

ISSN (Print): 2077-7973
ISSN (Online): 2077-8767
DOI: 10.6977/IJoSI.202504_9(2)

International Journal of Systematic Innovation



VOL. 09 NO. 02
April, 2025

Published by the Society of Systematic Innovation

***Opportunity Identification
&
Problem Solving***

The International Journal of Systematic Innovation

Publisher:

The Society of Systematic Innovation

Editorial Team:

Editor-in-Chief:

Sheu, Dongliang Daniel (National Tsing Hua University, Taiwan)

Executive Editor:

Deng, Jyhjeng (DaYeh University, Taiwan)

Yeh, W. C. (National Tsing Hua University, Taiwan)

Editorial Team Members (in alphabetical order):

- Cavallucci, Denis (INSA Strasbourg University, France)
- De Guio, Roland (INSA Strasbourg University, France)
- Feygenson, Oleg (Algorithm Technology Research Center, Russian Federation)
- Filmore, Paul (University of Plymouth, UK)
- Kusiak, Andrew (University of Iowa, USA)
- Lee, Jay (University of Cincinnati, USA)
- Litvin, Simon (GEN TRIZ, USA)
- Lu, Stephen (University of Southern California, USA)
- Mann, Darrell (Ideal Final Result, Inc., UK)

- Sawaguch, Manabu (Waseda University, Japan)
- Shouchkov, Valeri (ICG Training& Consulting, Netherlands)
- Song, Yong-Won (Korea Polytechnic University, Korea)
- Yu, Oliver (San Jose State University, USA)
- Zhang, Zhinan (Shanghai Jiao Tong University)

Managing Editor:

- Jimmy Li

Assistant Editors:

- Chiaoling Ni

Editorial Office:

- The International Journal of Systematic Innovation
- 6F, # 352, Sec. 2, Guanfu Rd, Hsinchu, Taiwan, R.O.C., 30071
- e-mail: editor@systematic-innovation.org
ijosi@systematic-innovation.org
- web site: <http://www.IJoSI.org>
- Tel: +886-3572-3200

INTERNATIONAL JOURNAL OF SYSTEMATIC INNOVATION

CONTENTS

APRIL 2025 VOLUME 9 ISSUE 2

FULL PAPERS

- A systematic approach to corporate innovation excellence
..... K. Altun. 1–13
- Structure learning of Bayesian networks using sparrow optimization algorithm
..... S.W. Kareem, H.Q. Awla, A.S. Mohammed. 14–25
- Secure mobile cloud data using federated learning and blockchain technology
..... G. Matheen Fathima, L. Shakkeera, Y. Sharmasth Vali. 26–36
- A comparative study of traditional machine learning models and the KNN-KFSC method for
optimizing anomaly detection in VANETs
..... R. Ch, D. Kavitha, S. Sowjanya C., S. Pallavi, V. Ramesh. 37–46
- Hybrid prediction model by integrating machine learning techniques with MLOps
..... P. Narang, P. Mittal, Nisha. 47–59
- Measuring the accuracy of time series reduction methods based on modified dynamic time
warping distance calculations
..... A. Jawale, A.K. Tripathy. 60–73
- Optimizing cloud-based intrusion detection systems through hybrid data sampling and feature
selection for enhanced anomaly detection
..... S. Viharika, N.A. Balaji. 74–84
- YOLOXpress: A lightweight real-time unmanned aerial vehicle detection algorithm
..... N.T. Tai, B. DucThang, N.N. Hung. 85–95
- A blockchain-based solution to combating identity crime and credit card application fraud using
data mining algorithms
..... A.J. Shakadwipi, D.C. Jain, S. Nagini. 96–104

A systematic approach to corporate innovation excellence

Koray Altun^{1,2*}

¹Department of Industrial Engineering, Bursa Technical University, Bursa, Turkey

²ARDiMER Digital Innovation Limited, Bursa, Turkey

*Corresponding author E-mail: koray.altun@btu.edu.tr

(Received 19 September 2024; Final version received 12 November 2024; Accepted 13 December 2024)

Abstract

While international standards on innovation management have gained interest, “excellence” in innovation management has not been thoroughly studied in the literature. To address this gap, this study proposes the “Innovation Excellence Model” for corporate innovation. This approach aims to provide a concise way of excellence in corporate innovation system design. This model focuses on three important components of the system: innovation execution system, innovation organization, and innovation engine. This model is based on three different innovation engines (idea-driven, analysis-driven, and research-driven) and proposes a card-based control system to balance workload and project flows. The integration of card-based control and its simulated case provides a tangible and effective means of translating theoretical concepts into practical execution. A novel key performance indicator, “CIP – Corporate Innovation Performance” is also introduced for monitoring the excellence degree. By fostering a holistic understanding of excellence in corporate innovation, the model enables organizations to navigate the design of innovation management system, propelling them toward excellence and growth.

Keywords: Corporate Innovation, Excellence Model, Innovation Management

1. Introduction

Excellence models and standards are two different approaches that organizations can use to improve their performance and achieve their goals. An “excellence model” can be considered a framework used to assess and improve organizational performance. It is typically a systematic approach that defines key areas of focus and outlines specific practices and behaviors that are associated with high levels of organizational performance (Mann and Grigg, 2004; Mohammad et al., 2011). Organizations that use an excellence model typically strive for “excellence” and seek to exceed minimum requirements.

On the other hand, “standards” focus on meeting minimum requirements. Excellence models are typically more comprehensive than standards, covering a wider range of performance areas and providing more detailed guidance on best practices.

In recent years, there has been a growing emphasis on the importance of adhering to international standards on innovation management (Hyland &

Karlsson, 2021). However, despite this trend, the topic of achieving excellence in innovation management has not been explored with the same level of rigor and comprehensiveness in academic literature.

Although there have been some initial efforts to tackle this issue, the field remains relatively new and uncharted in academic literature. In light of this gap, and with the aim of surpassing existing standards, this study puts forward a novel approach to corporate innovation: the “Innovation Excellence Model (IEM).”

IEM aims to provide a more comprehensive and systematic guide for organizations seeking to achieve excellence in their innovation practices. It aims to be a groundbreaking framework to achieve innovation excellence in corporate settings, which centers around three crucial components: the innovation execution system, innovation organization, and innovation engine.

IEM is anchored on three distinct innovation engines: idea-driven, analysis-driven, and research-driven, each emphasizing different approaches to innovation. It further puts forth a card-based control

system that promotes a balanced distribution of workloads and project flows, fostering seamless collaboration and efficient resource allocation.

In addition, the Corporate Innovation Performance (CIP) is proposed as a novel key performance indicator (KPI) enabling organizations to monitor their progress and level of excellence. This approach aims to simplify the process of achieving innovation excellence by providing a clear and visually appealing roadmap, facilitating organizations' ability to cultivate a culture of innovation and generate meaningful outcomes.

The remainder of this study is structured as follows. Section 2 presents an in-depth literature review on the concept of "excellence" in the context of innovation management, examining existing research and identifying gaps in the literature. In Section 3, the IEM is introduced, providing a comprehensive overview of the framework and outlining its key dimensions and components. Section 4 highlights a novel KPI specifically designed for corporate innovation, offering a reliable and effective means of monitoring and assessing an organization's innovation excellence level. Section 5 discusses how to balance the innovation engines. Finally, concluding remarks are presented in the final section.

2. Literature Review

To gather information on previous attempts to explore the concept of excellence in innovation, a thorough search of the Thomson Reuters' Web of Science/Knowledge database was conducted. Specifically, papers containing the terms "excellence in innovation" or "innovation excellence" in their titles, abstracts, or keywords were retrieved from the database.

A total of 47 publications were identified, with an h-index of 9 and a cumulative number of times cited reaching 364. Fig. 1 depicts the sum of times cited per year, revealing a steady increase in interest in this area over time.

Prior work on this topic remains relatively limited. This suggests that although interest in the topic has fluctuated over time, there is still a need for further research and development in the field.

Table 1 offers a list of the most frequently cited papers on this subject, providing a valuable resource for researchers seeking to delve deeper into this field of inquiry.

Dervitsiotis (2010) explored the potential of an "innovation excellence model" to enhance innovation performance in organizations, emphasizing the importance of leadership and culture in driving innovation.

Mele and Colurcio (2006) proposed a framework for measuring innovation excellence in the service

Table 1. A summary table of the literature

Year	Authors	Total citations
2010	Dervitsiotis, Kostas N.	46
2006	Mele, Cristina; Colurcio, Maria	44
2007	Martensen, Anne; Dahlgaard, Jens J.; Park-Dahlgaard, Su Mi; et al.	43
2016	Lee, Youngsu; Rim Suk-Chul	32
2009	Kimiloglu, Hande; Zarali, Hulya	25
2008	Dahlgaard-Park, Su Mi; Dahlgaard, Jens, J.	17

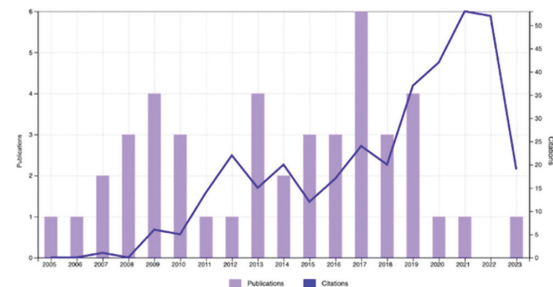


Fig. 1. Times cited and publication over time

sector, focusing on the integration of customer feedback and employee involvement in the innovation process.

Martensen et al. (2007) investigated the impact of ISO 9000 certification on innovation performance in small- and medium-sized enterprises (SMEs), highlighting the positive influence of ISO 9000 on innovation capacity and organizational learning.

Lee and Rim (2016) explored the effect of government funding on innovation excellence in South Korean SMEs, emphasizing the importance of strategic planning and risk management in leveraging public funds for innovation.

Kimiloglu and Zarali (2009) examined the role of leadership and organizational culture in fostering innovation excellence in Turkish companies, identifying a strong positive correlation between leadership style, organizational culture, and innovation performance.

Upon reviewing the retrieved papers, it becomes evident that there is only one existing model for innovation excellence in the literature, which is the "4P model" proposed by Dahlgaard-Park and Dahlgaard (2008).

The 4P model is a customized version of the EFQM Excellence Model, which is widely used in business excellence frameworks. While the EFQM model has four result factors, including Customer Results, Employee Results, Society Results, and Key Performance Results, the 4P model has only one result factor, which is Innovation Results. In addition, the 4P

model has two additional enablers' factors, which are Customer Orientation and Innovativeness.

The authors of the 4P model emphasize the importance of leadership, people, and partnerships as the key areas that companies should improve before focusing on process improvement (Dahlgaard-Park and Dahlgaard, 2008).

Dahlgaard et al. (2013) conducted a study where they reviewed various business excellence models and discussed their limitations, implications, and further development. They also considered the 4P excellence model as a simplified version of business excellence model. Although the 4P model was initially introduced as an IEM, it has undergone significant development and transformation. As a result of this evolution, the BEF: Business Excellence Framework has emerged as a new and advanced version of the 4P model. However, it is worth noting that the BEF is no longer an IEM but rather a comprehensive business excellence model.

Overall, the literature review suggests that research in this area remains relatively limited. Further exploration of this topic could help organizations develop more effective innovation management systems and enhance their overall innovation performance.

3. Proposed Approach

A robust innovation management system incorporates various components and processes to facilitate innovation throughout the organization. However, not all components of an innovation management system are equally important, and some are more critical than others in contributing to the functionality of the overall system. As such, it is essential to identify and prioritize these critical components during the setup phase of the innovation management system.

The IMP³ROVE – European Innovation Management Academy has developed an online benchmarking tool. It is based on A.T. Kearney's House of Innovation model (Diedrichs et al., 2006). A well-structured and reasonable list of innovation management components can be derived from this tool (Fig. 2).

The "IEM – Innovation Excellence Model" proposed in this study places emphasis on the most crucial components of a typical corporate innovation system during its "implementation/setup" phase. These components contribute significantly to the overall functionality of the system.

The model prioritizes three key components that work together synergistically:

- (i) Innovation execution system
- (ii) Back-end of innovation
- (iii) Innovation engines.

Lercher (2020) brings attention to a central concern; existing innovation models fall short in

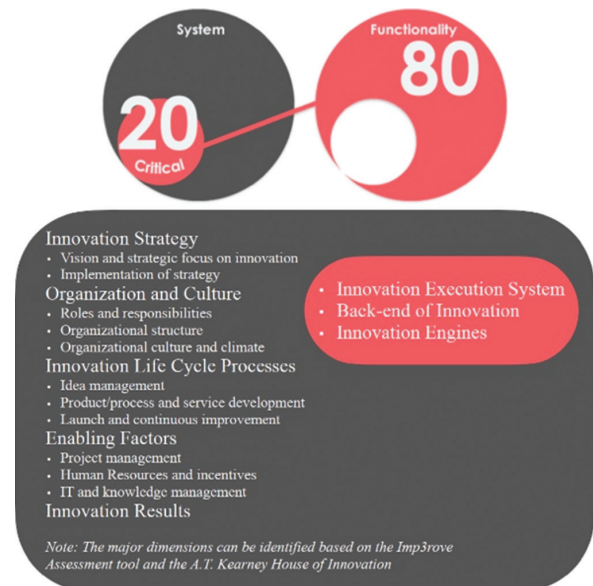


Fig. 2. Critical components of corporate innovation

adequately addressing innovation management within companies and its integration with corporate strategy to the extent required for effective entrepreneurial action and the comprehensive exploration of all innovative opportunities. In addition, these models often lack a practical orientation, encompassing only select aspects of the innovation process. It is important to underscore the particular emphasis on the deficiency of real-world orientation within this discourse.

The "Big Picture" model (Lercher, 2020) stands out as a distinctive example among rare frameworks, having been developed with a deliberate focus on real-world applicability. Similarly, the Arthur D. Little's IEM (Kirchgeorg et al., 2010) is another paradigm that has been meticulously crafted with a strong emphasis on real-world orientation.

Drawing inspiration from these perspectives, the model proposed in this paper is uniquely shaped by incorporating these valuable contributions. The resulting model represents a synthesis of these insights, underscoring its holistic approach that takes into account both theoretical constructs and pragmatic considerations. Fig. 3 provides an overview of the IEM, showcasing its key components and functionalities. The practical experience distinctly underscores the significance of these specific components as well.

3.1. Harmony of the Critical Components

The IEM perceives "corporate innovation" as a comprehensive approach encompassing management activities dedicated to fostering "innovation projects." Through this perspective, it delineates three fundamental cycles of innovation processes tailored

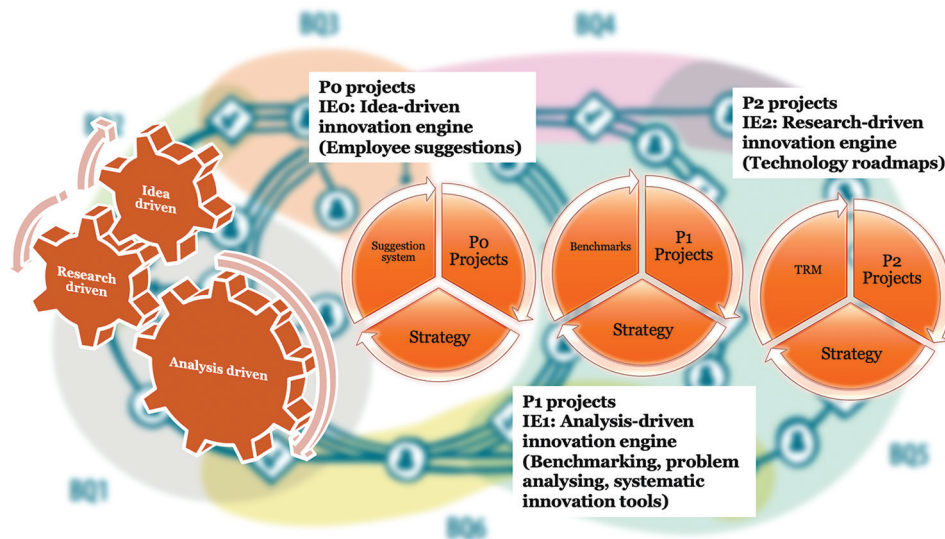


Fig. 3. An overview of the proposed model

for the primary project types which can be found in typical corporate innovation scenarios.

These cycles, namely *P0*, *P1*, and *P2*, serve as essential frameworks for effectively navigating and orchestrating innovation initiatives within organizations. Project cycles are set in motion by three distinct engines: the idea-driven engine, the analyses-driven engine, and the research-driven engine. The prominence of each engine's role can vary based on the competitive environment and strategic priorities of the organization.

The structuring of the innovation management system through project-based cycles draws inspiration from the Big Picture model. This model has been adapted to accommodate different types of projects, resulting in a differentiated approach that offers a fresh perspective.

The core of this approach is the differentiation among project categories, which leads to the emergence of a nuanced model. Furthermore, the innovation engines that propel these project cycles have been formulated by drawing insights from the descriptions of Kirchgeorg et al. (2010).

Table 2 provides a comprehensive comparison of the corresponding project types. This comprehensive comparison highlights the distinct characteristics and dimensions of each project type within the IEM.

Each project type serves specific objectives, faces varying levels of risk, and requires unique management approaches. The comparison table serves as a guide to understand the diversity of projects and their respective contributions to organizational innovation efforts.

3.2. Cycle of the *P0* Projects

"*P0* projects" are dedicated to continuous innovation. These projects are aimed at fostering

ongoing improvement and advancement within an organization. In general, the responsibility for *P0* projects can be distributed among various departments depending on the organization's structure and industry, those are more frequently handled by world-class manufacturing offices or total quality management offices in automotive industry.

Within the context of an automotive manufacturing company, an example could be the implementation of Lean Manufacturing principles on the assembly line to reduce waste, increase efficiency, and improve overall production quality. These projects prioritize incremental enhancements and align with the company's ongoing commitment to operational excellence.

"Corporate suggestion systems" can indeed be one of the main drivers of the *P0* projects. These systems provide a platform for employees to contribute their ideas and suggestions for improving processes, products, or services within the organization. Indeed, *P0* projects are typically grounded in the idea-driven innovation engine. This engine emphasizes the generation and exploration of new ideas as the primary driver for initiating projects. Emphasizing the idea-driven engine for *P0* projects can foster a culture of innovation within the organization. It encourages employees to actively contribute ideas, engage in problem-solving, and collaborate to drive impactful initiatives.

3.3. Cycle of the *P1* Projects

"*P1* projects" are structured around the "analysis-driven engine," characterized by distinct driving forces such as benchmark studies and systematic innovation tools. These projects predominantly revolve around the development of novel products or processes.

