. Average score differences between GES and GEN groups.

Finally, Fig. 3 shows the works delivered, in percentage, per unit. As can be observed, the behavior of the students in both groups is similar. The highest dropout rates are presented after the first unit. Despite the similarities in student behavior, the dropout rate is, in all units, higher in GES than in GEN.

3.2 Students Profiles

Figure 4 shows the clustering results projected on the first two principal components. This visualization clearly shows the two groups found. However, such groups do not have much to do with the language of the student but with the

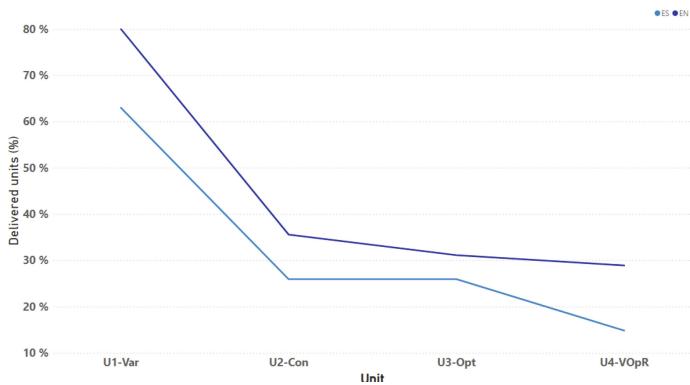


Fig. 3. Delivered units, in percentage, per unit.

performance. Because of this, we can see that for each cluster found -C1 and C2-, there are students that learn Spanish (ES) and English (EN).

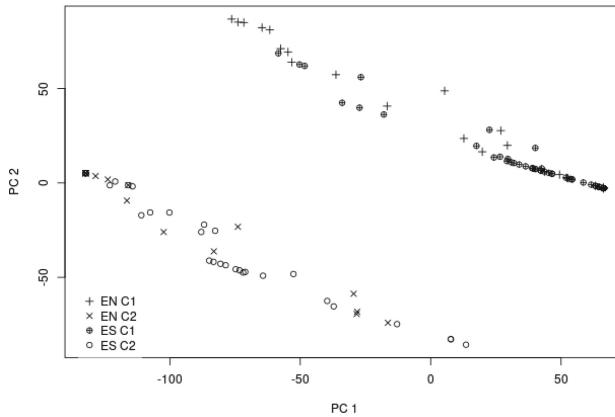


Fig. 4. Scatter plot showing the clustering results projected on the first two principal components. C_1 and c_2 refers to the cluster label while ES and EN correspond to the language in which the student learns.

The density of the two groups, according to the average score, is presented in Fig. 5. As can be seen, the profile of the students is more identified with their performance than with the language of instruction.

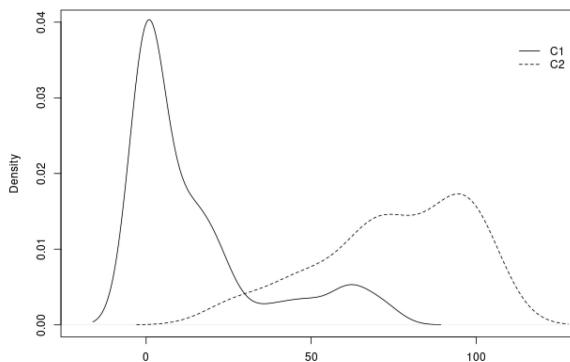


Fig. 5. Density of the average score of the students from clusters 1 and 2 identified.

3.3 Discussion

The findings of the computational analysis shed light on several key aspects regarding the impact of language of instruction on student performance in the context of business administration and management education. In particular, the statistical comparison between students instructed in Spanish (GES) and those in English (GEN) revealed intriguing patterns that warrant a deeper exploration.

- Language of Instruction vs. Performance: The stark contrast in performance between GES and GEN students raises intriguing questions about the role of language in academic success. Although one might initially assume that instruction in a student's native language would lead to better outcomes, our findings challenge this notion. It appears that factors beyond language proficiency play a significant role in student performance, as evidenced by higher dropout rates and lower success rates among GES students despite being instructed in their native language.
- Pedagogical Strategies: The disparities in performance between different units suggest potential variations in the effectiveness of the pedagogical strategies employed for each component of the course. Unit 2, for example, emerges as particularly challenging for both groups, indicating a possible need for instructional adjustments or additional support mechanisms to aid
- Student Engagement and Persistence: Dropout rates, especially after the first unit, highlight the importance of fostering student engagement and providing adequate support structures to mitigate attrition. Despite similarities in behavior patterns between GES and GEN students, the consistently higher dropout rates in the former group underscore potential barriers to academic continuity that warrant further investigation. Understanding the underlying factors driving student disengagement could inform proactive measures to improve retention and promote academic progression.

- Performance-based Student Profiles: The clustering analysis reveals that student performance, rather than language of instruction, is the primary determinant of group membership. This suggests that academic achievement transcends linguistic boundaries and underscores the importance of individual aptitude and study habits in shaping educational outcomes. By recognizing and catering to the diverse needs and abilities of students within each performance group, educators can tailor instructional approaches to maximize learning efficacy and foster academic success.
- Implications for Educational Policy and Practice: Our findings have broader implications for educational policy and practice, particularly in relation to language-inclusive pedagogical approaches and support mechanisms for diverse student populations. By acknowledging the multifaceted nature of academic success and addressing the underlying factors that influence student performance, institutions can strive to create more inclusive learning environments conducive to the holistic development of all learners.

4 Conclusions and Future Work

In this work, we have analyzed data of two different groups (GES and GEN) of students from a first-year course on Mathematics for Business. The only difference between two groups was that GES received the instructions in Spanish while GEN students in English.

The results suggest that, on average, students who decide to receive the instructions in a language other than their mother tongue achieve better scores. Furthermore, the success rate is larger and the dropout rate is lower. A possible reason for this finding is that studying in a different language of the mother tongue requires extra effort, and therefore this kind of student tends to work harder and thus achieves better scores.

Tracking the activities associated with a student allows one to identify those with a risk of dropping out. In future work, we want to expand the sample of students and include foreign students who study in Spanish and their mother tongue. We are also planning to extend this study to other courses in the humanities field.

References

1. Akhmedov, B.: Innovative cluster model for improving the quality of education. *Acad. Res. Educ. Sci.* **2**(3), 528–534 (2021)
2. Aldowah, H., Al-Samarraie, H., Fauzy, W.M.: Educational data mining and learning analytics for 21st century higher education: a review and synthesis. *Telematics Inform.* **37**, 13–49 (2019)
3. Algarni, A.: Data mining in education. *Int. J. Adv. Comput. Sci. Appl.* **7**(6) (2016)
4. Bakhshinategh, B., Zaiane, O.R., ElAtia, S., Ipperciel, D.: Educational data mining applications and tasks: a survey of the last 10 years. *Educ. Inf. Technol.* **23**(1), 537–553 (2018)