Table 2. Comprehensive comparison of the project types

Criteria	<i>P0</i> projects	<i>P1</i> projects	<i>P2</i> projects	<i>P3</i> projects	<i>P(-1)</i> projects
Primary objective	Continuous improvement	Enhancing existing products/processes	Research and development of new technologies/products	Radical innovation and breakthrough ideas	Reverse innovation from emerging markets
Risk level	Low	Moderate	High	Very high	High
Innovation type	Incremental innovation	Product/process innovation	Research-driven innovation	Radical innovation	Adaptation and innovation for new markets
Implementation area	Across departments or functions	Mainly within specific departments	Research and development departments	Cross-functional teams or innovation laboratories	Emerging markets and global adaptation
Key drivers	Employee suggestions, process optimization	Benchmark studies, systematic innovation tools	Strategic technology roadmaps, emerging technologies	Disruptive ideas, breakthrough innovations	Adaptation to local needs and contexts
Data and metrics	Process metrics, employee participation	Performance metrics, market analysis	Technological feasibility, innovation metrics	Market impact, transformation metrics	Market fit, local impact
Resource allocation	Distributed within organization	Dedicated teams and resources	R&D departments, specialized experts	Cross-functional teams, innovation laboratories	Local teams, market-oriented resources
Time horizon	Short-term, ongoing	Medium-term, project-based	Long-term research and development cycle	Long-term, high-risk	Market adaptation over the long term
Management approach	Continuous improvement mindset	Analysis-driven decision-making	Research and development process	Disruptive thinking, risk management	Market adaptation, flexible strategies
Innovation culture emphasis	Encouraging ideas, incremental gains	Analytical decision-making, efficiency	Technical expertise, research excellence	Radical thinking, experimentation	Adaptation, market responsiveness
Global versus local focus	Often internal, some external engagement	Mainly internal, some external engagement	Internal focus, technological exploration	Internal and external, disruptive potential	Local focus, market-specific innovation
Outcome expectation	Continuous enhancement of processes	Improved products or processes	Novel technologies, patentable inventions	Breakthrough products or services	Adapted products for new markets
Performance measurement	Process efficiency, employee involvement	Product performance, market share	Technological feasibility, innovation impact	Market disruption, transformative change	Market penetration, competitive success

Effective management of *P1* projects necessitates the establishment of a dedicated unit within the organization, one equipped with full-time resources and competencies. Given their intensity, *P1* projects require a specialized focus on innovation engineering.

An example in the automotive industry might involve the creation of a more fuel-efficient engine for a specific vehicle model. The analysis-driven engine could be employed to assess market demands, analyze competitive benchmarks, and formulate a precise strategy for integrating new technologies. This project

would emphasize the optimization of current product offerings while staying aligned with the company's overall innovation strategy.

Much like "*P0* projects," the inception of "*P1* projects" is rooted in a purpose-driven approach. While "*P0* projects" are directed toward continuous innovation and sustained enhancement, "*P1* projects" take on a more targeted orientation, with a focus on optimizing existing processes and products. In this sense, the innovation strategy for "*P1* projects" centers on meticulous analysis and strategic alignment. The

engagement of specialized units, equipped with the necessary expertise, underscores the commitment to precision and effectiveness.

The organizational structure for *P1* projects may vary based on the industry and context. However, the presence of a dedicated department specifically focused on managing “*P1* projects” is pivotal. This unit is entrusted with orchestrating the intricate facets of analysis, benchmarking, and innovation strategy. A team well-versed in innovation engineering is crucial for driving “*P1* projects” toward successful outcomes.

By highlighting the analysis-driven engine and strategic alignment, “*P1* projects” emphasize a deliberate and calculated approach to innovation. This methodology facilitates the integration of data-driven insights, benchmarking data, and systematic innovation tools into the project’s fabric. It underscores the importance of informed decision-making and precise execution, ultimately leading to the creation of impactful innovations within the organization.

3.4. Cycle of the *P2* Projects

“*P2* projects,” on the other hand, operate within the framework of the research-driven engine. These projects, exemplified by initiatives like New Technology Exploitation (NTE) projects (Bigwood, 2004), place a heightened emphasis on research. They encompass projects that delve deeply into research-driven exploration and technology development. Typically, these projects are overseen by organizations’ Research and Development (R&D) departments, given their research-intensive nature. An illustrative example could be the development of autonomous driving capabilities for a fleet of vehicles. This project could utilize the research-driven engine to explore emerging technologies, conduct in-depth research on self-driving systems, and develop cutting-edge algorithms. The project’s success hinges on staying at the forefront of technological advancements while aligning with the company’s long-term strategic technology roadmap.

NTE projects, falling under the umbrella of “*P2* projects,” embody a research-centric focus. These projects are marked by their dedication to leveraging available and emerging technologies. Their main driving force lies in strategic technology roadmaps. These roadmaps guide the direction of NTE projects by aligning them with the overarching technological strategy of the organization. This alignment ensures that innovation efforts are purposeful, directed, and in line with the long-term technological vision.

To effectively manage the portfolio of “*P2* projects,” technology roadmaps aligned with this strategic objective can be utilized. These roadmaps take into account the organization’s technological

aspirations and capabilities, offering a roadmap for the successful execution of “*P2* projects.” This approach enhances decision-making, resource allocation, and overall project management, ensuring that each project contributes meaningfully to the organization’s technological advancement.

Within organizations, the R&D departments play a pivotal role in steering the course of “*P2* projects.” These departments are equipped with the expertise needed to oversee the intricate research processes, technology assessments, and innovation strategies that underpin these projects. The collaboration of multidisciplinary teams within R&D departments is a key to driving the successful realization of “*P2* projects.”

3.5. Cycle of the *P3* Projects

P3 projects carry higher risk, are more innovative, and typically target more radical innovations compared to other projects. On the other hand, *P0*, *P1*, and *P2* projects tend to concentrate on improving existing processes, products, and available technologies, with lower risk associated.

P3 projects are all about innovation and creating something entirely new. For instance, consider a project where the goal is to develop a flying car. This type of endeavor goes beyond refining existing concepts; it is about embracing radical ideas and pushing boundaries. In this case, engineers and designers are inventing a completely novel mode of transportation, exploring uncharted territories of technology. *P3* projects are where ground-breaking ideas take shape and bring transformative change.

3.6. Cycle of the *P(-1)* Projects

For multinational corporations, a distinct cycle, denoted as “*P(-1)* projects,” might be required. These projects fall within the scope of “reverse” innovation, where solutions are developed in emerging markets and later adapted globally (von Zedtwitz et al., 2015), and they may necessitate the application of both analysis-driven and research-driven engines based on project specifics.

For an automotive company, this might involve creating an affordable and durable vehicle tailored to the needs of developing countries. The analysis-driven engine could be used to identify market gaps, while the research-driven engine could explore innovative manufacturing processes that suit the local context. This project highlights the unique challenges and opportunities of reverse innovation, aiming to address specific market demands.

The model proposed in this study does not incorporate this project type. Global innovation projects encompass various unique circumstances that

multinational companies need to consider. Factors such as communication and information flow (von Zedtwitz and Joachim, 2020), as well as organizational structures (von Zedtwitz et al., 2004), require specific configurations for these projects. Therefore, attempting to overly simplify them to fit within a generic innovation management system could be misleading for corporations.

4. Measuring the Performance

After the innovation system is planned, careful attention is given to evaluating its performance. This assessment is made using “CIP - Corporate Innovation Performance,” a comprehensive measure that shows how effective the system is. The CIP is calculated using Equation 1, which highlights a quantitative method for gauging the success and influence of the innovation structure.

$$CIP = \sum_{i=0}^2 MP_i \times IE_i \times ((EnP_i) \times (ExP_i) \times (SP_i)) \quad (1)$$

CIP: Corporate innovation performance

MP_i: Maturity degree

EnP_i: Engine performance

ExP_i: Project execution performance

SP_i: Strategy performance

IE_i: Importance degree of the innovation engine

In this equation, *CIP* stands for Corporate Innovation Performance, which is the ultimate result of this calculation. It serves as a quantified measure of how well the innovation system is performing within the organization.

MP_i refers to the Maturity Degree, which gauges how developed or advanced a particular aspect of the organization’s innovation framework is. This value captures the level of sophistication in terms of innovation practices.

IE_i signifies the importance degree of the innovation engine. This factor quantifies the significance of each innovation engine (idea-driven, analysis-driven, and research-driven) within the overall innovation process.

EnP_i represents Engine Performance, which evaluates how well each innovation engine operates in practice. It measures the efficiency and effectiveness of the engines’ functionalities.

ExP_i relates to Project Execution Performance, assessing how efficiently and successfully projects (*P0*, *P1*, *P2*) are executed within the organization. This component encapsulates the project management capabilities of the innovation system.

SP_i stands for Strategy Performance, which evaluates how well the organization’s innovation strategy aligns with its overall corporate objectives. It measures the strategic coherence between innovation endeavors and business goals.

The calculation involves multiplying these various factors together and then aggregating the results for each “*i*” value from 0 to 2, representing the three different project cycles. This holistic approach provides a holistic view of the organization’s innovation performance, considering multiple dimensions that contribute to the overall effectiveness of the innovation management system.

Consider the following hypothetical example to demonstrate the calculation of the *CIP* using the provided formula. Assume we have an automotive manufacturing company that is evaluating its CIP. The company employs three different project cycles: *P0*, *P1*, and *P2*. Each of these cycles operates at different levels of maturity, has varying importance degrees, and demonstrates diverse performance levels.

Let’s assign some arbitrary values to these parameters for the purpose of illustration:

For the P0 projects cycle (i = 0):

Maturity degree (*MP₀*) = 0.7

Importance degree (*IE₀*) = 0.4

Engine performance (*EnP₀*) = 0.8

Project execution performance (*ExP₀*) = 0.6

Strategy performance (*SP₀*) = 0.9

For the P1 projects cycle (i = 1):

Maturity degree (*MP₁*) = 0.6

Importance degree (*IE₁*) = 0.3

Engine performance (*EnP₁*) = 0.7

Project execution performance (*ExP₁*) = 0.5

Strategy performance (*SP₁*) = 0.8

For the P2 projects cycle (i = 2):

Maturity degree (*MP₂*) = 0.8

Importance degree (*IE₂*) = 0.3

Engine performance (*EnP₂*) = 0.9

Project execution performance (*ExP₂*) = 0.7

Strategy performance (*SP₂*) = 0.85

Plugging these values into the formula:

$$CIP = 0.12096 + 0.0504 + 0.1638 = 0.33516$$

The derived CIP value serves as a pivotal metric indicative of the organization’s ongoing need for continuous improvement. It stands as a dynamic gauge of the innovation system’s effectiveness, highlighting the extent to which the various dimensions of the innovation ecosystem are aligned and contributing to the company’s innovative prowess.

CIP is a measurement framework of the IEM. Its sub-level performance factors and indicators need consideration of detailed company-specific dimensions. Dziallas and Blind (2019) provide an extensive literature analysis on innovation indicators throughout the innovation process. A comprehensive

list of the corresponding factors and indicators can be found in Dziallas and Blind (2019).

In its essence, CIP reflects not only the present state of the innovation management system but also serve as a harbinger of future endeavors. Much like the Overall Equipment Effectiveness (OEE) metric that underscores manufacturing performance, CIP serves as a tailored, innovation-specific integrated KPI for the organization. Just as OEE provides insights into the efficiency of manufacturing processes, CIP offers a comprehensive view of the innovation landscape, encompassing engine maturity, project execution, strategy alignment, and more.

4.1. Determining the Maturity Degree

The IEM defines three distinctive maturity levels within the realm of corporate innovation. Each level is characterized by unique attributes, corresponding to specific coefficients that contribute to the overall CIP calculation.

1. *Level-1: Transparent Management (MP_i : 0.33)*

At this level, the innovation execution processes are clearly defined, fostering transparency in management practices. KPIs are established, providing a transparent framework for evaluating innovation endeavors.

2. *Level-2: Systematic Management (MP_i : 0.66)*

As innovation progresses to this level, a systematic approach prevails. System behavior becomes predictable, and data-driven management practices come to the fore. Decisions are guided by empirical insights, enhancing the efficiency and effectiveness of the innovation ecosystem.

3. *Level-3: Intelligent Management (MP_i : 1)*

At the pinnacle of the innovation hierarchy, intelligent management takes center stage. Real-time data become the cornerstone of decision-making, enabling the system to function optimally. Continuous adaptation and refinement are inherent, ensuring that the organization operates under optimal conditions.

These maturity levels underscore the evolution of the innovation framework, each contributing to the holistic calculation of CIP. The model not only provides a quantified measure of innovation performance but also delineates the path to advancing corporate innovation, from transparent management to intelligent, data-driven optimization.

4.2. Determining the Importance Degree

Determining the importance degree is a context-specific task, influenced by a variety of factors. These include the company's goals, the

market landscape, available resources, and industry alignment. In essence, it is a tailored evaluation, reflecting the strategic choices that align with the company's vision.

A prominent consultancy company contributes to this discussion through insightful reports (Kirchgeorg et al., 2010). These reports offer guidance to companies seeking direction in setting their importance degree levels. This external perspective acts as a compass, providing a wider view and potential benchmarks for organizations navigating this decision-making process. For instance, in the case of an automotive manufacturer, based on the report, importance degree levels might be aligned as follows, as depicted in Fig. 4.

It is important to note that while external references provide insight, the final determination of importance degree remains an internal endeavor. By blending internal assessment with external insights, companies create an importance degree framework that resonates with their unique goals.

4.3. Performance Metrics and Assessment

In this section, we discuss the performance metrics; EnP_i – Engine Performance, ExP_i – Project Execution Performance, and SP_i – Strategy Performance.

4.3.1. EnP_i – engine performance

EnP_i measures how efficiently and effectively each innovation engine operates. In the IEM, innovation engines include the idea-driven, analysis-driven, and research-driven engines, each supporting different types of innovation projects.

Key elements to assess:

- Efficiency rate: How quickly does each engine process ideas, analyses, or research? For example, the idea-driven engine should move ideas from conception to action quickly, showing high efficiency.
 - Success rate of outputs: What percentage of ideas or research findings lead to meaningful projects? A high success rate indicates that the engine consistently produces valuable outputs.
 - Adaptability to market changes: How well does the engine respond to changes in technology and market demand? For instance, an analysis-driven engine should quickly integrate market shifts and competitor insights.
- Assessment scale (0 to 1):
- Low: The engine operates slowly, rarely produces successful outputs, and is not responsive to market changes.

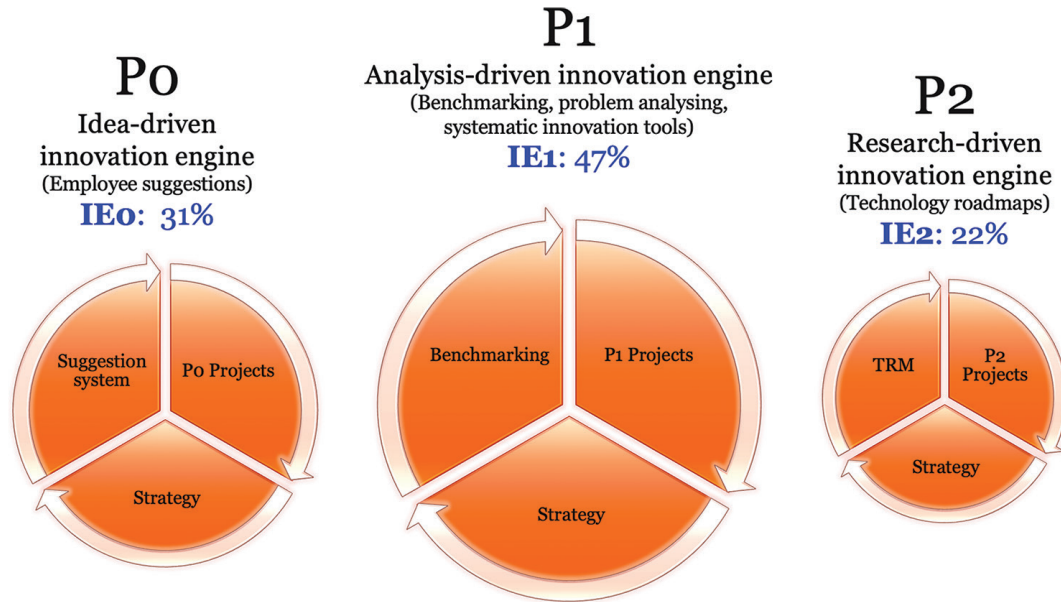


Fig. 4. Importance degree of innovation engines: Automotive industry rates

- Moderate: The engine functions with average speed and output success, with basic responsiveness to change.
- High: The engine is highly efficient, with a high output success rate and excellent adaptability.

4.3.2. ExP_i – project execution performance

ExP_i evaluates how well innovation projects ($P0$, $P1$, $P2$) are executed. This metric looks at timelines, resource usage, and achievement of project goals.

Key elements to assess:

- Timeliness: Are projects completed on schedule? This factor measures whether projects meet their deadlines, essential for maintaining momentum in innovation.
 - Resource utilization: Is the organization using resources efficiently? This includes staying within budget and maximizing manpower without waste.
 - Goal achievement rate: Are the project objectives being met? For instance, if a project aims to improve product efficiency by 10%, this metric evaluates whether that goal is achieved.
- Assessment scale (0 to 1):
- Low: Projects are often delayed, over budget, and few goals are met.
 - Moderate: Projects usually meet deadlines and budgets but may achieve only some of their goals.
 - High: Projects consistently stay on time, within budget, and meet or exceed their goals.

4.3.3. SP_i – strategy performance

SP_i assesses how well the innovation strategy aligns with broader organizational goals, such as market expansion or sustainability. It measures strategic alignment, market impact, and risk management.

Key elements to assess:

- Strategic alignment: Are innovation projects aligned with the company's strategic goals? This includes ensuring projects contribute to long-term objectives.
 - Market impact: Do the innovations positively impact the market, improve customer satisfaction, or provide a competitive edge? High market impact shows that the organization's innovations are valued externally.
 - Risk management: How effectively does the organization manage risks in its innovation activities? Effective risk management reduces the chance of project failures due to unforeseen challenges.
- Assessment scale (0 to 1):
- Low: Projects rarely align with strategic goals, have minimal market impact, and risk management is poor.
 - Moderate: Projects sometimes align with strategy, make a moderate market impact, and have basic risk management.
 - High: Projects align closely with goals, have a significant market impact, and provide risk management in place.

These metrics, assessed on a 0 to 1 scale, provide a structured approach to evaluating innovation performance. By regularly assessing EnP_p , ExP_p , and SP_p , organizations gain insights into their innovation system's strengths and areas for improvement, supporting a continuous journey toward excellence in innovation management.

5. Keeping the Innovation Engines Synchronized

The IEM takes into account corporate innovation as management actions for innovation projects. Following this perspective, it outlines three fundamental cycles of innovation processes that correspond to the primary project types found in typical corporate innovation scenarios. These cycles are known as $P0$, $P1$, and $P2$ project cycles. Each of these project cycles is set into motion by one of three distinct engines: the idea-driven engine, analysis-driven engine, and research-driven engine, respectively.

The choice of which engine to prioritize can vary based on the corporate's competitive landscape and strategic direction. This determination of engine balance constitutes a strategic choice. Deciding on these equilibrium rates is a pivotal step in the strategic journey. Equally crucial is the question of how to ensure the entire system operates in accordance with these designated rates.