5. Bernhofer, J., Tonin, M.: The effect of the language of instruction on academic performance. *Labour Econ.* **78**, 102218 (2022)
6. Bravo-Agapito, J., Bonilla, C.F., Seoane, I.: Data mining in foreign language learning. *Wiley Interdisc. Rev.: Data Min. Knowl. Disc.* **10**(1), e1287 (2020)
7. Dol, S.M., Jawandhiya, P.: A review of data mining in education sector. *J. Eng. Educ. Transformations* **36**(Special Issue 2) (2023)
8. Loyola, U., Incoba, de Olavide, U.P., Joanneum, F., Bielefeld, U.: Skill Assessment for PersonalIsed compEtence NetworkS (SAPIENS). <https://sapiens-project.net/inicio/> (2022). Accessed 12 Jul 2022
9. Moubayed, A., Injadat, M., Shami, A., Lutfiyya, H.: Student engagement level in an e-learning environment: clustering using K-means. *Am. J. Distance Educ.* **34**(2), 137–156 (2020)
10. Purpuri, S., Vasta, N., Filippi, R., Wei, L., Mulatti, C.: Does language shape the way we think? a review of the foreign language effect across domains. *Int. J. Bilingualism*, p. 13670069231225374 (2024)
11. Romero, C., Ventura, S.: Educational data science in massive open online courses. *Wiley Interdisc. Rev.: Data Min. Knowl. Disc.* **7**(1), e1187 (2017)
12. Salloum, S.A., Alshurideh, M., Elnagar, A., Shaalan, K.: Mining in educational data: review and future directions. In: Hassanien, A.-E., Azar, A.T., Gaber, T., Oliva, D., Tolba, F.M. (eds.) *Proceedings of the International Conference on Artificial Intelligence and Computer Vision (AICV2020)*, pp. 92–102. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-44289-7_9
13. Shareef, A.: Effect of linguistic modification on the physics examination scores of international students in higher education. In: 2022 Advances in Science and Engineering Technology International Conferences (ASET), pp. 1–6. IEEE (2022)
14. Shareef, A.: Language effect on academic examination scores of international students in higher education. In: 2022 Advances in Science and Engineering Technology International Conferences (ASET), pp. 1–5. IEEE (2022)
15. Siğirtmaç, A., Özbek, S.: Teaching English in early childhood. *Inonu Univ. J. Faculty Educ. (INUJFE)* **10**(1), 107–122 (2009)
16. Yang, H.H., Farley, A.: Quantifying the impact of language on the performance of international accounting students: a cognitive load theory perspective. *Engl. Specif. Purp.* **55**, 12–24 (2019)



Analysis of Socio-Economic Factors Associated to the Academic Performance in Paraguay

Federico Javier Beck¹ and Francisco Gómez-Vela^{2(✉)}

¹ Universidad Naiconal de Concepción, Concepción, Paraguay

² Data Science and Big Data Lab, Pablo de Olavide University, 41013 Seville, Spain
fgomez@upo.es

Abstract. In Paraguay, higher education institutions often have evening classes to continue the professional development and enhance the career prospects of the students after the work. It is known that the simultaneity of study and work affects academic performance although some studies suggest a minimal influence of work on academic performance. This paper analyzes the relationship between academic success and work occupation using classification models, considering the socioeconomic data and academic records of 84 Computer Engineering students from Universidad Nacional de Concepción (Paraguay). We also analyze the different factors involved in the study. Results suggest that a small number of these factors are necessary to identify the performance of the students.

Keywords: Higher education · educational data mining · classification · feature selection

1 Introduction

The work-study phenomenon is experiencing widespread diffusion and growth in both Europe and the Americas. Data collected on the proportion of 18–24 years old combining study and employment show significant variability, both globally and within continents, according to reports by the Organisation for Economic Co-operation and Development (OECD) (2023). In Europe, for example, countries such as the Netherlands show a rate of 53%, followed by Germany with 43%, Denmark with 33%, the United Kingdom with 21%, Poland and France with 9%, Portugal with 7%, and the lowest percentage is recorded in Italy, with 3% of young people working and studying. On the American continent, the figures are 18% in the United States and Brazil, 10% in Mexico, 8% in Chile, 7% in Colombia, and 12% in Argentina. On average, according to OECD data, 18% of 18–24 year-olds combine work and study in the countries studied. In the specific case of Paraguay, the National Institute of Statistics (INE) (2022) has provided statistics based on the Permanent Household Survey, which reveal that 18% of 15–29 year-olds work and study simultaneously, which is often seen as a reason for dropping out or underachievement.

The amount of data collected today in this regard offers us new opportunities to develop more accurate institutional policies by being able to predict students' learning behaviour and type of learning [11]. Early prediction is a tool that includes analytical methods to help teachers propose appropriate actions to solve identified problems in the education of students. Thanks to this detection and taking into account not only the right data, but also the timing and relevance of the data, institutions may ensure the progress of students [3].

There are several studies on the impact of carrying out a paid job and studying simultaneously, with mixed results [6], states that exceeding 30 h of work per week has a negative influence on academic performance. The highlights presented in [5] argue that exceeding 30 h of work per week decreases academic performance by one point compared to students who do not participate in the labour market, but that the work aspect is not the only factor that demotivates the group of students. Genett in [8], agrees with these results but adds that working in career-related jobs has a positive effect on performance.

New data analysis methods (such as machine learning or data mining approaches) are being widely used for information discovery from stored data to obtain trends and patterns with the use of different classification algorithms [20]. This analysis makes it possible to anticipate events, map the most relevant variables and be able to classify students, for example, as Achieved and Not Achieved (bivariate) and subsequently validate the results once provided and take pedagogical or institutional actions to minimise the failure rate [7]. Predictive systems in the educational field that attempt to predict student performance should not necessarily be based on socio-economic data, average or final grades, but could be related to behavioural or other characteristics that can predict test scores, in which case the student's potential failure can be anticipated earlier [3].

In this paper, we present a study focused on the context of the National University of Concepción (UNC), specifically on the degree in Business Informatics Engineering, which is part of the Faculty of Economics and Administrative Sciences (FCEA). The FCEA is renowned for its evening classes, which gives students the flexibility to devote their mornings and afternoons to other activities. The aim of this paper is to carry out a predictive analysis of the academic performance of these students based on socio-economic data, with a special emphasis on those students who do paid work during their studies.

The rest of the paper is organized as follows. Section 2 introduces the main concepts related to this study and the different techniques used. Then, in Sect. 3 the analysis is presented and, finally, in Sect. 4, the conclusions.

2 Material and Methods

In this paper, we propose the use of information theory to carry out a classification process on university student data from Paraguay. Information theory, introduced by [18], provides a mathematical framework for quantifying information and understanding communication systems. Central to this theory are the concepts of entropy and information gain, which have profound applications in the field of machine learning, particularly in classification problems.

Entropy [14], in the context of information theory, is a measure of the unpredictability or uncertainty in a dataset. For a classification problem, it quantifies the impurity or disorder among the classes within a given dataset. Formally, the entropy $H(D)$ of a dataset D with n classes is defined as:

$$H(D) = - \sum_{i=1}^n p_i \log_2 p_i$$

where p_i is the probability of an instance belonging to class i . Entropy reaches its maximum when all classes are equally likely, indicating maximum uncertainty.

Another important concept is Information gain (IG) [2], which is a metric used to measure the effectiveness of a feature in reducing the entropy of a dataset. When building decision trees, for instance, information gain helps in selecting the attribute that best splits the data into subsets, thereby reducing the overall entropy. The information gain $IG(D, A)$ of an attribute A is defined as:

$$IG(D, A) = H(D) - \sum_{v \in \text{Values}(A)} \frac{|D_v|}{|D|} H(D_v)$$

where $\text{Values}(A)$ are the possible values of attribute A , D_v is the subset of D for which attribute A has value v , and $\frac{|D_v|}{|D|}$ is the proportion of instances in D that belong to subset D_v . A higher information gain indicates a more significant reduction in uncertainty, making the attribute more useful for classification.

While information gain is a powerful tool, it has a bias towards attributes with more levels. Symmetrical uncertainty (SU) [10] addresses this by normalizing the information gain, offering a balanced measure that takes into account both the entropy of the attribute and the class. Symmetrical uncertainty is defined as:

$$SU(A, C) = 2 \cdot \frac{IG(A, C)}{H(A) + H(C)}$$

where $H(A)$ is the entropy of the attribute A , and $H(C)$ is the entropy of the class C . Symmetrical uncertainty ranges between 0 and 1, where 0 indicates no association and 1 indicates a perfect correlation between the attribute and the class. This metric is particularly useful for feature selection in machine learning, ensuring a more balanced and unbiased evaluation of attribute relevance.

In this section, we present the different methods and dataset used in this paper. First, we describe the dataset and the main features of it. Secondly, we detail the feature selection process and the machine learning methods used in this work. Finally, we describe the different measures used to rate the performance of the algorithms.

2.1 Dataset Description

The data used in this work were obtained by cross-checking data from an interview with 84 students in Business Informatics Engineering from the Faculty of Economics and Administrative Sciences at Universidad Nacional de Concepción.

The survey correspond to students from different courses, of which 56% are male and 44% female. The age range is 17 to 22 years with an average age of 21.21 years. The sample is represented by all the courses of the degree course in proportion to the population of the degree course. The largest population is to be found between the first and second year, representing 59% of the sample. With regard to employment, 44% of the students work, but the proportion of 33% of the students who are looking for work opportunities stands out. Of the 37 students who said they were working, the most common reason for working among the students was to finance their studies (30%). In addition, 46% indicated that they earn less than the minimum wage. When asked whether the lack of work experience makes it difficult for university students to enter the labour market, 92% of working students consider that the lack of work experience makes it difficult for them to enter the labour market. This suggests that almost all of those who already have work experience recognise the importance of previous experience when looking for a job. Table 1 presents the most important factors involved in this work as well as the target class, the success rate of students.

Table 1. Dataset factors.

Factor	id	type
course	cu	numeric
city	ci	categorical
age	ag	numeric
sex	sx	categorical
employed before	tr	categorical
currently working	et	categorical
work	tt	categorical
salary	sa	ordinal
cause of work	mt	categorical
work hours	ht	numerical
hope finding job	et	ordinal
work type	tt	ordinal
grants	be	categorical
attendance	as	numerical
succes rate	sr	categorical

2.2 Feature Selection

Attribute selection, or feature selection, is basically focused on improving model performance and reducing computational complexity. The process aims to identify and retain the most relevant features while discarding redundant or irrelevant ones, thus addressing the curse of dimensionality and enhancing model

interpretability. Scatter Search is an evolutionary metaheuristic that has been successfully adapted to feature selection [13]. Figure 1 presents the pseudocode. It starts with a population of solutions from which a subset of solutions is selected for the reference set Ref Set, which evolves through intensification and diversification mechanisms. Solutions from this set are combined to generate new solutions that update the reference set. The combination of solutions in sparse search is guided and not random, unlike genetic algorithms. Another important difference is that the reference set that evolves is smaller than the usual populations in evolutionary algorithms. In this work we set the hyperparameters values suggested in [13].

Algorithm 1. Pseudo-code of Scatter Search for Feature Selection

Require: Dataset D , Evaluation metric M , Maximum iterations max_iter , Reference set size P_size

Ensure: Optimal feature subset S^*

- 1: Initialize reference set P with P_size diverse feature subsets
- 2: Evaluate each subset in P using metric M
- 3: Set S^* to the best subset in P
- 4: **for** iteration = 1 to max_iter **do**
- 5: Generate subset combinations from P
- 6: **for** each combination (S_i, S_j) **do**
- 7: Combine S_i and S_j to create new subset S'
- 8: Apply local search to S' to improve it
- 9: Evaluate S' using metric M
- 10: **if** $M(S')$ is better than $M(S^*)$ **then**
- 11: Update S^* to S'
- 12: **end if**
- 13: **end for**
- 14: Update reference set P with the best solutions
- 15: **end for**
- 16: **return** S^*

2.3 Machine Learning Methods

In this subsection we describe de machine learning methods used in this work. The methods are Logistic regression, support vector regression and random forest. These methods are widely used in the literature to perform classifications task for educational datasets [1, 4, 17]. The methods are describe below.

- **Logistic Regression,** [9] Linear Regression is a basic and commonly used predictive analysis method. It models the relationship between a dependent variable and one or more independent variables by fitting a linear equation to observed data. The objective is to find the best-fitting straight line through the data points that can predict future values. The treatment performed for this algorithm is that all features must be numerical.

- **Support Vector Machine (SVM)**, [15] Support Vector Machines (SVM) are powerful supervised learning models used for classification tasks. They work by finding the optimal hyperplane that maximises the margin between different classes in the feature space. SVMs are especially effective in high-dimensional spaces and are known for their robustness in handling non-linearly separable data through kernel functions (kernels).
- **Decision Trees**, [12] are a simple but versatile classification method. They work by recursively dividing the dataset into subsets based on the value of the input features, creating a decision tree model. Each node represents a feature, each branch represents a decision rule and each leaf node represents an outcome. Decision trees are easy to interpret and can handle both numerical and categorical data.
- **Naives Bayes**, Naive Bayes classifiers [16] are based on Bayes' theorem, assuming independence between features. Despite this simplifying assumption, they work surprisingly well in many real-world applications. Naive Bayes calculates the probability of each class given a set of features and selects the class with the highest probability. It is particularly effective for text classification and other problems with high-dimensional feature spaces.

2.4 Validation

As validation metrics we have selected some widely used metrics from the literature [19], which are detailed below:

- **Accuracy**: Accuracy measures the proportion of correct predictions (both true positives and true negatives) relative to all predictions made.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

- **Precision**: Precision measures the proportion of correct predictions relative to all true predictions made.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

- **Recall** (Sensitivity or True Positive Rate): Recall measures the proportion of true positive (passed) cases that the model correctly identified.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

- **F1-score**: the harmonic mean of the precision and recall.

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

3 Computational Results

This section presents the analysis of the performance of students from the computer science degree. The aims of this section can be summarized as follows:

- i. Build a classifier to predict the success rate of students based on socio-economic factors,
- ii. analyze the information that factors have about the class label, and
- iii. identify a subset of informative factors to simplify the predictive model

The experiments were conducted using stratified cross validation with $k = 10$ folds.

3.1 Model Performance

In this section we applied the classification algorithms -NB, SVM, LR and DT- to the dataset. Table 2 presents the results. First column shown the classifier. Then the accuracy followed by the precision and recall respectively. It can be seen that SVM based model is the one that achieves the best performance in all metrics. NB and DT models have a very close performance to SVM. In contrast, LR is the algorithm with the worst model.

Table 2. Performance of different classification models.