Determining the right balance and maintaining the synchronization of these engines within the innovation management system is a strategic imperative. In this context, it becomes a pivotal issue to ensure that the selected engine's prominence aligns with the corporate strategy. This strategic alignment forms the bedrock for achieving innovation success. The selected engine

dictates the rhythm and emphasis of the innovation process, steering it toward optimal results in line with the overarching strategic goals.

This scenario highlights an environment with multiple projects in play. Successfully managing such an array of endeavors requires considering critical elements simultaneously. These encompass portfolio management, project cycle planning, and the equitable allocation of shared resources.

What amplifies the complexity of this landscape is the intrinsic nature of these projects, which revolve around innovation. This, in turn, introduces an element of uncertainty, underscoring the imperative of adept real-time decision-making as a pivotal driver of success.

5.1. Card-based Navigation

Due to its computational advantages and decreased vulnerability to uncertainty, a preference often emerges for card-based control approaches in real-world contexts. Card-based systems rely on the inherent signals of the existing system to authorize releases (Riezebos, 2006).

Among the card-based control systems documented in existing literature, COBACABANA – Control of Balance by Card-Based Navigation, as elaborated by Land (2009), shines as one of the most intricate and sophisticated.

Its operational mechanism seamlessly aligns with the intricate task of managing multiple projects within the IEM. This mechanism facilitates the execution of project releases from the portfolio, with the optimized allocation of cards ensuring the model's effective implementation.

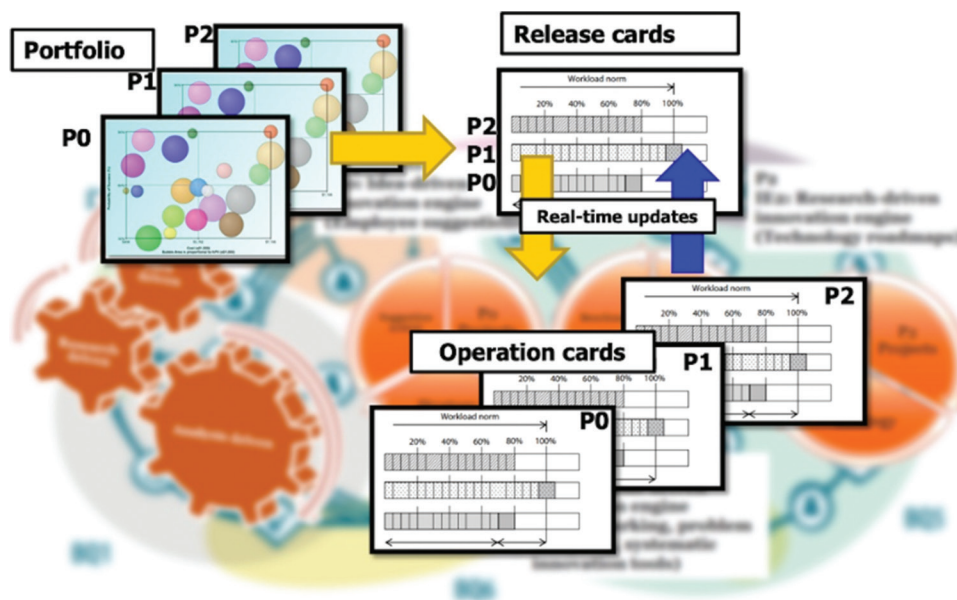


Fig. 5. Keeping innovation engines synchronized using a card-based control approach

As depicted in Fig. 5, the IEM takes tangible form. The determination of release card quantities is executed with careful consideration for maintaining a harmonious equilibrium among the project engines.

At this pivotal stage, the adoption of discrete event simulation is recommended for conducting “what-if” analyses and simulation optimization.

Following the quantification of release card quantities, project releases receive authorization if an adequate number of release cards is available on the panel. These release cards become integral to the authorized projects, and upon the completion of their respective stage gates, they revert to the panel. Within practical contexts, the management of these processes in real time finds a suitable ally in a BPM-Business Process Management system.

5.2. Simulation Modeling

In this section, we extend our discussion with a simulation study conducted using the Simio environment. The choice of Simio was driven by its ability to easily model working scenarios, especially

those involving limited Work-in-Progress (WIP) systems. Simio’s buffer logic concept enables these models to be implemented without extra complexity. Our simulation model employs state variables to model release cards in a constrained manner, tailored to specific project types. For instance, for $P0$ projects, the card limit was set at 31; for $P1$ projects, it was 47; and for $P2$ projects, it was 22. Fig. 6 displays the initial screen when running the simulation model.

Each new project entry is constrained by the availability of these cards. Depending on the project workload, card assignments are made during project entries. As projects complete their stage-gate phases, these cards are released, creating new capacity. The stage-gate sections can be customized according to each project’s workflow. In this example model, we assume that each project type follows a 4-phase process.

In Fig. 7, we illustrate a scenario where the absence of available cards blocks the entry of projects into the system. After the warm-up period, the simulation model continues to operate with the constrained capacities as reflected in the distribution of

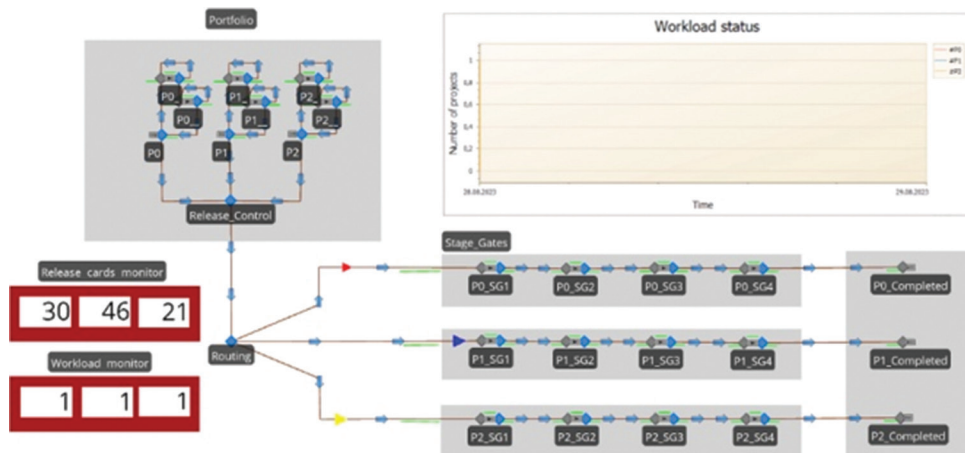


Fig. 6. The initial screen when running the simulation model

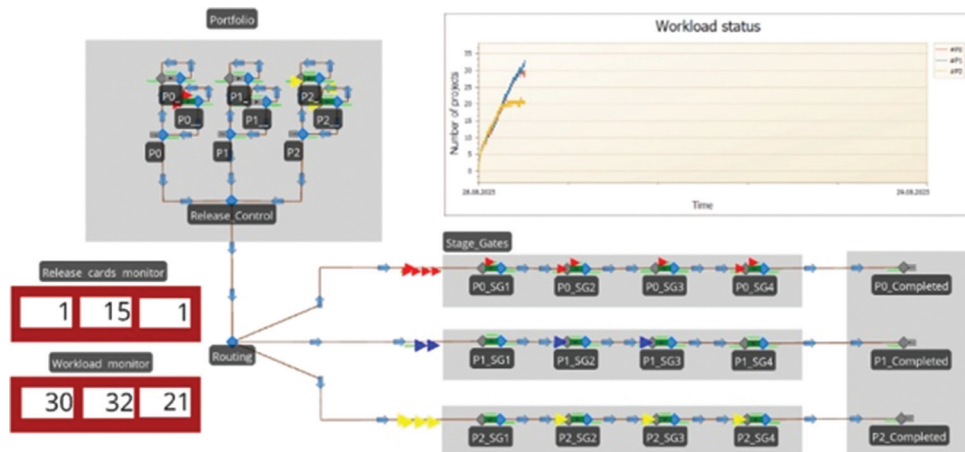


Fig. 7. Blocking the entry of projects into the system

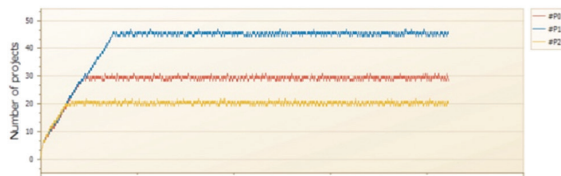


Fig. 8. The workload status

workload. The long-term workload status is depicted in Fig. 8, affirming the ability to control project workload balance through card-based control.

It is important to note that while this study establishes the feasibility of managing project workload using card-based control, a real-world implementation would involve using detailed operational data to fine-tune the system's parameters for optimal performance. In light of this, the model's adaptations can guide the design efforts toward an efficient real-world setup.

6. Concluding Remarks

Harmonizing IEM with theoretical foundations, the CIP metric, and practical strategies offers organizations a comprehensive toolkit for sustainable success. By fostering a holistic understanding of innovation engines, project types, resource allocation, and performance measurement, the model equips organizations to navigate the complexities of innovation management, propelling them toward excellence and growth.

Future research can contribute to the refinement and enrichment of the IEM, making it an even more potent tool for organizations seeking to excel in the dynamic landscape of innovation management. Several areas offer promising avenues for future research and exploration: Conducting in-depth case studies of organizations that have adopted the IEM can provide practical insights into its implementation, challenges faced, and lessons learned.

References

- Bigwood, M.P. (2004). Managing the new technology exploitation process. *Research-Technology Management*, 47, 38–42.
<https://doi.org/10.1080/08956308.2004.11671661>
- Dahlgaard-Park, S.M., & Dahlgaard, J.J. (2008). A strategy for building sustainable innovation excellence - A Danish study. In: K.J. Zink (Ed.), *Corporate Sustainability as a Challenge for Comprehensive Management*. Springer, New York, p77–94.
- Dahlgaard, J.J., Chen, C.K., Yang, J.Y., Banegas, L.A., & Dahlgaard-Park, S.M. (2013). Business

excellence models: Limitations, reflections and further development. *Total Quality Management & Business Excellence*, 24, 519–538.

<https://doi.org/10.1080/14783363.2012.756745>

- Dervitsiotis, K.N. (2010). A framework for the assessment of an organization's innovation excellence. *Total Quality Management & Business Excellence*, 21, 903–918.

<https://doi.org/10.1080/14783363.2010.487702>

- Diedrichs, E., Engel, K., & Wagner, K. (2006). *IMP3rove: Assessment of Current Practices in Innovation Management Consulting Approaches and Self-assessment Tools in Europe to Define the Requirements for Future Best Practices. European Innovation Management Landscape. Europe INNOVA Paper*, 2. Augsburg, Germany.

- Dzialis, M., & Blind, K. (2019). Innovation indicators throughout the innovation process: An extensive literature analysis. *Technovation*, 80–81, 3–29.

<https://doi.org/10.1016/j.technovation.2018.05.005>

- Hyland, J., & Karlsson, M. (2021). Towards a management system standard for innovation. *Journal of Innovation Management*, 9, 11–19.

https://doi.org/10.24840/2183-0606_009.001_0002

- Kimiloglu, H., & Zarali, H. (2009). What signifies success in e-CRM. *Marketing Intelligence & Planning*, 27(2), 246–267.

<https://doi.org/10.1108/02634500910945011>

- Kirchgeorg, V., Achtert, M., & Großschmidt, H. (2010). *Pathways to Innovation Excellence: Results of a Global Study by Arthur D. Little*. Available from: <https://www.adlittle.com/en/insights/viewpoints/pathways-innovation-excellence> [Last accessed on 2023 Aug 27].

- Land, M.J. (2009). COBACABANA (Control of balance by card-based navigation): A card-based system for job shop control. *International Journal of Production Economics*, 117, 97–103.
<https://doi.org/10.1016/j.ijpe.2008.08.057>

- Lee, Y., & Rim, S.C. (2016). Quantitative model for supply chain visibility: Process capability perspective. *Mathematical Problems in Engineering*, 2016, 4049174.

<https://doi.org/10.1155/2016/4049174>

- Lercher, H. (2020). *Big Picture - The Graz Innovation Model*. Available from: <https://ssrn.com/abstract=2965373>

- Mann, R., & Grigg, N. (2004). Helping the kiwi to fly: Creating world-class organizations in New Zealand through a benchmarking initiative. *Total Quality Management & Business Excellence*, 15(5–6), 707–718.

<https://doi.org/10.1080/14783360410001680198>

- Martensen, A., Dahlgaard, J., Dahlgaard, S.M., & Gronholdt, L. (2007). Measuring and diagnosing innovation excellence - simple contra advanced

- approaches: A Danish study. *Measuring Business Excellence*, 11, 51–65.
<https://doi.org/10.1108/13683040710837928>
- Mele, C., & Colurcio, M. (2006). The evolving path of TQM: Towards business excellence and stakeholder value. *International Journal of Quality & Reliability Management*, 23, 464–474.
<https://doi.org/10.1108/02656710610664569>
- Mohammad, M., Mann, R., Grigg, N., & Wagner, J.P. (2011). Business excellence model: An overarching framework for managing and aligning multiple organizational improvement initiatives. *Total Quality Management & Business Excellence*, 22, 1213–1236.
<https://doi.org/10.1080/14783363.2011.624774>
- Riezebos, J. (2006). POLCA Simulation of a Unidirectional Flow System. In: *Proceedings of the third International Conference on Group Technology/Cellular Manufacturing*. University of Groningen, Groningen, The Netherlands, p332–338.
- Von Zedtwitz, M., Corsi, S., Søberg, P.V., & Frega, R. (2015). A typology of reverse innovation. *Journal of Product Innovation Management*, 32, 12–28.
<https://doi.org/10.1111/jpim.12181>
- Von Zedtwitz, M., Gassmann, O., & Boutellier, R. (2004). Organizing global R&D: Challenges and dilemmas. *Journal of International Management*, 10, 21–49.
<https://doi.org/10.1016/j.intman.2003.12.003>
- Von Zedtwitz, M., & Joachim, M. (2020). Communication and Knowledge Flows in Transnational R&D Projects. In: *Managing Innovation in a Global and Digital World*. Springer Gabler, Wiesbaden, p227–251.

AUTHOR BIOGRAPHIES

Dr. Koray ALTUN is an Assistant Professor of Industrial Engineering at TU Bursa and a Research Collaborator at GLORAD. He is also the founder and manager of a startup focused on digital innovation and software consultancy. He holds a Ph.D. and a BSc degree in Industrial Engineering from Gaziantep University and Erciyes University, respectively. He has published papers in well-respected journals and has led consulting engagements in reputable companies with a focus on R&D, technology, and innovation management. His recent research interests include the Innovation Excellence Model, Systematic Innovation, Technology and Innovation Management, and R&D Management.

Structure learning of Bayesian networks using sparrow optimization algorithm

Shahab Wahhab Kareem^{1,2*}, Hoshang Qasim Awla³, Amin Salih Mohammed⁴

¹Department of Technical Information Systems Engineering, Erbil Technical Engineering College, Erbil Polytechnic University, Erbil, Iraq

²Department of Computer Technical Engineering, Al-Qalam University College, Kirkuk, Iraq

³Department of Computer Science, Faculty of Science, Soran University, Soran, Iraq

⁴Department of Software and Informatics, Salahaddin University, Erbil, Iraq

*Corresponding author E-mail: Shahab.kareem@epu.edu.iq

(Received 12 June 2024; Final version received 10 December 2024; Accepted 19 December 2024)

Abstract

Bayesian networks are powerful analytical models in machine learning, used to represent probabilistic relationships among variables and create learning structures. These networks are made up of parameters that show conditional probabilities and a structure that shows how random variables interact with each other. The structure is shown by a directed acyclic graph. Despite the NP-hard nature of learning Bayesian network structures, there has been significant progress in improving the accuracy of approximation solutions. The main focus is on score-based search strategies, which make use of functions to evaluate network models and identify structures with high scores. This study is significantly focused on structure learning Bayesian networks using the Bayesian Dirichlet equivalent uniform scoring function and metaheuristic search strategies. To this end, this paper presents the sparrow optimization algorithm (SOA), a new metaheuristic algorithm derived from the foraging behavior of sparrows. SOA performs a concurrent optimization in the solution space by simultaneously performing a local and global search that leads to the discovery of near-optimal structures. The results from our experiments on several benchmark datasets show that SOA yields overall better performance than SA and greedy search algorithms. In particular, it is claimed that by applying the proposed approach of SOA, the convergence speed is significantly higher compared with the existing ones; F1 score is 0.35 and 0.05 for the Hamming distance with better results. Given these results, signed operators prove to be very efficient in SOA's Bayesian network structure learning as a concept, especially for real-world use.

Keywords: Search and Score, Global and Local Search, Bayesian Network, Sparrow Search Optimization Algorithm, Structure Learning

1. Introduction

The Bayesian network is widely regarded as a widely used analytical model in machine learning for constructing the probabilistic framework of knowledge (Ji et al., 2012). It is feasible to employ knowledge design, reasoning, and inference systematically (Fortier et al., 2013). A directed acyclic graph (DAG) is utilized to illustrate the structure of a Bayesian network, consisting of two fundamental elements: the network's parameters and its structure. The structure is used to express dependencies on variability, whereas

the parameters are employed to describe conditional probabilities. The task of addressing the learning structure of a Bayesian network might be challenging in the absence of a well-defined search plan. However, significant efforts have been made to develop approximation algorithms for acquiring knowledge about the network structure. Achieving the appropriate NP-hard class is necessary to overcome the challenges associated with learning the structure of a Bayesian network from a dataset (Li & Chen, 2014). The main components of structural learning in Bayesian networks

are two distinct processes. While the second strategy employs a combination of score and search tactics, the first approach is focused on constraints (Margaritis, 2003). Until the desired metric value is reached, the scoring and search approach is used to methodically evaluate each potential network structure and explore the range of Bayesian network configurations. Using a function to evaluate the network and the provided data to maximize the score – the intended result – is the basis of score-based approaches (Fast, 2010). The Bayesian score and the information-theoretic score are the two primary criteria used to generate the score function approach. Bayesian networks are valuable tools in decision-making processes because they can discover connections between variables and make predictions utilizing uncertain data. The search and scoring component performs an essential part of the Bayesian network structure learning (BNSL) process. The process involves examining several potential network structures and assessing their suitability by utilizing scoring criteria that determine their level of compatibility. The complexity increases with the rise in the number of variables, leading to a significant expansion in the number of possible DAGs that can represent the relationships between variables. As the number of variables grows, a typical issue that arises is the NP-hard problem, which occurs in the majority of search spaces. The NP-hard problem of BNSL relates to the difficulty of determining the optimal network structure for a given dataset. This problem becomes more difficult when using search and scoring methods, which require analyzing a wide range of possible structures of networks and evaluating their sufficiency. The NP-hardness arises from the exponential growth of the search space with an increasing number of variables. This makes it practically impossible to thoroughly search over all possible options, particularly for huge datasets.