\mathcal{C}	Accuracy	Precision	Recall	F1-score
NB	81.8	0.87	0.89	0.88
SVM	83.1	0.88	0.90	0.89
LR	68.7	0.78	0.81	0.79
DT	80.7	0.86	0.89	0.87

3.2 Factors Analysis

The aim of this section is to analyze the information that the different factors have about the class label. For this experiment we use the training sample. For such purpose we present a ranking about the SU value of each factor with respect the class in Fig. 1. Those factors with $SU = 0$ were removed from the Figure.

3.3 Feature Selection

The aim of this section is to identify a subset of informative factors to build a predictive model. Table 3 presents the performance of the classifiers using the features selected by SS. As we can see, the feature selection algorithm has identified 4 features. Such features are cu, be, ex and as. SVM is again the best

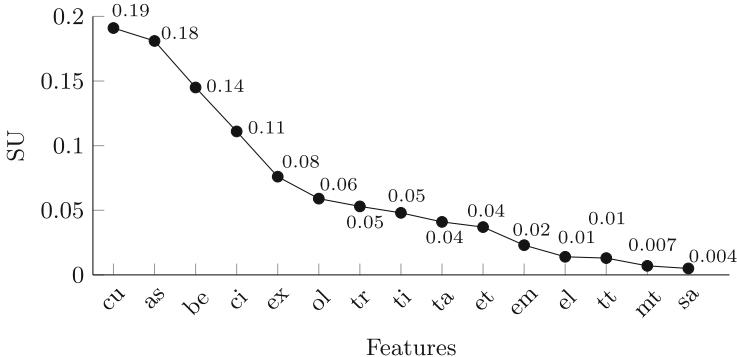


Fig. 1. Ranking of factors using SU measure and calculated over the full dataset.

classifier followed by NB. In this case LR outperforms DT. In all cases, except for LR, the performance of the classifier decreases when applying feature selection. The greatest difference was found for DT, which obtained a difference of around 4 percentage points. Then, for NB, the difference is of 3 points approximately. Finally, with SVM, the model only decreases around 1 percentage point. In contrast, for LR, the model improves the performance when applying feature selection. Such improvement is of around 8 percentage points.

Table 3. Performance of different classification models after applying feature selection.

\mathcal{A}	#Feats	acc	prec	recall	f1
SS 4					
NB	78.3	0.86	0.86	0.86	
SVM	81.9	0.86	0.93	0.89	
LR	77.1	0.82	0.89	0.85	
DT	75.9	0.82	0.87	0.84	

Finally, we analyze the subset of features identified across the cross validation. Figure 2 shows the frequency with which the different factors were selected. As it can be seen, cu and be were selected in all runs while ex in 8 out of 10 times. Then, the most selected factor was as with a frequency of 0.7. The frequency of ci and mi decreased to 0.30, and for tr to 0.20.

It is worth noting that the four features selected by SS correspond to the first four factors and that 3 out of these features -cu, as and be- correspond to the top 3 most informative factors. The remaining feature -ex- is the fifth most informative one.

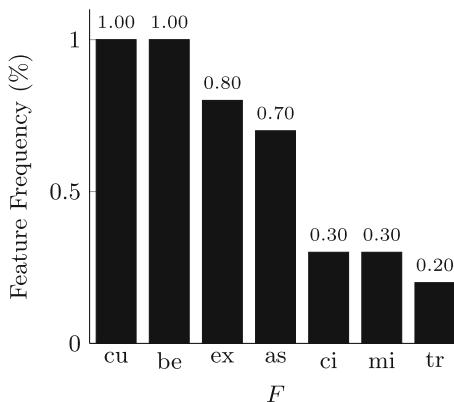


Fig. 2. Frequency of features selected by SS.

4 Conclusions

This work presents the analysis of 84 students in Business Informatics Engineering from the Faculty of Economics and Administrative Sciences at Universidad Nacional de Concepción. The aim of this study is to analyze the performance of the students based on a subset of socio-economic factors.

We first studied the predictive power of the factors studied by applying several classification algorithms -NB, SVM, LR and DT-. Results suggest that socio-economic factors allow us to predict the performance of the student with a high percentage of success ($\approx 80\%$) with most classifiers.

Then we analyzed the amount of information that each factor has about the class label (performance of the student). Results show that only a small number of factors are highly informative compared to the rest of factors. We found out that the course (cu), the attendance (as), the grant information (be) and the city (ci) are the most informative factors. This finding was supported by the feature selection process since these factors were the most selected.

It is important to highlight that the finding of this work about the importance of the factors do not imply causality. We consider that the importance of these factors are related to many other factors that require more research in order to find the causality. Future work include to extend the number of factors to improve the analysis.

Dents the flexibility to devote their mornings and afternoons to other activities. The aim of this paper is to carry out a predictive analysis of the academic performance of these students based on socio-economic data, with a special emphasis.

References

1. Ahmed, N.S., Sadiq, M.H.: Clarify of the random forest algorithm in an educational field. In: 2018 International Conference on Advanced Science and Engineering (ICOASE), pp. 179–184. IEEE (2018)
2. Azhagusundari, B., Thanamani, A.S., et al.: Feature selection based on information gain. *Int. J. Innovative Technol. Expl. Eng. (IJITEE)* **2**(2), 18–21 (2013)
3. Bernacki, M.L., Chavez, M.M., Uesbeck, P.M.: Predicting achievement and providing support before stem majors begin to fail. *Comput. Educ.* **158**, 103999 (2020)
4. Cabrera, A.F.: Logistic regression analysis in higher education: an applied perspective. *Higher Educ. Handbook Theory Res.* **10**, 225–256 (1994)
5. DeSimone, J.S.: The impact of employment during school on college student academic performance. Technical report, National Bureau of Economic Research (2008)
6. Elling, S.R., Elling, T.W.: The influence of work on college student development. *NASPA J.* **37**(2), 454–470 (2000)
7. Fernandes, E., Holanda, M., Victorino, M., Borges, V., Carvalho, R., Van Erven, G.: Educational data mining: Predictive analysis of academic performance of public school students in the capital of brazil. *J. Bus. Res.* **94**, 335–343 (2019)
8. Genett, A.: The effect of paid work on academic performance amongst UNH undergraduate students. *Perspectives* **9**(1), 7 (2017)
9. James, G., Witten, D., Hastie, T., Tibshirani, R., Taylor, J.: Linear regression. In: *An Introduction to Statistical Learning: With Applications in Python*, pp. 69–134. Springer, Cham (2023). doi: https://doi.org/10.1007/978-3-031-38747-0_3
10. Kannan, S.S., Ramaraj, N.: A novel hybrid feature selection via symmetrical uncertainty ranking based local memetic search algorithm. *Knowl.-Based Syst.* **23**(6), 580–585 (2010)
11. Kaur, P., Singh, M., Josan, G.S.: Classification and prediction based data mining algorithms to predict slow learners in education sector. *Procedia Comput. Sci.* **57**, 500–508 (2015)
12. Kothari, R., Dong, M.: Decision trees for classification: a review and some new results. In: *Pattern Recognition: From Classical to Modern Approaches*, pp. 169–184 (2001)
13. López, F.G., García-Torres, M., Batista, B.M., Pérez, J.A.M., Moreno-Vega, J.M.: Solving feature subset selection problem by a parallel scatter search. *Eur. J. Oper. Res.* **169**(2), 477–489 (2006)
14. Luukka, P.: Feature selection using fuzzy entropy measures with similarity classifier. *Expert Syst. Appl.* **38**(4), 4600–4607 (2011)
15. Mammone, A., Turchi, M., Cristianini, N.: Support vector machines. Wiley Interdiscipl. Rev. *Comput. Stat.* **1**(3), 283–289 (2009)
16. Murphy, K.P., et al.: Naive bayes classifiers. *Univ. Br. Columbia* **18**(60), 1–8 (2006)
17. Pimentel, J.S., Ospina, R., Ara, A.: Learning time acceleration in support vector regression: a case study in educational data mining. *Stats* **4**(3), 682–700 (2021)
18. Shannon, C.: Claude shannon. *Inf. Theory* **3**, 224 (1948)
19. Yacoubi, R., Axman, D.: Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In: *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pp. 79–91 (2020)
20. Yağcı, M.: Educational data mining: prediction of students' academic performance using machine learning algorithms. *Smart Learn. Environ.* **9**(1), 11 (2022)