The application of the information-theoretic score involves the utilization of several techniques, such as mutual information tests, minimum description length, normalized minimum likelihood, log-likelihood, Akaike information criterion, and Bayesian information criterion. The Bayesian score is utilized in several approaches, such as BDe (Bayesian Dirichlet, where “e” represents likelihood-equivalency), BDeu (Bayesian Dirichlet equivalent uniform, where “u” denotes uniform joint distribution), and K2 (Cooper & Herskovits, 1992).

The complexity of structure learning can be enhanced through the utilization of diverse search strategy approaches. The literature includes references to several algorithms, including Bee Colony (Li & Chen, 2014), Swarm Intelligence (Cowie et al., 2007), Ant Colony (Salama & Freitas, 2012), Hybrid Algorithm (He & Gao, 2018; Kareem & Okur, 2018; Li & Wang,

2017), Simulated Annealing Algorithm (Hesar, 2013), Bacterial Foraging Optimization (Yang et al., 2016), and Genetic Algorithms (Larrañaga & Poza, 1996). Numerous algorithms have been examined in the existing body of literature (Djan-Sampson & Sahin, 2004; Fan et al., 2014; Orphanou et al., 2018; Yuan et al., 2011; Rahier et al., 2019). Several algorithms have been proposed in the literature, such as the Breeding Swarm Algorithm (Khanateymoori et al., 2018), Binary Encoding Water Cycle (Wang & Liu, 2018), Pigeon Inspired Optimization (Kareem & Okur, 2019), Cuckoo Optimization Algorithm (Askari & Ahsae, 2018), and Minimum Spanning Tree Algorithm (Sencer et al., 2013). The utilization of swallow optimization, a cutting-edge metaheuristic approach, is prevalent in the field of structure learning within Bayesian networks. This study presents a novel approach to address the difficulty of acquiring knowledge about the architecture of Bayesian networks. This study provides a comparative assessment of the technique mentioned earlier. This paper presents the sparrow optimization algorithm (SOA), a new metaheuristic strategy based on sparrows’ foraging habits in enhancing the structure learning of Bayesian networks. SOA does the concurrent running of both local and global searches, thereby improving the discovery of almost optimal structures. In this paper, benchmark datasets are used to establish the superiority of SOA over conventional algorithms such as SA and greedy search, especially in terms of convergence rate and accuracy. The proposed algorithm can be considered very efficient – it does not take a long time to produce results and seems perfectly capable of dealing with big data. This work emphasizes that SOA can greatly enhance the speed of the BNSL while yielding much better results than other similar approaches.

The subsequent sections of this study are organized in the following manner: Section 2 provides an explanation of the principles underlying structure learning in Bayesian networks. Section 3 provides a concise summary of the SOA. Section 4 delves into the approach extensively and presents the experimental findings. The conclusions are presented in Section 5.

2. Structure Learning of Bayesian Networks

The composition of a Bayesian network has two distinct components, namely G and P. A DAG is the main category, consisting of a finite set of vertices (or nodes), V, that are connected by specified edges (or links), E. The representation of it is denoted by the symbol $G(V; E)$. The equation $P = P(X_i | Pa(X_i))$ describes the collection of conditional probabilistic distributions that are unique for each variable X_i , which corresponds to the vertices in a graph. In addition, it should be noted that the function $Pa(X_i)$ represents the

collection of parents of node X_i inside graph G (Cowie et al., 2007). This model enables the depiction of a basic probabilistic combination for a $(G; P)$ network in the following way:

$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | Pa(X_i)) \quad (1)$$

On the other side, the scoring system is dependent on several criteria, such as the minimal length of description, information and entropy, and Bayesian approaches (Campos, 2006). The posterior likelihood of the Bayesian network can be represented according to the principles of Bayesian inference as follows:

$$P(G/D) = P(D/G) \cdot P(G) / \sum_{G'} P(D/G') \quad (2)$$

The marginal likelihood $P(D|G)$ in Eq. (2) is defined as follows using the normality constant $P(D)$:

$$P(G/D) = \int P(D/G, \theta) P(\theta/G) d\theta \quad (3)$$

The prevailing belief is that $P(D)$ does not constitute a component of the structure of G 's Bayesian network. The variable represents the parameter of the model, whereas the prior probability is given as $P(G')$. Hence, it is possible to calculate the next distribution of the network structure, provided that the marginal probability of all potential topologies has been determined (Zhang & Liu, 2008). Structure learning approaches utilize score-based tactics by integrating the current and historical scores of the structure. The eventual representation of the score is (Heckerman et al., 1995):

$$Score(G, D) = \sum Score(X_i, Pa(X_i), D(X_i, pa(X_i))) \quad (4)$$

3. SOA

Metaheuristics refer to methods that draw inspiration from nature and are used to find possible solutions to complex computational optimization problems. Animals such as fireflies-BAT (Reddy & Khare, 2016), cuckoos (Gadekallu & Khare, 2017), ants, pigeons, fish, bees, and others have utilized their swarming behaviors in metaheuristics (Gandomi et al., 2013). The metaheuristics possess several fundamental qualities, including uniformity, flexibility, illation-free instruments, and the capability to ignore local optima (Mirjalili et al., 2014). The metaheuristic algorithm proposed by Segundo et al. (2019) is derived from the sparrow routine, which is used for hunting food. The SOA method is a reliable and resilient method designed for addressing stochastic population-based problems that require complex configurations involving several parameters and operating in three stages.

The recommended process was impacted by the way sparrows forage, that is, how they look

for food when they are in the air. The sparrow is a solitary creature that adapts its hunting strategy based on its needs. Nevertheless, distinct strategies emerge, and impressive models adhere to the essential principles of flight and navigation in a secure area, as supported by many research findings (Tucker, 1998; 2000). The objectives are assessed for the maximum level of flying accomplishment during different stages of delicate searching or hunting (Hedenström et al., 1999). The implementation methodology of flying in the framework involves the computation of the mechanical power required for navigation, determining the average speeds throughout the flight, and adapting to wind conditions (Hedenström et al., 1999). The sparrow is one of the fastest creatures in the world. The main hunting or searching activity takes place throughout the day, including in the morning. The prime source of nutrition is derived from minor to medium-sized prey and occasionally includes insects (Dekker, 2009).

Based on the above description of sparrows, the authors are able to formulate a mathematical model for developing the sparrow search algorithm. To simplify matters, they conceptualized the subsequent actions of the sparrows and devised related principles.

- (i) Producers often possess abundant energy reserves and offer browsing locations or instructions for all scavengers. Its primary function is to detect and locate regions abundant in healthy food resources. The energy reserves are contingent upon the individuals' fitness values being evaluated.
- (ii) When the sparrow recognizes its attacker, it starts chirping as an alerting signal. If the signal value outstrips the safety threshold, the producers must direct all those searching for resources to the designated safe region.
- (iii) Every sparrow has the probability to suit a producer by seeking out improved food sources, but the ratio of production to scroungers remains constant among the entire population.
- (iv) Sparrows with greater energy levels would function as producers. A group of famished scavengers will be more inclined to migrate to diverse positions in search of meals to acquire additional energy.
- (v) The scavengers trail behind the producer that can offer the highest quality nourishment to forage for food. Meanwhile, certain opportunistic individuals may continuously surveil the producers and engage in food competition to enhance their predation ratio.
- (vi) When sparrows on the outside of the group become aware of danger, they promptly move to a secure region to achieve a more advantageous situation. Conversely, sparrows situated in the center of the group exhibit unpredictable

movement patterns to maintain proximity to their peers. During the simulation experiment, the authors must use virtual sparrows to find sustenance. A specific matrix depicts the spatial distribution of sparrows:

$$X = \begin{bmatrix} X_{1,1} & \cdots & X_{1,d} \\ \vdots & \ddots & \vdots \\ X_{1,n} & \cdots & X_{n,d} \end{bmatrix} \quad (5)$$

$$F_X = \begin{bmatrix} f(X_{1,1}, X_{1,2}, \dots, X_{1,d}) \\ f(X_{12,1}, X_{2,2}, \dots, X_{2,d}) \\ \vdots \\ f(X_{n,1}, X_{n,2}, \dots, X_{n,d}) \end{bmatrix} \quad (6)$$

Each item in the “FX” array represents the fitness value of a single sparrow, whereas the variable “n” shows the total number of sparrows. During the search phase in the SOA, food is prioritized for those with higher fitness values. Furthermore, producers also take on the responsibility of acquiring food supplies and directing population movement as a whole. Because of this, the producers are able to look at a wider variety of resources for food than the scavengers. The producer’s location is changed at each iteration in the following ways, per rules (i) and (ii):

$$V_{i,j}^{t+1} = \begin{cases} V_{i,j}^t \cdot \exp\left(\frac{-i}{\alpha \cdot iter_{max}}\right) & \text{if } R_2 < ST \\ V_{i,j}^t + Q \cdot L & \text{if } R_2 > ST \end{cases} \quad (7)$$

In this context, the variable t shows the current iteration, while j ranges from 1 to d. The notation $V_{(i,j)}$ represents the value of the jth dimension of the ith sparrow at iteration t. The term “itermax” is a constant that denotes the upper limit of iterations. Let α denote a random number that falls within the interval (0, 1). The variable R_2 , which ranges from 0 to 1, denotes the alert value. ST, where ST ranges from 0.5 to 1.0, denotes the safety level. The variable Q exhibits stochasticity and adheres to a normal distribution. The matrix L is a vector space with dimensions $1 \times d$, where each element is equal to 1. In instances where the resource-to-search ratio (R_2) falls below the search threshold (ST), signifying the lack of predators, the producer commences the wide search mode.

If the value of R_2 is greater than or equal to ST, it indicates that certain sparrows have become aware of the predator’s presence, and each sparrow needs to rapidly reposition to alternative secure locations. Regarding the individuals who scrounge, it is necessary to implement and uphold regulations (iv) and (v). As previously said, certain scavengers monitor the producers with greater frequency. Upon discovering

that the producer has located high-quality sustenance, they promptly abandon their present location to vie for nourishment. If they emerge victorious, they will promptly obtain the producer’s food. Otherwise, they will persist according to the guidelines (v). The formula for updating the position of the scrounger is described as follows:

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{X_{worst}^t - X_{ij}^t}{j^2}\right) & \text{if } i > n/2 \\ X_p^{t+1} + |X_{ij}^t - X_p^{t+1}| \cdot A^+ \cdot L & \text{otherwise} \end{cases} \quad (8)$$

The parameter “XP” shows the optimal position set by the manufacturer. The abbreviation Xworst describes the current world position that is usually viewed as the most unfavorable. The matrix A is a $1 \times d$ matrix in which each element is assigned a random value of either 1 or $\Omega 1$. A^+ is the result of multiplying the transposition of A by the inverse of the product of A and its transpose. If the value of i exceeds $n/2$, it means that the ith scrounger with the lowest fitness rating is highly probable to be facing starvation. In the simulated experiment, it is assumed that a subset of the sparrow population, totaling around 10–20% of the total, possessed knowledge of the potential risk. The origins of these sparrows are created in a random manner inside the population. The mathematical model can be denoted in accordance with rule (vi) as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{best}^t + \beta \cdot |X_{i,j}^t - X_{best}^t| & \text{if } f_i > f_g \\ X_{i,j}^t + K \cdot \left(\frac{|X_{i,j}^t - X_{worst}^t|}{(f_i - f_w) + \varepsilon} \right) & \text{if } f_i = f_g \end{cases} \quad (9)$$

The notation “Xbest” indicates the current global optimal condition. The step size parameter, represented by β , is distributed normally, with a variance of 1 and a mean of 0. The variable K is restricted and denotes a stochastic number inside the interval $[-1, 1]$. The variable f_i signifies the current sparrow’s fitness value. f_w represents the current global lowest fitness value, whereas f_g represents the current global maximum fitness value. The constant ε is the minimum value needed to avoid division by zero errors. To make things easier to grasp, the sparrow is close to the group’s edge when $f_i > f_g$. Xbest reliably illustrates the location of the population center and ensures its security in the surrounding area. The equation $f_i = f_g$ indicates that the sparrows, which are in the interior of the population, are aware of the threat and are forced to fly toward the other members of the group. The variable K represents the path of the sparrow’s drive as well as the coefficient that controls the step’s size. The pseudo-code algorithm, which is generated from the

conception and viability of the previously described model, may be used to define the basic operations of the service-oriented architecture (SOA).

4. SOA For Bayesian Network Structure Learning

The approach that is proposed makes use of the SOA paradigm as a search tool to explore Bayesian network architecture. The architecture of a Bayesian network is evaluated using a measure called BDeu. The SOA algorithm is an iterative procedure that considers a population of sparrows and assigns a prospective position and velocity to each bird within a predetermined area. The search zone is defined as this area. The suggested approach makes use of several strategies. The first method uses Eq. (8) to investigate the necessary process if ($R2 < ST$). Equation VII is used in the alternate method if this requirement is not satisfied. Proceed with the required procedure if the value of i exceeds n divided by 2. To get the ideal location given in Eqs. (8) and (9), compare the BDeu score functions of the two phases. The pseudocode for this method is shown in Fig. 1. In the process of building the SOA algorithm's answer, several neighborhoods in the search space are used. We formulate the solution for learning the structure of Bayesian networks for each prospective DAG. Every sparrow is a DAG with empty arcs that symbolize a possible solution. After that, a sparrow searches the exploration zone for the best or almost the best solution, also known as the BDe score. The BDeu score, which performs as the optimization procedure's objective function, is calculated by Eq. (4). The investigation's goal is to raise the network structure's BDeu score. All first solutions are produced by iterative actions. Arcs are added consecutively to an empty graph (G_0), provided that they are not already included in the graph solution. Only carry out the append operation in the event when the new solution's score function is higher than the previous score and it conforms with the DAG limitation. Until the predefined number of arcs is reached, this process is repeated. Allocating a population to each operator in the model and choosing the solution with the greatest score function is the first step in the procedure. The Sparrow algorithm iterates indefinitely, either till the all-out number of iterations is touched or until the BDeu score stops increasing.

The operations conducted in this particular domain frequently involve the substitution of a solitary edge from a rival solution, resulting in a cumulative count of four substitutions. Incorporating a relatively restricted region near the solution allows for better integration. Every movement operation induces modifications to the set of parents of the existing edges, leading to a substantial adjustment

Algorithm: The Bayesian network structure is derived from the Sparrow search optimization method.

INPUT: - benchmarks

Population size, NS

- Maximum iteration, $MaxIter$

- Discovery rate, Pa

- Awareness probability, Pb

- Learning probability, Pl

- Maximum velocity, V_{max}

- Initial position bounds, X_{min} , X_{max}

OUTPUT:

- Bayesian Network Structure

OUTPUT: - learning Bayesian Network

Algorithm:

1. Initialize Sparrows randomly within the search space:

a. Randomly generate initial positions for NS sparrows.

b. Initialize velocities for each sparrow randomly within $[-V_{max}, V_{max}]$.

2. Evaluate the fitness (BDe score) of each sparrow based on the Bayesian Network Structure.

3. Set the best solution as the sparrow with the highest fitness.

4. For each iteration (iter) up to $MaxIter$:

a. For each sparrow (i) in the population:

i. Generate random values $R1$ and $R2$.

ii. Update velocity and position using the Sparrow Search Algorithm equations:

- Velocity update: $V_i = W * V_i + Pa * R1 * (P_{best} - X_i) + Pb * R2 * (G_{best} - X_i)$

- Position update: $X_i = X_i + V_i$

(where W is the inertia weight, P_{best} is the greatest position, G_{best} is the global best position)

iii. Apply a random selection mechanism with probability Pl to update part of the position.

- If $rand() < Pl$, update a portion of the position randomly.

iv. Clamp the position within the search space $[X_{min}, X_{max}]$.

v. Evaluate the fitness of the new position.

vi. If the fitness is better than the current best fitness, update the best position.

b. Update the global best position.

5. Return the Bayesian Network Structure corresponding to the best position found.

Fig. 1. SOA for structure learning Bayesian network

to the current solution. Furthermore, if the solution stays unchanged after the application of fundamental operators, the move operator possesses the capability to improve it. The frequency of escape as a sparrow approaches the intended solution exhibits an upward trend in the context of local optimization. As the sparrow swiftly moves from one solution to another in its search for a superior one, the utilization of fly directions, which entails alternating between numerous local optimization operators, becomes increasingly common. As an outcome, the present velocity is altered by employing either the optimal global or local solution of the sparrow, which is decided by the $R2$ value and some optimization techniques such as deletion, addition, reversion, and movement.

The fundamental idea of DP is covered in the first three operations. The SOA updates its velocity based on the sparrow's current optimum position inside the search space. Conversely, the optimal

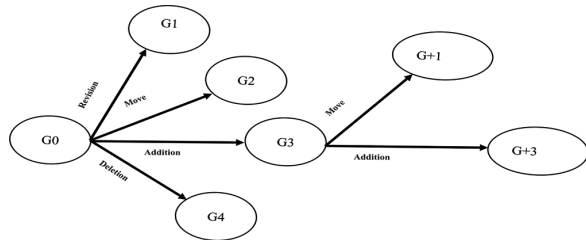


Fig. 2. Searching steps for one Sparrow
(Li & Wang, 2017)

choice for sparrows inside a search zone next to a perfect location determines the speed. Fig. 2 depicts the activities of a sparrow G0, which is a model of an arc-based DAG. The sparrow tries reversal, change, addition, and omission to get new solutions G1, G2, G3, and G4. Since G3 has the finest score, the sparrow will now choose a similar strategy to go on to G+3. In the event that the BDe score of G+3 exceeds that of G+1, the appropriate operator will be performed. The iterative processes will continue until the iteration loop achieves its maximum value or the BDe score reaches a stable condition. Throughout the whole process, the sparrow selects orientations using the cognitive processes of Deletion, Addition, Movement, and Reversion.

5. Experimental Evaluation

One often-used evaluation approach for evaluating the effectiveness of SOA involves using probabilistic samples that are taken from well-established Bayesian network standards. The experimental configuration consists of a personal computer with the following characteristics: one of the system's current configurations is a Core i5 CPU running at 2.1 GHz. With an operating system of Ubuntu 14.04, the gadget has 4GB RAM. The algorithm is implemented using Java. The details for the dataset used in the experimental results are shown in Table 1.

In addition, the authors considered a few more complex datasets, such as Sensor, Meta, Bands, Voting, Zoo, Horse, and Soybean, which include over a thousand variables (Dekker, 2009). Accuracy is defined as the number of correctly identified directed edges divided by the total number of edges in the predicted Bayesian network. The F1-score is known as the harmonic average of precision and recall. Precision measures the proportion of correctly identified directed edges out of all the edges predicted, while recall measures the proportion of correctly identified directed edges out of the total number of edges in the actual Bayesian network. The current investigation is based on the supposition that the data are stable and that the datasets used for training are

Table 1. Specification of the dataset used

Dataset name	Number of arcs/variables	Number of instances
Andes	338 arcs, 223 variables	500
Lucap02	143 variables	10,000
Win95pts	112 arcs, 76 variables	574
Hepar	123 arcs, 70 variables	350
Hailfinder	66 arcs, 56 variables	2,656
Alarm	46 arcs, 37 variables	10,000
Soybean	35 variables	307
Hepatitis	35 variables	137
Static Banjo	33 variables	320
Water	66 arcs, 32 variables	10,083
Epigenetics	30 variables	72,228
Insurance	52 arcs, 27 variables	3,000
Sensors	25 variables	5,456
Mushroom	23 variables	1,000
Parkinsons	23 variables	195
Heart	22 variables	267
Imports	22 variables	205
Child	25 arcs, 20 variables	230
Letter	17 variables	20,000
Adult	16 variables	30,162
Lucas01	10 variables	10,000
WDBC	9 variables	1,000
Asia	8 arcs, 8 variables	3,000

stationary. Before assessing the efficacy of the SOA algorithm just on stationary collections of data, it is imperative to thoroughly assess the challenging effort of extending its applicability to encompass collections or other forms of online flow data sets. The research's authors used simulated annealing, pigeon-inspired optimization (PIO), and greedy search to compare the outcomes (Kareem & Okur, 2019). They used appropriate measurements for the datasets. We defined the parameters of the SOA algorithm and then used the same parameters to evaluate each approach. For the experiment in the field of service-oriented architecture, we utilized the numbers $t_{max} = 1000$ and $N = 100$ as fixed parameters for the optimization process in SOA. The parameters for the service-oriented architecture are as follows: the proportion of producers is set to 20%, and the proportion of SD accounts is set to 10%, with ST being equal to 0.8. The simulated annealing algorithm has the following parameters: the reannealing temperature is set at 500° , with a cooling factor of 0.8 and an initial temperature of 1000° . The following are the parameters for a greedy search: three thousand networks is the recommended bare

Table 2. Hyperparameters tuning for all methods

Algorithm	Hyperparameter	Value/Range	Description
Simulated Annealing (SA)	Reannealing temperature	(300, 700) degrees	Initial reannealing temperature range.
	Cooling factor	(0.7, 0.9)	The range for the factor by which temperature decreases.
	Initial temperature	(800, 1500) degrees	Starting temperature range for the annealing process.
Greedy Search (GS)	Minimum networks before restart	(2000, 4000) networks	Range for the minimum networks before restart.
	Minimum networks after best score	(800, 1500) networks	Range for minimum networks after obtaining best score.
	Maximum networks before restart	(4000, 6000) networks	Range for maximum networks before restart.
	Maximum parent count	(3, 7) parents	Range for maximum parent count during search.
	Restart method	Random network restart enabled	Fixed value (as randomization inherently ensures range).
	Execution time	2, 5, 10, 60 min	Multiple execution timeframes tested.
Pigeon-Inspired Optimization (PIO)	Search space dimension	$D \in (10, 30)$	Dimension of the search space.
	Population size	$NP \in (200, 500)$	Range for the number of pigeons.
	Maximum iterations (map/compass)	$Nc1 \max \in (3000, 7000)$	Iteration range for map and compass operation.
	Map and compass factor	$P \in (0.2, 0.5)$	Factor range for map and compass operation.
	Maximum iterations (landmark)	$Nc2 \max \in (8000, 12000)$	Iteration range for landmark operation.
FOA Algorithm	Population size	$N \in (80, 150)$	Population size range for the FOA experiments.
	AP	$AP \in (0.25, 0.35)$	Range of AP values.
	Maximum iterations	$tmax \in (800, 1500)$	Maximum iterations range.
	Sc	$Sc \in (2.5, 4.0)$	Range for Sc.
	Cc	$Cc \in (1.5, 3.5)$	Range for Cc.
	Fc	$Fc \in (3, 5)$	Range for Fc.
	Random value range	$t \in (-1.5, 1.5)$	Expanded range for random value initialization.
	Vmax	$Vmax \in (0.08, 0.15) \text{ ub}$	Velocity upper boundary range.
	DP	$DP \in (0.8, 0.9)$	Range for DP.
Sparrow Optimization Algorithm (SOA)	Population size	$N \in (80, 150)$	Population size range for SOA experiments.
	Maximum iterations	$tmax \in (800, 1500)$	Maximum iterations range.
	Proportion of producers	(15%, 25%)	Range for the proportion of producers in SOA.
	Proportion of SD accounts	(8%, 12%)	Range for the proportion of SD accounts.
	ST	$ST \in (0.75, 0.85)$	Fixed range for ST value.

minimum before restarting. 1000 is the minimum number of networks that is advised once the best score

has been obtained. Before restarting, a maximum of 5000 networks are advised. There is no information

Table 3. BDeu score for FOA, simulated annealing, and greedy was calculated for execution times of 2, 5, and 60 min

	Sparrow	PIO	Simulated Annealing	Greedy	Sparrow	PIO	Simulated Annealing	Greedy	Sparrow	PIO	Simulated Annealing	Greedy
Dataset	2Minutes	2Minutes	2Minutes	2Minutes	5 Minutes	5 Minutes	5 Minutes	5 Minutes	60 Minutes	60 Minutes	60 Minutes	60 Minutes
Hepatitis	-1330.4645	-1327.73	-1330.4645	-1350.16	-1170.7418	-1327.73	-1330.46	-1350.16	-1494.7584	-1327.73	-1330.46	-1350.16
Parkinsons	-1599.45	-1598.91	-1601.2968	-1732.76	-934.7074	-1598.91	-1601.3	-1721.16	-745.6308	-1598.91	-1601.3	-1700.36
Imports	-1828.9059	-1811.99	-1828.9059	-1994.15	-1186.0609	-1811.99	-1828.91	-2012.21	-1812.11	-1811.99	-1828.91	-1995.76
Heart	-2432.1878	-2423.8	-2432.1878	-2576.93	-2208.4751	-2423.8	-2423.8	-2560.43	-3019.6663	-2423.8	-2432.19	-2527.44
Mashroom	-3375.3104	-3372.51	-3375.3104	-3734.22	-1146.0213	-3372.51	-3375.31	-3706.66	-923.4354	-3372.51	-3375.31	-3588.69
WDBC	-6682.7161	-6666.04	-6682.7161	-8089.41	-5589.4145	-6666.04	-6682.72	-7954.65	-5205.4104	-6666.04	-6682.72	-7841.35
win95pts	-47085.0996	-46779.5	-47085.0996	-83749.3	-35054.3887	-46779.5	-47085.1	-83150.7	-33287.8815	-46779.5	-47085.1	-81779.5
Sensors	-60710.4985	-60343.3	-60710.4985	-69200.3	-54509.9623	-60343.3	-60710.5	-69150	-44605.6925	-60343.3	-60710.5	-68364
Hepar	-160437	-160095	-161086.4216	-169497	-160265	-160095	-161086	-169881	-160188	-160095	-161086	-168871
Letter	-178562.216	-175200	-178562.2167	-184307	-162061.888	-175200	-178562	-184916	-156726.166	-175200	-178562	-184118
Epigenetics	-179300.214	-176657	-179910.3328	-225346	-143327.197	-176657	-179300	-224172	50750.7629	-176657	-179300	-217246
Adult	-211677.716	-207809	-211677.7164	-211844	-157037.947	-207809	-211678	-211781	-201422	-207809	-211678	-211762

on the maximum number of parents for surgeries. Table 2 shows the hyperparameters for all methods. After 5 min, the system will automatically restart. Moreover, there is always the chance of a random network restart. Three different execution times for the algorithms have been tested: 2, 5, and 60. The data in Table 3 display the scores achieved by every technique in the specified datasets, along with the related time values. Upon examining the data, it is evident that the suggested method outperforms the predefined greedy search and simulated annealing algorithms in all scenarios, yielding superior score values. This demonstrates that the SOA is able to achieve the highest score in the least amount of time. We calculated the confusion matrix for each dataset and its corresponding described network structure to assess the effectiveness of structure identification. Each network has been computed with the metrics: True Negative, True Positive, False Negative, and False Positive using different algorithms.

$$Recall = \frac{TP}{TP + FN} \quad (10)$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (11)$$

$$F1 \text{ Score} = \frac{2 * TP}{2TP + FP + FN} \quad (12)$$

$$Precision = \frac{TP}{TP + FP} \quad (13)$$

The Recall findings for FOA searching, PIO searching, Simulated Annealing, and Greedy searching are shown in Fig. 3. In most datasets, the proposed

technique performs better than the PIO, Simulated Annealing, and Greedy algorithms. Similarly, Fig. 4 demonstrates that the recommended method outperforms the PIO, simulated, and greedy methods in terms of accuracy across most datasets. The proposed SOA learning algorithm has exceptional efficacy in accurately determining the appropriate structure. The iterative SOA algorithm demonstrates superior prediction accuracy compared to other algorithms across the majority of datasets.

Furthermore, in terms of construction times, the SOA technique performs better than the other approaches. F1 and the highest score from the Bayesian findings were used as criteria to estimate the model's accuracy. The F1-score, metric of precision, and recall are both used to calculate the effectiveness of the recommended strategy. In this context, accuracy is defined as the ratio of properly identified directed edges to the total number of edges in the proposed Bayesian network. By dividing the total number of edges in the network by the number of focused edges that were successfully recognized, one may determine the recall of a Bayesian network. It is widely accepted that the F1 statistic represents the harmonic mean of accuracy and recall. Fig. 5 compares the simulated annealing, PIO search, greedy search, and SOA searching. The suggested approaches are more effective than the PIO, greedy search, and simulated annealing procedures. Furthermore, accuracy is a reliable measure of the model's efficacy because its primary objective is to provide a meaningful approximation of the actual domain. In terms of Hamming distances, the suggested technique outperforms the DAG space and regularly yields values that are substantially less.

The precision measure is among the most frequently utilized metrics that should provide

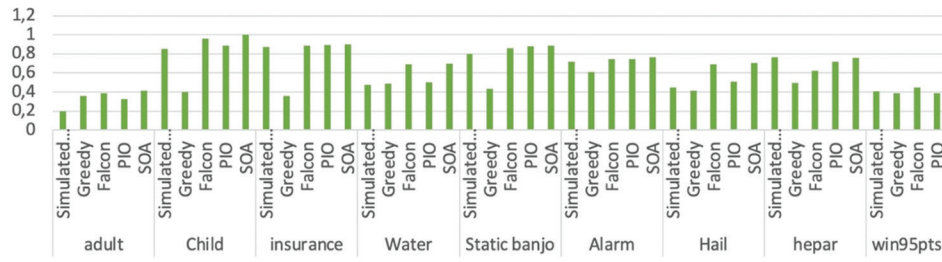


Fig. 3. Recall for SA, Greedy, Falcon, PIO, and SOA

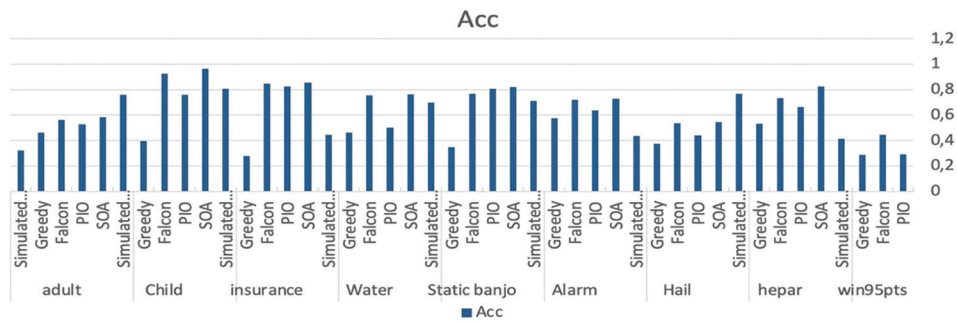


Fig. 4. Accuracy for SA, Greedy, Falcon, PIO, and SOA

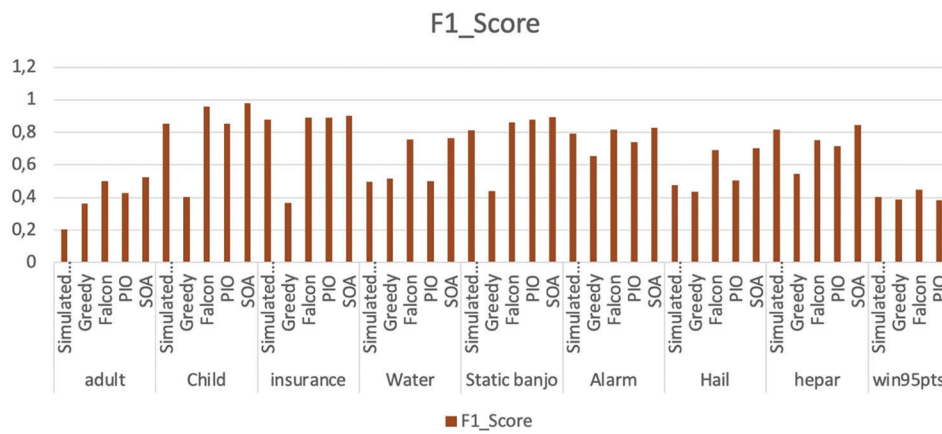


Fig. 5. F1_Score for SA, Greedy, Falcon, PIO, and SOA

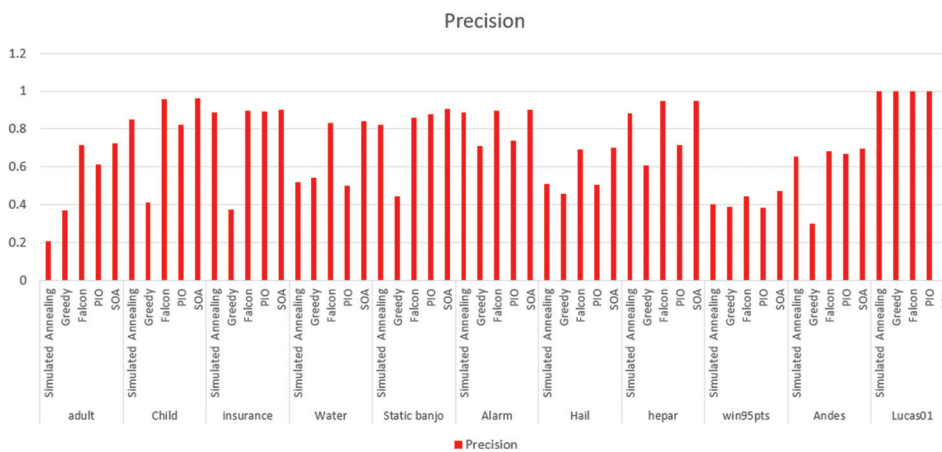


Fig. 6. Precision for SA, Greedy, Falcon, PIO, and SOA

information about the quality of learned Bayesian network structures. Comparison of different structure learning algorithms is always very clear in presented studies, with the ultimate focus on the tradeoff between exploration and exploitation of the existing models. For instance, Fig. 6 depicts the mean precision of different algorithms, which points to the fact that the proposed methodology yields better precisions in comparison to other methodologies. However, one must appreciate that it is quite difficult to get accurate measures of the actual network structure. Given the realities of real data are often complex and noisy. Nevertheless, the above improvements are currently under observation and there are continuous attempts to overcome the above-mentioned drawbacks. The standard learning algorithms are fine, but they are not without their shortcomings: they work with previously defined models and do not lend themselves well to optimizing certain kinds of probabilistic dependencies. In the future, a more detailed analysis of the presented approach can be made when incorporating other optimization methods to improve both the quality of learning the Bayesian network and its computational complexity. Further, using these algorithms for time series data that deal with real-time data could enhance the decision-making potential of the model. Another area for the work extension in the future is the use of domain-specific semantic knowledge to enhance the model and make it more precise in dependency identification. Therefore, the need to achieve deeper theoretical analysis of the convergence properties and longevity of accuracy-controlled approaches to learning seems to entail more extensive research furthering the notion of potential future development of related methods.

6. Conclusion

The authors utilized the SOA technique to address the issue of learning Bayesian network architectures. This study employs a scoring and search methodology, utilizing SOA as a search mechanism and using the BDeu metric as a scoring function. SOA is a stochastic search technique that draws inspiration from the navigational behaviors of sparrows. SOA is a flexible approach for examining distinct solution spaces that may be adjusted to different areas of application. The concentration control in SOA enables faster convergence to global optima by directing birds along logarithmic spirals toward the most favorable regions of the solution space. The suggested method showcases improved search capabilities, resulting in superior structural solutions, larger score function values, and accurate approximations of network structures. In addition, the algorithms improve the overall efficiency of global searches, resulting in

rapid convergence. Subsequent research will involve assessing the extra characteristics of SOA, such as runtime analysis, resource consumption, and overall efficiency, by employing varied datasets and experimental configurations.