Interactive Telegram Bot as a Support Tool for Teaching in Higher Education

Juan Luis Suárez-Díaz^(✉), Nuria Rodríguez-Barroso,
and Guillermo Gómez-Trenado

Department of Computer Science and Artificial Intelligence, Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Granada, Granada, Spain
{jlsuarezdiaz,rbnuria,guillermogomez}@ugr.es

Abstract. In this paper, we present an interactive Telegram bot designed as a support tool for the Artificial Intelligence course, developed with the intention of providing students with an innovative and accessible mechanism to complement traditional learning. This bot is capable of interpreting various commands that can be sent through Telegram, and implements several functionalities: from tips and guides for the development of practical exercises to instant feedback methods that allow students to assess their performance in the various tasks to be carried out. Although the work focuses on the Artificial Intelligence course, we discuss the potential of this tool to be extended to a wide majority of computer science subjects. This proposal aims to improve the learning experience, establishing a flexible and effective educational paradigm for various university teaching disciplines in computer science. The software used is open source and available on GitHub.

Keywords: Telegram bot · Interactive learning · Support tool

1 Introduction

The rapid advancement of technology has driven a significant transformation in teaching and learning methods in higher education. The growing availability of digital resources and the adoption of innovative technological tools have opened new opportunities to improve the quality and accessibility of education [2], especially in highly demanded and constantly evolving fields such as Artificial Intelligence.

In this context, the present work focuses on the introduction and analysis of an educational resource designed to support university-level computer science education by leveraging this technological advancement: a Telegram bot specifically designed for interaction with students in computer science courses.

Telegram is a cloud-based instant messaging application created in 2013, which has evolved in many aspects over the past decade, currently enjoying considerable popularity. Among its features is the ability to create and program

bots to engage in conversations, process the messages they receive, and craft tailored responses, which can be applied to a wide variety of problems [5, 8, 10].

In the field of computer science, there is a wide range of complex concepts and techniques that require a dynamic and adaptable educational approach to ensure student understanding and success [4]. Recognizing this need, the Telegram bot developed in this work emerges as a versatile and accessible tool that complements traditional learning by providing students with an interactive platform to explore, practice, and deepen their understanding of fundamental concepts.

This bot is not limited to simply offering static information or reading resources. It also stands out for its ability to actively interact with students, offering instant feedback, personalized guidance, and many other functionalities. In today's digital era, the use of bots is quite common on streaming or messaging platforms like Twitch or Discord, which are very popular among young audiences, the majority demographic in universities. Therefore, a Telegram bot with these characteristics is likely to be attractive to students. Telegram is a more widely known platform than the previously ones mentioned. This makes it easier to encourage more students to use it. It also offers superior privacy features, the capacity for sending large volumes of files, and a powerful API for programming bots. The conversational nature of the bot provides an intuitive and seamless experience with quick responses and visually appealing messages, and compared to traditional educational applications or platforms, Telegram offers more accessibility and convenience. This allows access to the bot from anywhere and at any time, even from mobile devices, and notifications are received instantly when the request is completed.

There is a notable amount of related work about bots or conversational agents applied to the educational field, mainly as virtual assistants [3, 11] or with integrated artificial intelligence mechanisms [1, 14]. The latter has been significantly improved in recent years thanks to generative models like ChatGPT [6]. Our proposal, instead of focusing on the conversational aspect, aims to provide students, through explicit commands via the bot, with the most useful tools and to leverage direct communication with the bot to make its use as simple, straightforward, and accessible as possible. There are proposals for Telegram bots in the educational framework, although their use generally reduces to functionalities already present in educational platforms, such as providing material or conducting multiple-choice questions on specific topics [9, 13]. In our bot proposal, in addition to the above, we highlight elements such as instant feedback, remote execution, and backups, among many others.

The developed bot is designed as a reinforcement tool for the Artificial Intelligence course at the University of Granada, although the properties discussed in this work can be abstracted and applied to a wide variety of computer science courses in the university environment. The specific implementation of the bot for the Artificial Intelligence course is open and available on GitHub¹.

The work is structured as follows: Sect. 2 presents the bot with its features and a technical description of its construction and implementation. Section 3

¹ <https://github.com/jlsuarezdiaz/IA-BOT-UGR-24>.

describes how the developed bot has been applied to the Artificial Intelligence course. Section 4 outlines the features that can be considered for designing a bot of these characteristics for any university-level computer science course. Section 5 summarizes the results of the application of the bot in the Artificial Intelligence course during the past academic year 2023–2024. Finally, Sect. 6 concludes with final observations.

2 Bot Presentation

In the following section, we present and analyze the concept of the Telegram bot. To do this, we describe the technical aspects of its internal functioning and the necessary requirements to implement a Telegram bot. Subsequently, we explain how it is used and the basic structure for interacting with it.

2.1 Description

The creation of a Telegram bot can be divided into three phases [7]: registering the bot on Telegram, implementing its software, and putting it into operation.

Registering a Telegram bot can be done easily through Telegram's own interface. Within the platform, there is a special bot called **BotFather**², with which one can start a conversation to register a new bot. The procedure is step-by-step guided, and upon completion, we will obtain the user identifier that allows communication with our bot, and a token to access the API through which the bot can send and receive messages.

At the software level, there are several libraries that allow us to communicate with the aforementioned API using a programming language. In our case, we have chosen to use the Python library `python-telegram-bot` [12]. This library provides the necessary interface to both set the bot to listen for incoming messages and to send text messages and multimedia content of the types supported by Telegram. In this way, requests received by the bot can be processed in Python, and an appropriate response can be crafted and sent to the user through the interface.

Finally, to put the bot into operation, it is enough to link the developed software with the bot token obtained during registration, which can also be done through the software interface. Lastly, the software needs to be run on a machine with internet access.

2.2 Bot Interface

Once created and put into operation, it is possible to contact the bot through the Telegram chat as if it were any other conversation. Access is through its user identifier, which we obtained from the previous process. Once in the conversation, we can communicate with the bot through the various programmed

² <https://t.me/BotFather> (@BotFather).

commands. Initially, it is common to include introductory commands, such as `/start` or `/help`, which then prompt the bot to provide information about the other commands. In the case of our bot, the initial conversation prompts the user to register. Figure 1a shows the initial conversation.

Regarding the commands that can be programmed, the possibilities are limitless and can be adapted to the requirements of different courses. In Sects. 3 and 4, we describe the functionalities specifically programmed for the Artificial Intelligence course and identify generic concepts for applying a similar bot in other computer science courses, respectively.

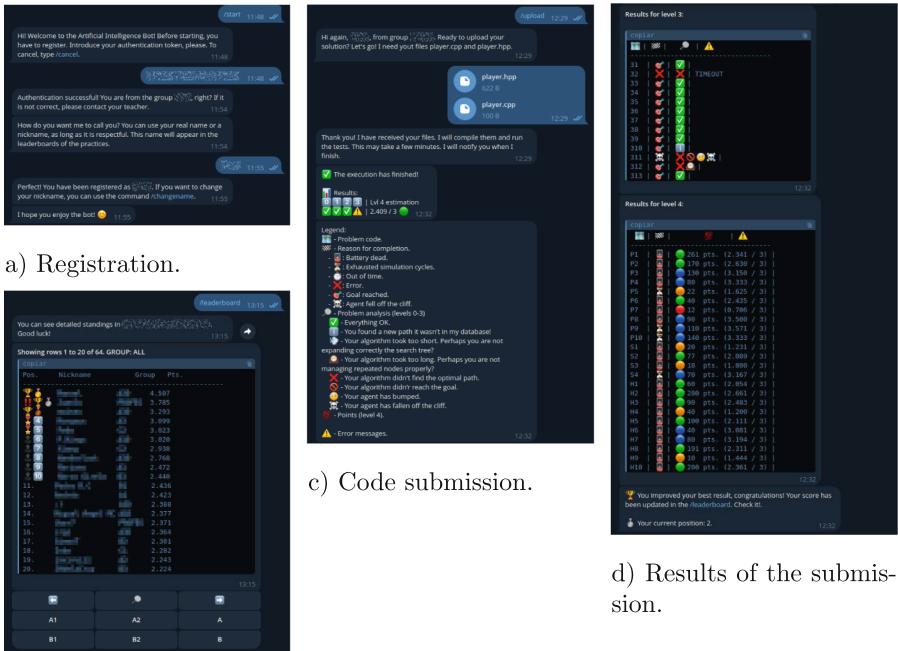


Fig. 1. Examples of conversations with the Artificial Intelligence Bot. In a), the users register in the bot to check if they belong to the course. In b), one of the practice leaderboards is queried. Conversations c) and d) show the process of uploading a code to the bot and retrieving the results of the executions. More examples and detailed explanations can be found at <https://github.com/jlsuarezdia/IA-BOT-UGR-24#Examples>.

3 Application of the Bot to the Artificial Intelligence Course

In this section, we describe how the developed Telegram bot was used for the Artificial Intelligence course practices during the 2023/2024 academic year at

the University of Granada. The implemented bot was used throughout the three course practices, offering different functionalities for each. The three practices, described below, address the problems of path-finding with reactive-deliberative agents and game solving using adversarial search techniques.

3.1 Reactive and Deliberative Agents

The objective of the first two practices is to make students become familiar with agents moving through a closed world and with path-finding algorithms. For the first practice, the goal is to define rules for a reactive agent to discover most of the world in which it operates. The objective for the second practice is to implement various path-finding algorithms in the same software. The practices consist of several levels, incremental in difficulty. For the first practice, each level introduces new obstacles for the reactive agent. For the second practice, a new path-finding algorithm is proposed to be developed for each of the first levels, assuming that the agent is purely deliberative: it knows the entire map surrounding it from the beginning. Finally, there is a special level where the agent does not know the map and must discover the terrain on its own while reaching certain goals that allow it to score points. The objective in this level is to maximize this score within a certain period of time.

In these two practices, the bot was deployed on a university cluster with multiple computing nodes. Through these nodes, the bot could send additional jobs according to the requests it received. The functionality implemented for this practice allowed students to evaluate the discovered map percentage in the first practice, the correctness of the search algorithms they were implementing in the initial levels of the second practice, and automatically obtain the score achieved on different maps in the special level. The functionalities, which are mostly common for both practices, are detailed below:

- **Remote execution and validation of algorithms.** Through this functionality, students could upload their implemented code files to the bot, which would then compile and execute them. The codes are run in a common server different from the computers of the students. The codes to be run are stored in folders that can only be accessed by the bot itself and by the teachers of the practice. Executions were performed on different tests that allowed assessing the correctness of the implemented algorithms. After the execution, the bot returns a summary indicating if the algorithm was correctly implemented, and suggest possible problems of the implementation.
- **Test visualization.** When the bot receives a command of this type, it responds to the user with the requested test, attaching a picture of the map. In the second practice, it also includes the expected results in terms of consumption and path length, and also provides a capture of a possible path that should have been obtained with the algorithm.
- **Assessment of map discovery in a wide variety of tests.** For the reactive agents practice, the bot runs hundreds of tests on lots of different maps

and under a broad range of initial conditions. Then, it returns the percentages obtained on each of the tests, so that the students can easily detect the main reasons that may cause their agents to get stuck. The bot also provides a general leaderboard with all the students who wish to participate, so that the participants can know how their solution compares to the rest of the participants in the course.

- **Execution of the special level.** Similarly, in the special level of the second practice, the bot passed the code through several maps, some of which were unknown to the students, to offer them an estimation of the exploration capacity of the implemented behavior. For this level, a new leaderboard was provided as well for those who desired to participate.
- **History and retrieval of previous solutions.** The solutions uploaded to the bot are stored, and students can consult them at any time, and even ask the bot to resend the code associated with a particular solution if they wish to retrieve it.

Figure 1 shows examples of the conversation with the bot for the main functionalities developed for this block of the subject. The conversations are focused on the second practice.

3.2 Adversarial Search and Games

The objective of this practice is for students to learn to develop adversarial search algorithms, such as alpha-beta pruning, and apply this knowledge to develop a heuristic for a game consisting of an adaptation of Parcheesi where the randomness of the dice is removed.

In this case, goal is to defeat three predefined opponents, which are artificial intelligences designed for this game, progressively increasing in difficulty. In this practice, the bot was redeployed on the previously mentioned cluster. The functionality implemented for this practice is similar to that of the previous practice, allowing validation of the correctness of the implemented adversarial search algorithm, remote execution of games, and participation in a tournament among the students.

- **Test visualization.** Similar to the previous practice, this command allows students to check if they have implemented their algorithm correctly. For this purpose, they are shown the result they should obtain when playing certain predefined games.
- **Remote execution.** Although opponents were available to play against over the network, the bot allowed these games to be played remotely through computing nodes to speed up the games and also allow parallel executions, as these games could take a long time. As the games finished, the bot responded with the result.
- **Participation in tournaments.** Students who wished to do so could upload their solution and participate in a deterministic Parcheesi competition with the developed algorithm and heuristic. This competition had a purely recreational purpose and was separate from the practice evaluation. Every day,

the bot internally ran a tournament where the algorithms of those who had decided to participate faced off, and it compiled a classification with the final results. This tournament saw a significant increase in participation and interest in the practice.

4 Generalizing the Bot to University Computer Science Education

Although the previous section shows the specific application of a Telegram bot to the Artificial Intelligence course, the implemented functionalities can be applied in many areas of university computer science education in general. In this section, we abstract and discuss these functionalities, as well as highlighting additional functionalities that could be included in a bot of this nature.