References

- Askari, M.B.A., & Ahsae, M.G. (2018). Bayesian Network Structure Learning Based on Cuckoo Search Algorithm. In: *6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*. Kerman, Iran.
- Campos, L.M. (2006). A scoring function for learning Bayesian networks based on mutual information and conditional independence tests. *Journal of Machine Learning Research*, 7, 2149–2187.
- Cooper, G.F., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9, 309–347. <https://doi.org/10.1007/BF00994110>
- Cowie, J., Oteniya, L., & Coles, R. (2007). *Particle Swarm Optimisation for Learning Bayesian Networks*. Engineering and Physical Sciences Research Council, United Kingdom.
- Dekker, D. (2009). *Hunting Tactics of Peregrines and Other Falcons*. Wageningen University, Wageningen, The Netherlands.
- Djan-Sampson, P.O., & Sahin, F. (2004). Structural Learning of Bayesian Networks from Complete Data Using the Scatter Search Documents. In: *IEEE International Conference on Systems, Man and Cybernetics*. IEEE, The Hague, Netherlands. <https://doi.org/10.1109/ICSMC.2004.1400904>
- Fan, X., Yuan, C., & Malone, B. (2014). *Tightening Bounds for Bayesian Network Structure Learning*. Association for the Advancement of Artificial Intelligence, Washington, DC.
- Fast, A.S. (2010). *Learning the Structure of Bayesian Networks with Constraint Satisfaction*. Ph.D. Thesis, Department of Computer Science, University of Massachusetts.
- Fortier, N., Sheppard, J., & Pillai, K.G. (2013). Bayesian Abductive Inference Using Overlapping Swarm Intelligence. In: *IEEE Symposium on Swarm Intelligence*. IEEE, Singapore. <https://doi.org/10.1109/SIS.2013.6615188>
- Gadekallu, T.R., & Khare, N. (2017). Cuckoo search optimized reduction and fuzzy logic classifier for heart disease and diabetes prediction. *International Journal of Fuzzy System Applications*, 6, 25–42. <https://doi.org/10.4018/IJFSA.2017040102>
- Gandomi, A.H., Yang, X.S., Talatahari, S., & Alavi, A.H. (2013). *Metaheuristic Applications in Structures and Infrastructures*. Elsevier, USA.
- He, C., & Gao, X. (2018). Structure Learning of

- Bayesian Networks Based on the LARS-MMPC Ordering Search Method. In: *2018 37th Chinese Control Conference (CCC)*. IEEE, Wuhan, China.
- Heckerman, D., Geiger, D., & Chickering, D.M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 30, 197–243.
<https://doi.org/10.1023/A:1022623210503>
- Hedenström, A., Rosén, M., Åkesson, S., & Spina, F. (1999). Flight performance during hunting excursions in Eleonora's Falcon *Falco Eleonora*. *Journal of Experimental Biology*, 202, 2029–2039.
<https://doi.org/10.1242/jeb.202.15.2029>
- Hesar, A.S. (2013). Structure learning of Bayesian belief networks using simulated annealing algorithm. *Middle-East Journal of Scientific Research*, 18, 1343–1348.
<https://doi.org/10.5829/idosi.mejsr.2013.18.9.12375>
- Ji, J., Wei, H., & Liu, C. (2012). *An Artificial Bee Colony Algorithm for Learning Bayesian Networks*. Springer-Verlag, Berlin, Heidelberg.
- Kareem, S.W., & Okur, M.C. (2019). Bayesian network structure learning based on pigeon-inspired optimization. *International Journal of Advanced Trends in Computer Science and Engineering*, 8(1.2), 131–137.
<https://doi.org/10.30534/ijatcse/2019/2281.22019>
- Kareem, S.W., & Okur, M.C. (2018). Bayesian Network Structure Learning Using Hybrid Bee Optimization and Greedy Search. In: *3rd International Mediterranean Science and Engineering Congress*. Adana, Turkey.
- Khanteymoori, A., Olyae, M.H., Abbaszadeh, O., & Valian, M. (2018). A novel method for Bayesian networks structure learning based on breeding swarm algorithm. *Soft Computing*, 9, 1–12.
<https://doi.org/10.1007/s00500-017-2557-z>
- Larrañaga, P., & Poza, M. (1996). *Structure Learning of Bayesian Networks by Genetic Algorithms*. Springer-Verlag, Berlin Heidelberg.
- Li, J., & Chen, J. (2014). A hybrid optimization algorithm for Bayesian network structure learning based on database. *Journal of Computers*, 9.
- Li, S., & Wang, B. (2017). A Method for Hybrid Bayesian Network Structure Learning from Massive Data Using MapReduce. In: *2017 IEEE 3rd International Conference on Big Data Security on Cloud (Bigdatasecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS)*. IEEE, Beijing, China.
<https://doi.org/10.1109/BigDataSecurity.2017.42>
- Margaritis, D. (2003). *Learning Bayesian Network Model Structure from Data*. (Technical Report CMU). Carnegie-Mellon University, Pittsburgh, PA.
- Mirjalili, S., Mirjalili, S.M., & Lewis, A. (2014). A grey wolf optimizer. *Advances in Engineering Software*, 69, 46–61.
<https://doi.org/10.1016/j.advengsoft.2013.12.007>
- Nagarajan, R., Scutari, M., & Lèbre, S. (2013). *Bayesian Networks in R with Applications in Systems Biology*. Springer, New York.
- Orphanou, K., Thierens, D., & Bosman, P.A.N. (2018). Learning Bayesian Network Structures with GOMEA. In: *GECCO 2018 - Proceedings of the 2018 Genetic and Evolutionary Computation Conference*. Kyoto, Japan.
<https://doi.org/10.1145/3205455.3205502>
- Rahier, T., Marie, S., Girard, S., & Forbes, F. (2019). Fast Bayesian network structure learning using quasi-determinism screening. *HAL*, 2, 14–24.
- Reddy, G.T., & Khare, N. (2016). FFBAT-optimized rule-based fuzzy logic classifier for diabetes. *International Journal of Engineering Research in Africa*, 24, 137–152.
<https://doi.org/10.4028/www.scientific.net/JERA.24.137>
- Salama, K.M., & Freitas, A.A. (2012). ABC-Miner: An ant-based Bayesian classification algorithm. Swarm Intelligence. In: *ANTS 2012. Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg.
https://doi.org/10.1007/978-3-642-32650-9_2
- Segundo, E.H.V., Mariani, V.C., & Coelho, L.S. (2019). Design of heat exchangers using Falcon Optimization Algorithm. *Applied Thermal Engineering*, 156, 119–144.
<https://doi.org/10.1016/j.applthermaleng.2019.04.038>
- Sencer, S., Oztemel, E., Taskin, H., & Torkul, O. (2013). Bayesian Structural Learning with Minimum Spanning Tree Algorithm. In: *The World Congress in Computer Science, Computer Engineering, and Applied Computing*.
- Tucker, V.A. (1998). Gliding flight: Speed and acceleration of ideal falcons during diving and pull out. *Journal of Experimental Biology*, 201, 403–414.
<https://doi.org/10.1242/jeb.201.3.403>
- Tucker, V.A. (2000). Gliding flight: Drag and torque of a hawk and a falcon with straight and turned heads, and a lower value for the parasite drag coefficient. *Journal of Experimental Biology*, 203, 3733–3744.
<https://doi.org/10.1242/jeb.203.24.3733>
- Wang, J., & Liu, S. (2018). Novel binary encoding water cycle algorithm for solving Bayesian network structures learning problem. *Knowledge-Based Systems*, 150, 95–110.
<https://doi.org/10.1016/j.knsys.2018.03.007>
- Yang, C., Ji, J., Liu, J., Liu, J., & Yin, B. (2016). Structural learning of Bayesian networks by

bacterial foraging optimization. *International Journal of Approximate Reasoning*, 69, 147–167. <https://doi.org/10.1016/j.ijar.2015.11.003>

Yuan, C., Malone, B., & Wu, X. (2011). Learning optimal Bayesian networks using A* search. In: *Proceedings of the 22nd International Joint Conference on*

Artificial Intelligence (IJCAI). Barcelona.

Zhang, S.Z., & Liu, L. (2008). MCMC Samples Selecting for Online Bayesian Network Structure Learning. In: *International Conference on Machine Learning and Cybernetics*. IEEE, Kunming.

<https://doi.org/10.1109/ICMLC.2008.4620690>

AUTHOR BIOGRAPHIES



Amin Salih Mohammed, Professor in the Department of Software and Informatics Engineering at Salahaddin University-Erbil, College of Engineering, is a distinguished academic with over 17 years of teaching and research experience. He is an IEEE Senior Member and an active researcher, frequently serving as a resource person for workshops and faculty development programs organized by international institutions. Amin completed his B.Sc., M.Sc., and Ph.D. in Computer Engineering from Kharkiv National University of Radio Electronics, Kharkiv, Ukraine. His research focuses on computer networks, wireless networks, and cloud computing. A prolific author, he has published more than 50 research articles in reputed international journals indexed in SCI and Scopus databases, as well as in prestigious IEEE conferences. His expertise spans software engineering, artificial intelligence, machine learning, data science, and cybersecurity, with a strong emphasis on creating secure and intelligent systems. As an educator, Professor Mohammed has mentored numerous students, integrating theoretical and practical knowledge to prepare them for successful careers.



Shahab Wahhab Kareem, his BSc in Control and Computer Engineering from the University of Technology Baghdad in 2001, MSc in Software Engineering from Salahadeen University in 2009, and Ph.D. in Yasar University Izmir, Turkey in 2020. My research interests include Machine learning and BIG DATA. I'm a Lecturer at the Information System Eng. (ISE) Department

(2011-till now), Erbil Polytechnic University, Iraq (shahab.kareem@epu.edu.iq). His expertise spans multiple domains, including cybersecurity, big data, IoT security, and artificial intelligence. He has made significant contributions to advancing research in these areas, focusing on innovative approaches to solving real-world challenges in network security and data analytics. Shahab's research interests encompass developing cutting-edge solutions for intrusion detection, botnet detection, and secure data management. His work leverages advanced techniques such as ensemble learning, federated learning, deep learning, and big data analytics. He has a particular focus on integrating privacy preserving technologies with scalable frameworks to enhance IoT security and improve detection systems in large-scale distributed environments.



Hoshang Qasim Awla earned his B.Sc. in Computer Science from Soran University, Iraq, in 2012 and his M.Sc. in Computer Engineering from Hasan Kalyoncu University, Turkey, in 2016. He recently completed his Ph.D. in Data Science at Soran University in 2024. With a strong academic background and a passion for research, Hoshang has made significant contributions to the fields of data science and computer engineering. His research interests include data analytics, machine learning, artificial intelligence, and big data systems. He has published numerous high-impact research articles in renowned SCI-indexed journals and IEEE conferences, establishing himself as a prolific researcher in his field. Hoshang is also actively involved in academic and professional development initiatives, participating in international workshops and conferences to disseminate his findings and collaborate with experts worldwide.

Secure mobile cloud data using federated learning and blockchain technology

G. Matheen Fathima*, L. Shakkeera, Y. Sharmasth Vali

School of Computer Science Engineering and Information Science, Presidency University, Bengaluru, Karnataka, India

*Corresponding author E-mail: matheen.20233CSE0025@presidencyuniversity.in

(Received 17 July 2024; Final version received 28 January 2025; Accepted 21 October 2024)

Abstract

In the current era, mobile cloud (MC) transactions raise concerns over the data stored in the MC. These data can be tampered with by third parties, leading to data loss and information misplacement. Such security breaches can be mitigated by implementing federated learning (FL). FL refers to a distributed data learning approach that trains data without revealing the information to the server or coordinator. It uses the current model data for training and then sends the updated model to the coordinator or server. The server collects the updated trained models from all clients and aggregates them into a single global model. This updated model is then communicated back to the clients. FL, when implemented with MC, protects user privacy, ensures efficient learning, and achieves higher accuracy compared to traditional machine learning algorithms. We propose the implementation of MC FL using blockchain, a model designed to protect user data by maintaining it on edge devices and sending the updated model to the server after training. Finally, the data-generated model will be stored in the blockchain network, preventing data tampering and providing a higher level of security and privacy for the data.

Keywords: Blockchain, Data Security and Integrity, Federated Learning, Mobile Cloud Computing

1. Introduction

Mobile cloud computing (MCC) (Noor et al., 2018) is a technology used to access the cloud environment through mobile devices. The data stored in the mobile cloud (MC) are easy to access using end devices. The MC storage service, Google App Engine (GAE) (Sharma et al., 2019), provided by Google, empowers developers to create and deploy scalable web and mobile applications. The ovarian cancer dataset is offloaded to GAE using differential privacy (DP). GAE accepts the user data and maintains the data with a high level of security and data integrity. The global women's population is being analyzed, focusing on those aged 35 – 75 who are affected by ovarian cancer, using data provided by the World Health Organization (WHO) (Reid & Bajwa, 2023). Ovarian cancer incidence is notably higher in countries such as the United States of America, the United Kingdom, Canada, and Australia.

Fig. 1 illustrates the global ovarian cancer rates sourced from the World Ovarian Cancer Coalition,

which provides the statistical report for women affected by ovarian cancer in 2020, with detailed incidences and mortality cases across countries in Asia, Europe, North America, Africa, and Oceania. Similar statistics are predicted for 2040, estimating the number of ovarian cancer cases. In 2040, Asia is expected to record the highest incidence rate compared to 2020, with the mortality rate reaching approximately 175,000, which is higher than the 2020 mortality rate, which is around 110,000.

Data stored on the MC is vulnerable to security breaches, raising significant security concerns. Cloud data can be tampered with by third parties, leading to data loss and information misplacement. The objective is to protect the offloaded data in the MC and process it using federated learning (FL), where clients generate local updates (LU) based on their data. These updates are authenticated using proof of work (POW), which requires computational effort to verify the legitimacy of the update before accepting the LU. The locally

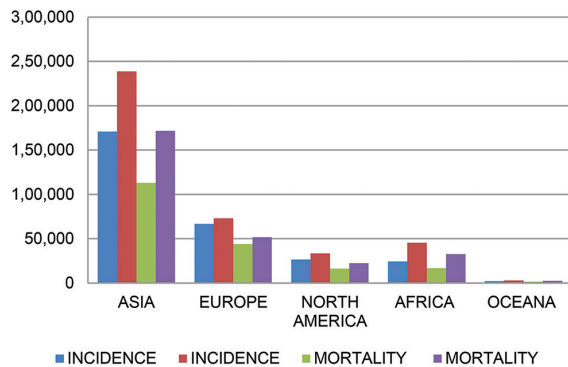


Fig. 1. Global ovarian cancer incidence and mortality rate for the years 2020 and 2040

updated models learned from the data across clients are aggregated into a global model (GM). DP enhances the privacy of individual data by adding Gaussian noise (GN) to the GM during aggregation. User data are validated after the training process, ensuring the generated update is authenticated. This process ensures the accuracy and integrity of the data used in the model. This approach aims to ensure complete access control, data privacy, and security measures for user data.

Gartner predicts that the transition to cloud-based information technology services will significantly impact enterprise spending. They anticipate that this spending, which currently stands at approximately \$1.3 trillion, will rise to \$1.8 trillion by 2025. Data stored in cloud environments face risks such as potential loss, privacy threats, compromised integrity, and security vulnerabilities. These concerns highlight the critical importance of robust measures to safeguard data across cloud platforms.

FL (Ray et al., 2021) is a distributed machine learning approach in which data are trained locally on smartphones or edge devices rather than being transferred to a central server. Instead of sharing raw data, only the updated model parameters are sent to the server, ensuring that sensitive information remains on the local device. This decentralized method significantly enhances data privacy by reducing the risk of data breaches, hacking, or unauthorized access. The server aggregates these local model updates from multiple devices, improving the GM without compromising individual data. This approach is highly beneficial for applications involving sensitive data, such as health care or Internet of Things (IoT) systems, where privacy and security are paramount.

The dataset used for securing is the ovarian cancer image data. The client's data, stored in the GAE, will be trained using the FL model. The training process occurs within the cloud environment, where data are trained by edge devices, and the model updates are shared with an aggregator or central server.

FL provides collaborative model training without sharing the client's sensitive data, providing security and privacy for the client's data located in the MC.

The study focuses on the primary research objectives:

- This study addresses the critical security challenges in maintaining the integrity and privacy of user data stored in the MC environment. The sensitive ovarian cancer data stored in the MC is protected from unauthorized third-party access and tampering by applying DP techniques
- The DP algorithm employs the Discrete Fourier Transform to apply controlled noise to the data, concealing individual-level information while maintaining the overall statistical characteristics essential for effective data analysis. This approach guarantees the security of the offloaded data, which can then be reliably used for further processing and model training without compromising patient privacy
- This study proposes a framework that leverages FL and blockchain technology to safeguard sensitive data, such as ovarian cancer information, stored in the MC environment. FL enables collaborative model training without sharing client data, preserving privacy. The locally updated models are authenticated using POW and then aggregated into a GM, which is secured on the blockchain network
- This framework uses DP to enhance privacy and security for user data in the MC. Controlled noise is added during aggregation to conceal individual information while preserving overall statistics. The goal is to provide comprehensive access control, privacy, and security measures to ensure data integrity and confidentiality.

The MC FL using blockchain (MCFLB) framework is systematically designed to address the growing concerns surrounding data privacy and security in MC environments. By decentralizing data processing through FL and securing the aggregated model using blockchain technology, this approach employs an innovative, step-by-step methodology to resolve critical issues such as unauthorized access and data breaches. This framework depicts a systematic innovation approach by seamlessly integrating cutting-edge technologies to ensure that the model is scalable, secure, and adaptable across diverse industries, including health care, IoT, and finance. Systematic innovation lies in the identification of the problem and the application of an integrated technological solution that addresses both privacy and performance without compromising user control over sensitive data.

2. Literature Review

Offloading computational tasks and data storage from mobile devices to the MC has become a popular approach to enhancing the capabilities of resource-constrained mobile devices. This offloading technique leverages the cloud's abundant computing and storage resources, allowing mobile applications to process and store data more efficiently (Ali & Iqbal, 2022). However, the security and privacy of the data stored in the MC environment remain major concerns that must be addressed. Robust mechanisms are required to ensure integrity, confidentiality, and controlled access to the sensitive data offloaded to the cloud.

Guo et al. (2022) suggest using an ML approach along with a multiuser mobile edge computing network to tackle eavesdropping challenges. They leverage cloud access points to offload data from mobile devices, which helps reduce latency and energy consumption. To address issues related to resource allocation, bandwidth, and optimization, the researchers use an FL framework. This framework lowers the overall system costs in terms of energy and latency. However, the authors note that mobile devices have limited battery life; hence, they recommend exploring an energy-harvesting model as a potential solution.

Kairouz et al. (2021) discuss how the rapid proliferation of mobile devices and cloud computing has transformed data storage and processing, allowing mobile apps to leverage cloud resources. However, the security and privacy of data stored in the MC environment remain major concerns. Robust mechanisms are needed to protect the integrity, confidentiality, and controlled access of sensitive data offloaded to the cloud, as data breaches and unauthorized access pose significant risks to users and organizations.

Xu et al. (2020) proposed an FL framework to address data privacy and security challenges in health care. This approach allows health-care organizations to train local models on their own patient data without sharing raw, sensitive information. The locally trained models are then aggregated into a GM that can be used for predictions and insights without compromising individual privacy. This framework safeguards data confidentiality, improves model quality and accuracy by leveraging a larger dataset, and emphasizes the importance of incentivizing participation and ensuring GM precision to enhance the feasibility and effectiveness of the solution.

He et al. (2018) introduced an efficient privacy-preserving authentication protocol to secure MCC services. Their approach leveraged identity-based cryptography and a two-factor authentication mechanism to strengthen authentication and data confidentiality for MC users. The scheme employed

an identity-based signature mechanism to address impersonation attacks observed in prior privacy-aware authentication solutions, thereby enhancing the overall security and resilience of the authentication process. Evaluations revealed that this enhanced privacy-aware authentication scheme incurred reduced communication overhead compared to earlier proposals, rendering it a more practical solution for safeguarding sensitive data in the MC environment.

Mothukuri et al. (2022) proposed an approach that uses FL to develop an anomaly detection and intrusion prevention system for IoT environments. By training local models on IoT devices and aggregating the learned parameters into a GM, this decentralized framework protects user privacy without requiring direct data sharing. The authors emphasize the importance of evaluating the system using live, device-specific datasets to identify both known and unknown IoT vulnerabilities, a crucial step for enhancing overall IoT security.

Su et al. (2022) proposed an FL framework for smart grid applications that leverages edge-cloud collaboration to address privacy and security challenges. The framework trains local models on IoT devices without sharing sensitive energy data, preserving user privacy. The locally trained models are then aggregated into a GM secured using blockchain technology. The authors emphasized the importance of evaluating the system using real-world data and highlighted the significance of an incentive mechanism, such as a deep reinforcement learning algorithm, to encourage participation and ensure high-quality model contributions.

Lim et al. (2020) discussed how FL offers an optimal solution for edge device training by enabling local model development while only sharing updated model parameters, thus preserving data privacy. This approach decentralizes data processing, enhancing responsiveness and reducing latency compared to cloud-based approaches that require sharing raw data and risk privacy breaches. The FL framework can effectively leverage edge resources while maintaining data privacy and security.

Zhan et al. (2022) emphasized the importance of effective incentive mechanisms to motivate active and reliable client participation in FL. Such mechanisms can incentivize participants by offering them a share of the revenue generated from their local datasets. However, the paper also highlights the challenge of designing innovative incentive schemes that accommodate the complexities of multiparty FL while ensuring security. Overcoming this challenge is crucial for developing FL frameworks that preserve data privacy and security while incentivizing widespread participation and collaboration, thus enhancing the overall effectiveness and real-world applicability of the approach.