- **Support materials.** The student-bot interaction can be used for the bot to provide answers to common questions that students may have. The responses can be textual or multimedia material, providing a greater explanatory capacity. Likewise, responses can be oriented towards specific course objectives and be expressed in the form of tests, hints, or tips.
- **Remote execution.** If computing nodes are available, the bot can be configured to execute tasks or code received from students and provide feedback results on those tasks. This reduces the computational burden on student end and levels the playing field by providing an execution environment under the same hardware conditions for everyone.
- **Instant feedback.** Whether through previous remote execution or simply by calculating metrics on the results, the bot can provide instant feedback to student work at any time, thus promoting their motivation with the course and active participation.
- **Tournaments or rankings to encourage participation and motivation.** When the task allows it, a bot with the described characteristics allows for the organization of tournaments or competitions among students who wish to participate. This can make tasks more engaging and provide extra motivation to complete them. Competitiveness, considered purely for recreational purposes and outside the evaluative part of the course, can contribute to pushing the work done to the limit and improving performance and learning.
- **Backup of work.** A bot with the described characteristics in this work allows for storing copies of all work or code sent to it. Thus, in case of loss or erroneous modification of any file by a student, it is possible to recover a correct version of that file if it had been previously sent to the bot.
- **Automatic software grading.** Just as it can provide feedback on the work sent by students, the bot can also be used to automatically evaluate code snippets, check for compilation and execution errors, and validate that it works correctly by adding appropriate tests to the bot.

5 Usage Statistics

The following section presents some usage statistics extracted from the use of the bot by students during its application in the Artificial Intelligence course mentioned in Sect. 3. The collected statistics allow us to analyze how the use of the bot positively influenced both participation and grades obtained in the practices where it was applied.

Firstly, Table 1 presents some general usage statistics of the bot by students. Out of the total 395 students across all groups who took the course in the academic year 2023–2024, the bot was used at some point in each of the practices by about 69–76 % of the students. The massive amount of code uploads, with more than 27,000 uploads and an average of 85 per student also indicates the positive reception of this bot by them. In Fig. 2a we show several stats about participation and average grades. We can see here significant improvements in the average grades when using the bot, both in participation and performance, especially in those who participate in the leaderboard of the second and third practices.

Table 1. Bot usage stats of the course

	Practice 1	Practice 2	Practice 3	Total
Total students in the subject	-	-	-	395
Students presenting the practice	334	276	304	-
Students using the bot	301	284	273	-
Students presenting the practice and using the bot	292	253	273	-
Students participating in the bot leaderboards	301	59	116	-
Total code submissions	12963	6859	7589	27411
Average submissions per student	43.06	24.15	27.79	84.60

Lastly, it was also observed that student performance increased proportionally to the feedback provided by the bot. To examine this, we observed how the average grade obtained in the practices varied as the number of uploads to the bot by each student increased. Figure 2 also depicts this evolution, including the number of students who made more than n uploads to the bot, with n increasing in increments of 5, and the corresponding average grade of that group of students. These graphs show a clear trend of growth in the grade as the number of students participating in the calculation of the average is high enough. Once the number decreases, some variability is observed. In the second practice, the growth trend is more pronounced, with the average grade increasing from less than 6 among the total number of students making 5 or more uploads to the bot to an average grade of 8.5 among students making more than 80 uploads to the bot. From these results, two different interpretations can be drawn, which are not necessarily exclusive: the bot may reflect the motivation of the students

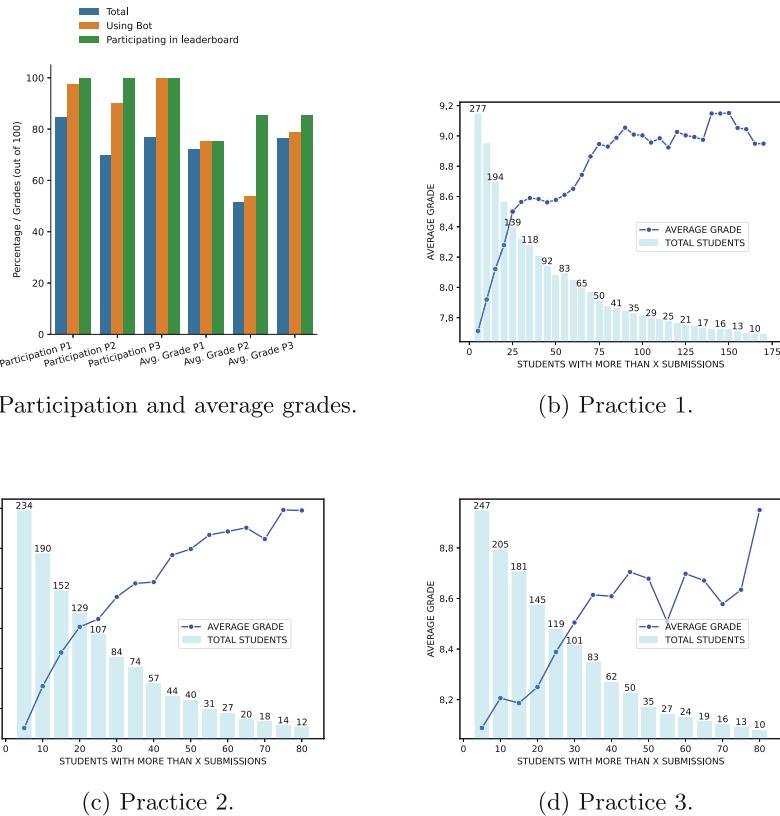


Fig. 2. General participation stats and evolution of average grades according to number of submissions per practice.

and/or the facilities provided by the bot may influence their motivation and performance.

6 Conclusions

In this work, we have presented a methodology to enhance learning through Telegram bots, enabling conversations to reinforce concepts and obtain feedback and advice on the development of assignments. The methodology has been applied in the Artificial Intelligence course with quite promising results in terms of participation and performance of the students.

At a general level, the use of bots can be extrapolated to most subjects in university-level computer science education, including functionalities as described in this document. In an environment that offers unlimited programming possibilities, expanding this methodology to other areas of knowledge could be equally beneficial. Furthermore, it would be interesting to explore the

integration of additional functionalities in bots, such as automatic generation of personalized exercises, or group study sessions, leveraging Telegram chats. These future lines of research could contribute to improving the educational experience of students.