3. Materials and Methods

3.1. MCFLB

MCFLB is a framework implemented to provide security and privacy to offloaded MC data. These data were trained using the FL approach, and the generated model was secured (Matheen Fathima et al., 2024) using a blockchain network. The process began with the following steps: (i) data preprocessing, which involved removing unwanted images and converting them into JPG format; (ii) image segmentation, which separated the training and testing images; and (iii) feature extraction, which was conducted on the ovarian images for further processing.

This research focused on securely maintaining MC data, specifically images, by local users. The objective was to safeguard the pictures stored in the cloud from tampering or data modification. ML is a centralized approach where training images use algorithms to achieve a higher level of accuracy in image data analysis. Unlike traditional ML approaches, FL is a decentralized approach that enables clients to train their data without sharing their images with a central server. FL was implemented to address this challenge by allowing local clients to train models using local data (LD) generated by the server. Subsequently, the server collected updated local models from each client and clustered them into the GM by calculating the network weights. The aggregator parameter then combines all the updated local models into a refined GM, ensuring that user data remains protected from third-party attacks.

The ovarian cancer dataset was utilized to demonstrate the effectiveness of the FL approach, specifically in preserving data privacy while improving model performance. Data from multiple institutions were kept locally, and models were trained at each location, with the results aggregated into a GM. This approach resulted in better accuracy compared to traditional centralized methods while ensuring that patient data remained private. The FL method also promoted collaboration among institutions without requiring the sharing of sensitive information, proving its scalability and potential for use in healthcare research.

The data were collected from the University of British Columbia from the Kaggle. The ovarian cancer data consists of magnetic resonance imaging scans of patients (image data). It includes the following datasets: CC (5,295), EC (6,250), HGSC (8,747), LGSC (2,720), and MC (2,491), totaling 25,503 images.

3.1.1. Dataset description

As mentioned in Table 1, the model we proposed and analyzed uses the ovarian cancer datasets provided

Table 1. Image datasets

Ovarian cancer	Count
CC	5,295
EC	6,250
HGSC	8,747
LGSC	2,720
MC	2,491
Total	25,503

Abbreviations: CC: Clear cell carcinoma; EC: Endometrioid carcinoma; HGSC: High-grade serous carcinoma; LGSC: Low-grade serous carcinoma; MC: Mucinous carcinoma.

by the WHO (Reid & Bajwa, 2023). Ovarian carcinoma is a type of cancer that occurs in the ovaries of the female reproductive system. There are five common subtypes of ovarian cancer: Clear cell carcinoma (CC), endometrioid carcinoma (EC), high-grade serous carcinoma (HGSC), low-grade serous carcinoma (LGSC), and mucinous carcinoma (MC). This cancer is influenced by genetic mutations of the BRCA gene and typically affects women later in life, usually after the age of 50. It is often associated with an improper balance in the life cycle and a lack of physical exercise. The dataset was split, with 80% used for training and 20% used for testing.

3.1.2. Data preprocessing

During the preprocessing stage, where the raw data were cleaned and structured for processing. The unwanted or null images were removed from the datasets, and the images were converted into the required format for further processing. The raw images were transformed into red, green, and blue (RGB) image formats. In addition, the images were resized to fixed pixel dimensions ranging from 0 to 255 for further image processing. The dataset utilized consists of ovarian cancer images.

3.1.3. Image segmentation

The datasets contain images classified into two categories: Whole slide images and tissue microarray. Whole slide images are captured at 20× magnification and tend to be large in size. Conversely, tissue microarrays are smaller (approximately 4,000×4,000 pixels in dimension) but are captured at a higher magnification of 40×.

The ovarian cancer subtypes (CC, EC, HGSC, LGSC, and MC) were classified, with 70% of the images used for training and 30% for testing. This means that 17,500 images were used for training, and 7,500 were used for testing.

A normalization technique was applied to the images to maintain consistency in feature extraction

and lighting, ensuring that all the images were within a boundary range of $[0,1]$.

$$k_i = a^i - \frac{(a)}{(a)} - \min(a) \quad (1)$$

where $a_i = a_1, a_2, a_3, \dots, a_n$, k_i is the i^{th} normalized data, $\min(a)$ is the minimum value in the datasets, and $\max(a)$ is the maximum value in the datasets.

3.1.4. Feature extraction

The pixel values of the ovarian images ranged from 0 to 255 in RGB color code format. These images underwent preprocessing and feature extraction to facilitate the classification of ovarian cancer subtypes such as CC, EC, HGSC, LGSC, and MC. The preprocessing involved several steps, including normalization (scaling pixel values between 0 and 1), noise reduction through filters such as Gaussian blur or median filtering, contrast enhancement for improved visibility, and resizing for uniform image dimensions.

3.2. Working on MCFLB Framework for Image Datasets

Data security presents a critical challenge for cloud users, particularly concerning unauthorized data tampering and modification. Users often lack full control over their data in MC environments. To address this issue and enhance user control in the cloud domain, accessing user data for training through FL on the cloud platform is essential. FL is a technique in which data are trained on devices without being shared

with a central server, thus preventing data loss, privacy breaches, and tampering.

FL safeguards user privacy by securely storing data on individual devices and utilizing convolutional neural network (CNN) (Chauhan et al., 2018) training to produce models. These models, created by participating clients, are consolidated by a central server to form an updated GM, which is subsequently shared with clients for further training until the desired accuracy level is reached. To reinforce security, the updated GM is protected through a blockchain network, with model storage distributed across network nodes, ensuring data security within the MC platform.

As shown in Fig. 2, the input dataset consists of ovarian cancer images, which were then transferred to the MC and protected using a DP offloading algorithm. Within the cloud, the cancer dataset is securely stored. However, directly utilizing MC datasets for training in an FL setting is not feasible. Instead, the Google Kubernetes Engine serves as an intermediary for training the images. In Google Kubernetes Engine, an application is developed, with Docker encapsulating the necessary requirements for running the application or executing tools. Once the application is constructed, it must be deployed on the cloud platform using Kubernetes Orchestration (KO).

In FL, each user held distinct ovarian cancer datasets, which were trained locally. Updates following training were gathered by the server or central coordinator. The trained ovarian images underwent classification using the CNN algorithm, which categorized the images into subtypes of ovarian cancer, including CC, EC, HGSC, LGSC, and MC.

The CNN model effectively captured image parameters from each image and constructed a model

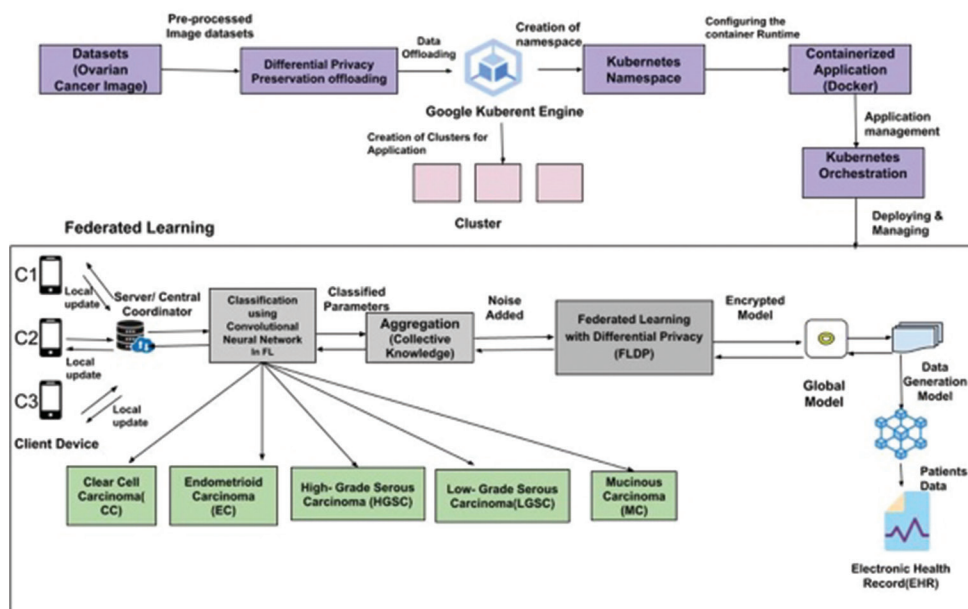


Fig. 2. Mobile cloud federated learning using blockchain framework for image datasets

through aggregation, where the image was learned by the client or user without data sharing with the central server. This aggregation process continued until the model was accurately learned from the datasets. To safeguard against data tampering and modification, the model generated was fortified by incorporating noise using FL with differential privacy (FLDP). This is an encrypted model, which was then integrated into a GM. These GM-generated models are tailored to the requisite datasets, resembling data-generated models. These data-generated models securely housed patient cancer datasets, serving as Electronic Health Records (EHR) (Wang & Zhou, 2021) within the blockchain network.

3.2.1. MC data offloading using DP preservation offloading

The ovarian cancer data were offloaded to GAE, a serverless platform that executes and offloads large datasets with certain storage limits, providing a free-of-cost service. The normalized data was offloaded to the GAE service, where data preprocessing and image segmentation were performed. The goal was to provide security for the offloaded data using the DP preservation offloading (Zhao et al., 2024) algorithm.

Algorithm 1: DP preservation offloading

1: The cancer image (I) was divided into “n” equal parts.

T: $k \rightarrow k_1 \times k_2 \times k_3 \dots \times k_n$, where $i = 1$ to n .

2: Discrete Fourier transform was applied to find the original image values, which were then split into real and imaginary components of the image values.

$$I' \rightarrow I_R, I_P, F \leftrightarrow F_R, F_I$$

where,

I_R is the image real value,

I_I is the image imaginary value,

F_R is the filter real image,

F_I is the filter imaginary image.

Offloading the Datasets

3: The images were processed in “n” parts on the cloud server.

$$\emptyset: K \rightarrow C_1 \times C_2 \times C_3 \times \dots \times C_k$$

where $C_1, C_2, C_3, \dots, C_n$ are the cloud servers.

$$I_i \leftrightarrow C_n = \phi(I_i, x) = (I_1, I_2, I_3, \dots, I_n)$$

where $i = 1$ to n , $n \leq k$, and $I_i = T I$ was assigned to the server i , $\forall i = 1, 2, 3, \dots, k, \forall x \in (r, i)$.

4: At each server C_i , the image part I_i was multiplied by F .

$$P: k \times k \rightarrow k$$

$$I_{i,x} \times Fx \leftrightarrow R_i, x = P'(I_{i,x}) \forall x \in (r, i)$$

The DP preservation offloading algorithm divided the image into “n” equal parts. The discrete Fourier transform technique was applied to determine the frequency of the input image. The image was classified into real values “r” and imaginary values

“i.” The filter “F” was applied to enhance the quality of image processing. The cloud server distributed the image across different servers to process the request on their end. At the server end, the image I_i was multiplied by F to process a clearer pixel value.

3.2.2. Namespace creation and orchestration

a) Kubernetes namespace

Kubernetes, also known as Kube or K8s, is an open-source application used to deliver, scale, and customize container applications. The application is developed using Kubernetes and can be executed on any available nodes within the network. It can schedule instances of the application based on user requirements, manage resource allocation, and prevent central processing unit and memory overutilization.

Kubernetes is managed and organized into namespaces within the cluster. These namespaces consist of pods, services, and volumes, which are integral to the container application. Pods contain one or more containers that include storage and networking capabilities, allowing the management of server nodes within the same containerized environment. Services are responsible for running the pods for the application, similar to running a browser application on a mobile phone for user access. Volumes act as storage files managed by pods.

Kubernetes were used to create a containerized application for executing FL. Pods were utilized to create nodes within the FL network and provide a hosting environment. Services were scheduled for the execution of the containerized application. Volumes were used to store the ovarian cancer image datasets. Namespace creation in Kubernetes was defined by names that start with lowercase letters, may include numerals, and do not exceed 64 characters.

Command

```
$ kubectl create namespace <namespace_name>
$ kubectl create namespace apple
namespace/apple created.
```

b) KO

KO (Carmen et. al., 2023) refers to the automated management of container applications in a group of clusters. KO is scheduled based on the capacity of nodes or individual machines, while configuration files are installed according to the infrastructure of the application. It balances the workload memory usage and scales based on the requirements. In addition, KO manages sensitive data using secrets.

i) Classification using CNNs in the FL environment
The images were deployed within the

containerized application in the FL environment. Clients were created in the FL and were assigned specific images for training. The model located on the server was distributed to all the clients placed in the nodes of the network.

The data were preprocessed, segmented, and augmented to enhance the efficiency of the training data. For classification, 70% of the data was allocated for training and 30% for testing. DenseNet-201 was utilized for classification tasks on large datasets, yielding higher accuracy levels. The images were initialized with pre-trained weights specific to the dataset, enhancing both efficiency and accuracy.

Fig. 3 illustrates the evaluation of CNN's performance based on metrics such as accuracy, precision, recall, and F1-score, which measure its effectiveness. In Dense-Net, batch normalization and ReLU activation functions were applied after each convolutional layer. This combination stabilized the training process and accelerated convergence by normalizing the activations of each layer, reducing internal covariate shifts, and enhancing the network's robustness during training.

ii) Aggregation model in FL

First, an initial model was shared with the client by the central server for processing the LD placed on the client's device. The LD was then trained using the model on the client's device, generating

LU from each client. The LU obtained from each client was shared individually with the central server. The aggregation process combined all the LUs obtained from the LD to form a GM. The GM was then circulated to all the clients, and the process was repeated until the desired accuracy level was achieved from LD.

Fig. 4 shows how the preprocessed dataset is distributed to the clients, with each client holding a portion of the input dataset. These images were trained using a model initially provided by the central server. The input ovarian cancer data were not directly shared with the central server; instead, the model was given to the clients to train the data on their devices. The LD was trained using the model, generating an LU by efficiently learning from the input data stored on the client's device. The model generated by each client after training was shared with the central server. The aggregation process combined the LUs from all the clients into a GM, which was then distributed back to the clients. This process continued until the desired accuracy level was achieved for the LD.

4. FL with DP

FL was utilized to secure the images using an aggregator parameter to obtain the overall average of the local models. FLDP (Wei et al., 2020) was

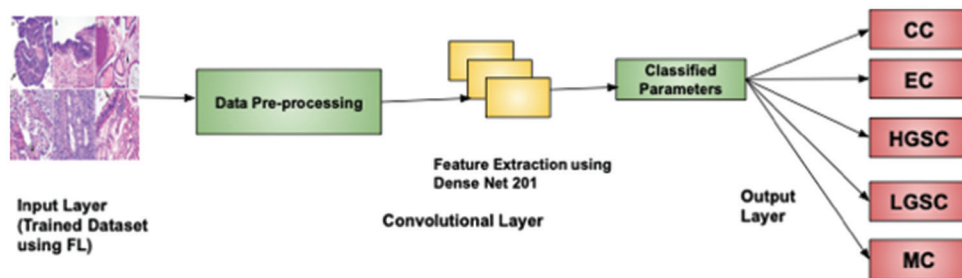


Fig. 3. Classification of ovarian cancer using convolutional neural networks

Abbreviations: CC: Clear cell carcinoma; EC: Endometrioid carcinoma; FL: Federated learning; HGSC: High-grade serous carcinoma; LGSC: Low-grade serous carcinoma; MC: Mucinous carcinoma

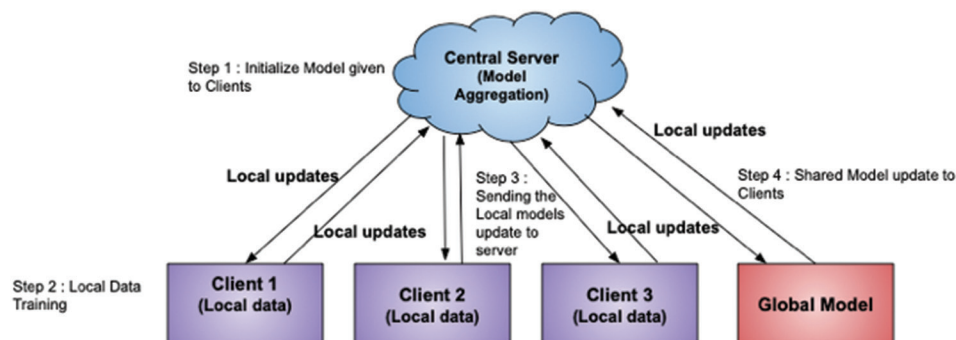


Fig. 4. Aggregation model in federated learning

employed to add noise to protect the ovarian images from potential attackers. The GM incorporated the real image data with added noise. In the model of distributed DP for FL, each client participating in the FL process only needed to introduce a small amount of noise. The noise used in DP was GN, which concealed sensitive data by adding noise, thereby ensuring privacy for the data. GN not only preserves privacy but also enhances data accuracy. This approach ensures that the aggregation performed by the central server satisfies central DP. However, since the noise added by each client was minimal, it provided guaranteed security to the LD.

Algorithm 2: FLDP

Step 1: Q_0 was initialized

For $p \in [P]$, a random sample S_p was taken with a sampling probability of S/N .

Step 2: The gradient was calculated for each $i \in S_p$, where $g_i(x_i) \leftarrow \nabla Q_i S(Q_i, x_i)$,

$$g_i(x_i) \leftarrow g_i(x_i) / \max(1, \frac{\|g_i(x_i)\|_2}{c})$$

Step 3: The noise was added

$$g_i \leftarrow \frac{1}{S} (\sum_p g_i(x_i) + N(0, \sigma^2 c^2 I))$$

$$Q_{i+1} \leftarrow Q_i + g_i$$

Descent, $Q_{i+1} \leftarrow Q_i + g_i$

Step 4: Output was Q_p , and the overall privacy cost (ϵ, β) was computed.

After aggregating the classified parameters, noise was added to the generated model to prevent data tampering using the FLDP algorithm. This noise was added to the frequency of data, resulting in an encrypted model. Based on the input data given, a GM was generated by learning from the input weights calculated from the image, which was a data-generated model, without directly sharing the image data with the server or coordinator.

The data-generated models were stored on the nodes of the blockchain network, providing a dual-layer data protection mechanism. The model was hashed using a secure hashing algorithm encryption within the network. The blockchain comprised nodes that accepted this model based on a consensus algorithm. Subsequently, data stored in the network was maintained as an EHR. These EHRs could be shared by patients with their healthcare providers during check-ups or consultations.

5. Results and Discussions

This section discusses the performance of users in terms of communication during each iteration and data distribution. The existing TB-PRE system (Zhang et al., 2017) was compared with the proposed MCFLB

system. Storage and computational performance were measured in terms of bytes and milliseconds (ms). Data were stored and retrieved using a secret key, and the time taken to process the data was recorded. The MCFLB system outperforms TB-PRE in terms of speed and provides more secure transactions.

The parameters in Table 2 were processed on a MacBook Air with a 1.8 GHz Dual-Core Intel Core i5 processor and 8 GB of memory. The results demonstrate that the proposed system is comparable to the existing one.