References

1. Artiles Rodríguez, J., Guerra Santana, M., Aguiar Perera, M.V., Rodríguez Pulido, J.: Agente conversacional virtual: la inteligencia artificial para el aprendizaje autónomo. *Pixel-Bit: Revista de Medios y Educación* **62**, 107–144 (2021)
2. Domingo-Coscollola, M., Bosco-Paniagua, A., Carrasco-Segovia, S., Sánchez-Valero, J.A.: Fomentando la competencia digital docente en la universidad: Percepción de estudiantes y docentes. *Revista de Investigación Educativa* **38**(1), 167–182 (2020)
3. González-Moreno, J.C., Ramos-Valcárcel, D.: Integrando agentes conversacionales en la web 2.0 como asistentes personales. *Actas de las Jornadas de Enseñanza Universitaria de Informática, JENUI 2008* (2008)
4. Hazzan, O., Lapidot, T., Ragonis, N.: Guide to Teaching Computer Science. Springer, Cham (2020). <https://doi.org/10.1007/978-3-030-39360-1>
5. Korotaeva, D., Khlopotov, M., Makarenko, A., Chikshova, E., Startseva, N., Cheymysheva, A.: Botanicum: a telegram bot for tree classification. In: 2018 22nd Conference of Open Innovations Association (FRUCT), pp. 88–93. IEEE (2018)
6. Labadze, L., Grigolia, M., Machaidze, L.: Role of AI chatbots in education: systematic literature review. *Int. J. Educ. Technol. High. Educ.* **20**(1), 56 (2023)
7. Modrzyk, N.: Building telegram bots: develop bots in 12 programming languages using the telegram bot API. Apress (2018)
8. de Oliveira, J.C., Santos, D.H., Neto, M.P.: Chatting with arduino platform through telegram bot. In: 2016 IEEE International Symposium on Consumer Electronics (ISCE), pp. 131–132. IEEE (2016)
9. Pereira, J.: Leveraging chatbots to improve self-guided learning through conversational quizzes. In: Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality, pp. 911–918 (2016)
10. Setiaji, H., Paputungan, I.V.: Design of telegram bots for campus information sharing. In: IOP Conference Series: Materials Science and Engineering, vol. 325, p. 012005. IOP Publishing (2018)
11. Tamayo, P.A., Herrero, A., Martín, J., Navarro, C., Tránchez, J.M.: Design of a chatbot as a distance learning assistant. *Open Praxis* **12**(1), 145–153 (2020)
12. Toledo, L.: Python-telegram-bot documentation (2023)
13. Tsidylo, I., Samborskiy, S., Mazur, S.I., Zamoroz, M.: Designing a chat bot for learning a subject in a telegram messenger. In: ICTERI Workshops, pp. 1329–1340 (2020)
14. Wehr, Y.E.L., Baluis, W.L.R.: Chatbot basado en inteligencia artificial para la educación escolar. *Horizontes Revista de Investigación en Ciencias de la Educación* **7**(29), 1580–1592 (2023)

Author Index

A

- Alaiz-Moretón, Héctor 247
Alonso-Diez, Pablo 279
AlOtaibi, Nourah 201
Álvarez-Aparicio, Claudia 25, 157, 212
Álvarez-Crespo, Marta-María 257
Analide, Cesar 135
Andrysiak, Tomasz 35
Aramburu, Eduardo Jorge Sanjurjo 145
Araújo, Madalena 279
Aveleria-Mata, Jose 247

B

- Barbosa, Jorge 288
Basurto, Nuño 237
Bayón-Gutiérrez, Martín 247
Bayón-Martínez, Raúl 157
Becerra-Alonso, David 332
Beck, Federico 312
Beck, Federico Javier 322, 342
Bouma, Debby 98

C

- Caballero-Gil, Pino 191
Cabeza-Lopez, Eduardo Manuel 227
Calvo-Rolle, José Luis 257
Cambra, Carlos 237
Campazas-Vega, Adrián 25, 157, 179, 212
Carmona del Jesús, Cristóbal José 301
Casado-Vara, Roberto 237
Castillo Campo, Óscar 168
CheSuh, Lawrence Nforh 247
Costa, Antonio 135
Costa, Tiago 135
Cunha, Jose 135
Curiel-Herrera, Leticia Elena 227

D

- de la Puente Secades, Francisco Javier 212
de la Puerta, José Gaviria 145
De la Torre Vico, Rubén 122

- del Pozo, Arantza 88
Delinschi, Daniela 112
Diaz-Gorrin, Jackson 191
Díaz-Longueira, Antonio 257
Diez, Xabier Ceballos 145
Díez-González, Javier 279

E

- Enrique-Sánchez, Enzo M. 312
Erdei, Rudolf 112
Esteban-Costales, Gonzalo 157, 179, 212

F

- Felemban, Muhamad 201
Fernández-Díaz, Ramón Ángel 247
Ferrero-Guillén, Rubén 279
Fontenla-Romero, Óscar 257
Francisco Borja, Garnelo Del Río 66

G

- Gamiz, Idoia 3
García Lleyda, David 168
García Vico, Ángel Miguel 301
García-Torres, Miguel 312, 332
Gayoso Martínez, Víctor 168
Gómez-Trenado, Guillermo 352
Gómez-Vela, Francisco 342
Gómez-Vela, Francisco A. 322
González-Cantón, Alejandro 77
González-Fernández, Irene 25
González-Fernández, Rodrigo 179
González-Santamaría, Miguel A. 77
González-Santamaría, Miguel Ángel 25
Guardiola, Gorka 77
Guerrero-Higueras, Ángel Manuel 25, 157, 179, 212

H

- Hernández Encinas, Luis 168
Herrero, Alvaro 237
Hoffmans, Charlotte 98

I

- Insfrán-Coronel, Diego R. 312
Inyesto-Alonso, Laura 157

J

- Jacob, Eduardo 3

L

- Lage, Oscar 3
Lera, Francisco J. Rodríguez 66
Llamas, Camino Fernández 66
López-Cobo, Isabel 332
López-Fernández, Aurelio 312, 322, 332
Luque-Baena, Rafael M. 269
Luque-Polo, Gabriel 269

M

- Magán-Carrión, Roberto 122
Mahmood, Sajjad 201
Martín Muñoz, Agustín 168
Martín, Sergio Rubio 247
Martínez, Fabiola 332
Martínez-Gutiérrez, Alberto 279
Matei, Oliviu 112
Matellán-Olivera, Vicente 25, 157, 179, 212
Merino-Gomez, Alejandro 227
Michelena, Álvaro 257
Miguel-Diez, Alberto 179
Moreno, Mikel 88

N

- Navarro, Adrián Martínez 145

O

- Olivera, Vicente Matellán 66

P

- Pasca, Emil Marian 112
Pastor, Iker 88
Pastor-López, Iker 145
Perez, Hilde 279
Pungilă, Ciprian 15
Pungila, Ciprian 56

R

- Regueiro, Cristina 3
Requejo, Iker González 145

Rincon, Jaime Andres 227

Rodriguez Lera, Francisco J. 77

Rodríguez-Barroso, Nuria 352

Rodríguez-Gómez, Rafael Alejandro 122

Rodríguez-Lera, Francisco Javier 25

Roelofsma, Peter 98

Ruiz-Gonzalez, Ruben 227

S

- Saganowski, Łukasz 35
Sainz-Villegas, Leticia 237
Sánchez-Calleja, Iván 279
Segurola, Lander 88
Sobrín-Hidalgo, David 25
Soriano, Enrique 77
Sosa, Perla 332
Stratan, Lia 56
Suárez-Díaz, Juan Luis 352
Subirats, José Luis 269

T

- Tonu, Rebeca 15
Torres-Báez, José A. 322
Torres-Báez, Juan B. 322
Treur, Jan 98

U

- Unzilla, Juanjo 3
Urda, Daniel 237

V

- Valdivia Alcalá, Diego Ismael 301
van den Hout, Niek Jan 98
Vázquez-Noguera, Silvia 332
Vega-González, Christian 179

Y

- Yayla, Oğuz 45
Yılmaz, Yunus Emre 45

Z

- Zayas-Gato, Francisco 257
Zola, Francesco 88
Zwarts, Natalia 98