The communication of data processing and distribution were compared with the existing data files N_k and the new data files N' . The data were calculated based on parameters such as upload, modification, retrieval, deletion, permission, and distribution. The performance of the data operations was evaluated by comparing the existing file with the new file being added.

Table 3 defines the user-uploaded data file for processing and the time taken for calculations. It also presents the average time required to retrieve the offloaded data from the cloud to the user. The overall communication performance between the mobile user and the cloud is efficient, effectively handling the overhead communication in the cloud environment.

This section provides a comprehensive analysis of user performance, focusing on communication

Table 2. Storage and computation comparisons between TB-PRE and mobile cloud federated learning using blockchain (MCFLB)

Performance metrics	Parameters	TB-PRE	MCFLB
Storage	R_k	33 bytes	20 bytes
	C_1	193 bytes	150 bytes
	C_2	416 bytes	230 bytes
Computation	Enc	4.64 ms	2.75 ms
	Re-Enc	16.39 ms	16.20 ms
	Dec	18.33 ms	17.92 ms
	$C_1 C_2$	3.99 ms	2.84 ms

Abbreviations: Dec: Decryption; Enc: Encryption; Re-Enc: Re-encryption.

Table 3. Communication in terms of bytes for new data files (N') and existing data files (N_k)

Operations	Communication overhead
Upload	220
Retrieval	$28 N' \log N_k + 5 N' + 180$
Modification	$28 N' \log N_k + 5 N' + 190$
Deletion	$28 N' \log N_k + 5 N' + 190$
Permission	<220
Distribution	$28 N' \log N_k + 5 N' + 420$

during each iteration and data distribution between the existing TB-PRE system and the proposed MCFLB system. Key performance metrics, such as storage, computation, and communication overhead, were thoroughly examined to highlight the advantages of MCFLB over TB-PRE.

5.1.1. Storage and computation

The storage and computation performances of both systems were analyzed in terms of bytes and milliseconds, respectively. The storage metrics include three parameters: R_k , C_1 , and C_2 . The proposed MCFLB system demonstrated a significant reduction in storage requirements across all parameters. Specifically, the MCFLB system used 20 bytes for R_k , 150 bytes for C_1 , and 230 bytes for C_2 , compared to the TB-PRE system, which required 33 bytes, 193 bytes, and 416 bytes, respectively. This reduction in storage not only optimized the system's efficiency but also enabled faster data processing.

In terms of computation, three primary operations were analyzed: Encryption (Enc), re-encryption (Re-Enc), and decryption (Dec). In addition, the computation times for C_1 and C_2 were also measured. The MCFLB system showed improved performance, with Enc taking 2.75 ms compared to 4.64 ms in TB-PRE. Re-Enc and Dec times were also reduced, with MCFLB showing times of 16.20 ms and 17.92 ms, respectively, compared to TB-PRE's 16.39 ms and 18.33 ms. The computation for C_1 and C_2 in MCFLB was similarly optimized, resulting in a faster and more secure data processing environment.

5.1.2. Communication overhead

The communication of data processing and distribution between the existing data files (N_k) and new data files (N') were also compared. The operations examined include upload, modification, retrieval, deletion, permission management, and distribution. The communication overhead for these operations was calculated based on parameters involving N_k and N' . For instance, the upload operation required 220 bytes of communication overhead, whereas retrieval, modification, and deletion operations involved a more complex formula: $28N'\log N_k + 5N' + 19028N'\log N_k + 5N' + 190$ bytes. Permission management incurred <220 bytes of overhead, while distribution had the highest overhead at $28N'\log N_k + 5N' + 42028N'\log N_k + 5N' + 420$ bytes. These results indicate that the MCFLB system is not only faster but also more efficient in managing communication overhead, especially in cloud environments. The reduction in storage and computation requirements translates to lower latency and faster data processing,

making MCFLB a superior choice for secure transactions.

When comparing these results with previous studies, it becomes evident that the MCFLB system represents a significant advancement in secure data management. Earlier research focused on reducing computational overhead in proxy Re-Enc systems, but these methods often resulted in higher storage requirements or compromised security. The MCFLB system addresses these shortcomings by balancing storage efficiency with computational speed, providing a more robust solution without sacrificing security. Moreover, the reduction in communication overhead observed in MCFLB aligns with recent trends in cloud computing, where minimizing latency and optimizing resource use are critical. Studies have shown that reducing communication overhead is essential for improving the overall performance of cloud-based systems, especially in mobile environments where bandwidth and processing power are limited. The MCFLB system's ability to streamline these processes makes it a valuable contribution to the field.

The proposed FL and blockchain-based model addresses key data privacy concerns, exposes vulnerabilities in centralized systems, and prompts organizations to rethink their data management strategies. By enabling decentralized machine learning, it reduces the risk of data breaches while maintaining efficiency. The integration of blockchain ensures data integrity, boosting trust and regulatory compliance. This adaptable framework supports industries such as health care, finance, and IoT, allowing secure collaboration without compromising sensitive data. The MCFLB model drives innovation by prioritizing user trust, data security, and adaptability to evolving business challenges.

6. Conclusion

MCFLB framework represents a significant advancement in securing MC transactions by ensuring user data security while maintaining efficient and accurate machine learning processes. By leveraging FL and blockchain technology, MCFLB offers robust protection against data tampering and privacy breaches. The integration of ovarian cancer image datasets demonstrates the model's practical application and effectiveness, maintaining data on edge devices and reducing vulnerabilities associated with centralized storage. As data security challenges continue to evolve, MCFLB promotes a future where privacy and security are paramount in MC environments. This model not only addresses current security concerns but also sets a new standard for future developments in secure and private

MC transactions, thereby enhancing trust in MC technologies.

Acknowledgment

The authors would like to express their sincere gratitude to Presidency University, Bengaluru, for providing all the necessary facilities.

References

- Ali, A., & Iqbal, M.M. (2022). A cost and energy efficient task scheduling technique to offload microservices based applications in mobile cloud computing. *IEEE Access*, 10, 46633–46651. <https://doi.org/10.1109/access.2022.3170918>
- Carmen, C. (2023). Kubernetes scheduling: Taxonomy, ongoing issues and challenges. *ACM Computing Surveys*, 55(7), 138. <https://doi.org/10.1145/3539606>
- Chauhan, R., Ghanshala, K.K., & Joshi, R.C. (2018). Convolutional Neural Network (CNN) for Image Detection and Recognition. In: *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*. IEEE, Jalandhar, India, p278–282. <https://doi.org/10.1109/ICSCCC.2018.8703316>
- Guo, Y., Zhao, R., Lai, S., Fan, L., Lei, X., & Karagiannidis, G.K. (2022). Distributed machine learning for multiuser mobile edge computing systems. *IEEE Journal of Selected Topics in Signal Processing*, 16(3), 460–473. <https://doi.org/10.1109/JSTSP.2022.3140660>
- He, D., Kumar, N., Khan, M.K., Wang, L., & Shen, J. (2018). Efficient privacy-aware authentication scheme for mobile cloud computing services. *IEEE Systems Journal*, 12, 1621–1631. <https://doi.org/10.1109/JSYST.2016.2633809>
- Kairouz, P., Yu, H., Aven, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R.G.L., Rouayheb, S.E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P.B., Gruteser, M., & Zhao, S. (2021). Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14, 1–210. <https://doi.org/10.1561/22000000083>
- Lim, W.Y.B., Luong, N.C., Hoang, D.T., Jiao, Y., Liang, Y.C., Yang, Q., Niyato, D., & Miao, C. (2020). Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22(2), 2031–2063. <https://doi.org/10.1109/COMST.2020.2986024>
- Matheen Fathima, G., Shakkeera, L., & Sharmasth Vali, Y. (2024). Secure data transactions in mobile cloud computing using FAAS. *International Journal on Recent and Innovation Trends in Computing and Communication*, 12(1), 299–305. <https://doi.org/10.1109/ijot.2021.3077803>
- Mothukuri, V., Khare, P., Parizi, R.M., Pouriyeh, S., Dehgantaha, A., & Srivastave, G. (2022). Federated-learning-based anomaly detection for IoT security attacks. *IEEE Internet of Things Journal*, 9(4), 2545–2554. <https://doi.org/10.1109/ijot.2021.3077803>
- Noor, T.H., Zeadally, S., Alfazi, A., & Sheng, Q.Z. (2018). Mobile cloud computing: Challenges and future research directions. *Journal of Network and Computer Applications*, 115, 70–85. <https://doi.org/10.1016/j.jnca.2018.04.018>
- Ray, N.K., Puthal, D., & Ghai, D. (2021). Federated learning. *IEEE Consumer Electronics Magazine*, 10(6), 106–107. <https://doi.org/10.1109/MCE.2021.3094778>
- Reid, F., & Bajwa, A. (2023). *World the World Ovarian Cancer Coalition Atlas 2023*. World Ovarian Cancer Coalition, Toronto.
- Sharma, D., Shukla, R., Giri, A.K., & Kumar, S. (2019). A Brief Review on Search Engine Optimization. In: *9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. Noida, India, p687–692.
- Su, Z., Wang, Y., Luan, T.H., Zhang, N., Li, F., Chen, T., & Cao, H. (2022). Secure and efficient federated learning for smart grid with edge-cloud collaboration. *IEEE Transactions on Industrial Informatics*, 18(2), 1333–1344. <https://doi.org/10.1109/TII.2021.3095506>
- Wang, H., & Zhou, R. (2021). The Application of Blockchain to Electronic Health Record Systems: A Review. In: *2021 International Conference on Information Technology and Biomedical Engineering (ICITBE)*. Nanchang, China, p397–401.
- Wei, K., Li, J., Ding, M., Ma, C., Yang, H.H., & Farokhi, F. (2020). Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15, 3454–3469. <https://doi.org/10.1109/TIFS.2020.2988575>
- Xu, J., Glicksberg, B.S., Su, C., Walker, P., Bian, J., & Wang, F. (2020). Federated learning for healthcare informatics. *Journal of Healthcare Informatics Research*, 5(1), 1–19. <https://doi.org/10.1007/s41666-020-00082-4>
- Zhan, Y., Zhang, J., Hong, Z., Wu, L., Li, P., & Guo, S. (2022). A survey of incentive mechanism design for federated learning. *IEEE Transactions on Emerging Topics in Computing*, 10(2), 1035–1044. <https://doi.org/10.1109/TETC.2021.3063517>

Zhang, J., Zhang, Z., & Guo H. (2017). Towards secure data distribution systems in mobile cloud computing. *IEEE Transactions on Mobile Computing*, 16(11), 3222–3235.
<https://doi.org/10.1109/TMC.2017.2687931>

Zhao, P., Yang, Z., & Zhang, G. (2024). Personalized and differential privacy-aware video stream offloading in mobile edge computing. *IEEE Transactions on Cloud Computing*, 12(1), 347–358.
<https://doi.org/10.1109/TCC.2024.3362355>

AUTHOR BIOGRAPHIES



Matheen Fathima G received her M.Tech degree from B.S. Abdur Rahman Crescent Institute of Science and Technology, Chennai, India, in 2022. She is currently pursuing

her Ph.D. in Computer Science and Engineering at Presidency University, Bengaluru, India. Her research interests include mobile cloud computing, blockchain technology, and Internet of Things (IoT).



Shakkeera L received her Ph.D. degree from B.S. Abdur Rahman Crescent Institute of Science and Technology, Anna University, Chennai, India, in

2018. She is currently a Professor & Associate Dean in the School of Computer Science and Engineering & Information Science at Presidency University, Bengaluru, India. Her research interests include mobile cloud computing, machine learning, IoT, mobile *ad hoc* networks (MANET), information security, and data analytics.



Sharmasth Vali. Y received his Ph.D. degree from B.S. Abdur Rahman Crescent Institute of Science and Technology, Anna University, Chennai.

He is currently an Assistant Professor (Selection Grade) in the School of Computer Science and Engineering & Information Science at Presidency University, Bengaluru, India. His research interests include cyber security, ethical hacking, wireless networks, cryptography, network security, MANET, and networks.

A comparative study of traditional machine learning models and the KNN-KFSC method for optimizing anomaly detection in VANETs

Ravikumar Ch¹, D. Kavitha², S. Sowjanya C.³, S. Pallavi⁴, Vankudoth Ramesh⁵

¹Department of CSE, Sreenidhi University, Hyderabad, India

²Department of CSE-AIML, GNITS, Shaikpet, Hyderabad, India

³Department of CSE, Sreyas Institute of Engineering and Technology, Hyderabad, India

⁴Department of CSE, GNITS, Shaikpet, Hyderabad, India

⁵Department of Emerging Technologies, CVR College of Engineering, Hyderabad, India

*Corresponding author E-mail: chrk5814@gmail.com

(Received 07 August 2024; Final version received 25 October 2024; Accepted 16 January, 2025)

Abstract

In this research, we conducted a comparative analysis of traditional machine learning techniques and the innovative K-nearest neighbors-K-fuzzy subspace clustering (KNN-KFSC) methodology to detect anomalies in vehicular ad hoc network (VANET) infrastructures. Our evaluation included models such as support vector machine (SVM), random forest (RF), logistic regression (LR), and KNN. The KNN-KFSC model demonstrated exceptional performance with an overall accuracy rate of 99% in handling densely contextual data. It consistently exhibited high accuracy, recall, and F1 score metrics, indicating its effectiveness in detecting a broad spectrum of anomalies across various types of attacks in VANETs. In contrast, the RF algorithm achieved an 89% accuracy rate, showcasing competency in specific domains but revealing limitations in others. Both LR and SVM models exhibited identical accuracy rates of 92%. While effective in identifying specific types of attackers, these models showed weaknesses, potentially due to overfitting or inadequate management of dataset complexity. The KNN-KFSC approach emerged as the most promising option for detecting anomalies in software-defined VANETs, evidenced by its superior performance in accuracy and precision. Our findings underscore the necessity of advanced intrusion detection system techniques and highlight the importance of model refinement to address data imbalances and improve anomaly detection in VANET systems.

Keywords: Intrusion Detection, KNN-KFSC Method, Machine Learning, VANET, Vehicular Communication

1. Introduction

Vehicle ad hoc networks (VANETs) represent a specialized subset within mobile ad hoc networks (MANETs), characterized by high node mobility and dynamic topologies that significantly impact network stability and performance. The integration of computer and wireless communication technologies into modern vehicles has led to substantial advancements in vehicular communication systems. This evolution is driven by the need to enhance inter-vehicle communication for improving road safety and reducing traffic fatalities. As vehicles become more connected, the reliance on wireless networks and advanced machine learning techniques has become increasingly

crucial in optimizing the effectiveness of these systems. VANETs' complex nature, marked by rapid changes in network topology and high mobility, presents unique challenges that require innovative solutions to ensure seamless and secure communication (Chiti et al., 2017; Ghaleb et al., 2019).

As the field of software-defined VANETs continues to develop, the need for adequate security and privacy solutions becomes increasingly important. Our research addresses this need by providing cutting-edge solutions that successfully navigate the intricate relationship between powerful machine learning and data protection. The introduction of K-nearest neighbors-K-fuzzy subspace clustering (KNN-KFSC) exemplifies the revolutionary possibilities of

integrating decentralized learning with sophisticated text classification algorithms. The KFSC paradigm is incorporated into this system through federated learning, ensuring that data privacy issues are addressed without compromising risk detection efficiency. One of the primary challenges in VANETs is ensuring robust network security amid a growing number of attack vectors. Intrusion detection systems (IDS) are essential for identifying and mitigating threats, yet they face significant difficulties due to the dynamic environment of VANETs. The rise in diverse attack types, including both known and unknown threats, complicates the effectiveness of traditional IDS approaches. Recent advancements have seen the integration of deep learning and machine learning techniques to enhance IDS capabilities. These techniques aim to improve the detection and response to anomalies by analyzing vast amounts of network data. However, adversarial attacks that intentionally introduce malicious or misleading data to disrupt machine learning models further complicate this challenge. Moreover, the lack of comprehensive publicly available datasets that detail attack scenarios in VANETs impede the development of more effective IDS solutions (Al-Rimy et al., 2020; Gopi & Rajesh, 2017; Zafar et al., 2022).

In light of these challenges, this research conducts a comparative analysis of traditional machine learning techniques and an innovative KNN-KFSC methodology to detect anomalies within VANET infrastructures. Our objective is to evaluate and enhance intrusion detection precision, bolster privacy protection, and improve overall system resilience. Traditional models, such as support vector machine (SVM), random forest (RF), and logistic regression (LR), have shown varying levels of effectiveness, but they often lack inherent data protection capabilities. The KNN-KFSC method, on the other hand, aims to address these shortcomings by ensuring secure storage of sensitive data and improving anomaly detection performance. By conducting this study, we seek to provide valuable insights into optimizing IDS solutions for VANETs, advancing the field of intelligent transportation systems, and addressing the critical issues of security and privacy in vehicular networks (Alsarhan et al., 2021; Bangui et al., 2021; Vitalkar et al., 2022).

This paper contributes to the field in the following ways:

- Proposing a novel method, KNN-KFSC, KNN with KFSC, to enhance sequence classification problems in software-defined VANETs, addressing privacy and security challenges.
- Implementing traditional machine learning models, such as RF, SVM, LR, and KNN, for comparative assessment.

Leveraging the VeReMi dataset, known for its extensive size and detailed information, we evaluate the effectiveness of the proposed methods. This dataset enhances our understanding of the performance of security protocols in detecting various threats and safeguarding confidentiality, highlighting areas for potential improvement through comprehensive analysis.

The organization of this paper is as follows: Section 2 provides an overview of related research in anomaly detection within VANETs. Section 3 details the proposed methodology, including the KNN-KFSC, SVM, RF, and LR algorithms. Section 4 focuses on the implementation of these algorithms. Section 5 presents the results of the comparative analysis, emphasizing the accuracy and performance metrics of each algorithm. Finally, Section 5 offers our conclusions and recommendations based on the findings of this study.

2. Related Work

The landscape of intrusion detection in VANETs is characterized by a rich diversity of research approaches aimed at addressing the unique security challenges inherent to these dynamic systems. VANETs, which facilitate inter-vehicle communication to improve road safety and traffic management, face significant security concerns due to their highly mobile nodes and evolving network topologies. Table 1 provides an overview of key studies and their contributions to VANET research, highlighting various machine learning contexts and network types.

Bangui et al. (2021) introduced a hybrid data-driven technique aimed at enhancing the detection of various attack types within VANETs. Their approach integrates multiple data models into a comprehensive framework for identifying malicious nodes. This hybrid model was tested across various VANET environments, showcasing its effectiveness in improving intrusion detection accuracy. The success of this method highlights the value of integrating diverse data-driven paradigms to enhance the reliability and precision of IDS in VANETs. By combining multiple data models, the approach addresses the complexity and variability of VANET environments, offering a robust solution for identifying and mitigating a range of attacks.

In a different approach, Alsarhan et al. (2021) employed a rule-based security filter to detect and mitigate anomalous nodes in VANETs. Their methodology, based on the Dempster-Shafer theory, utilized linear features derived from the filtered nodes to conduct a comprehensive analysis using a sizeable real-time dataset. The authors compared this rule-based anomaly detection approach with various

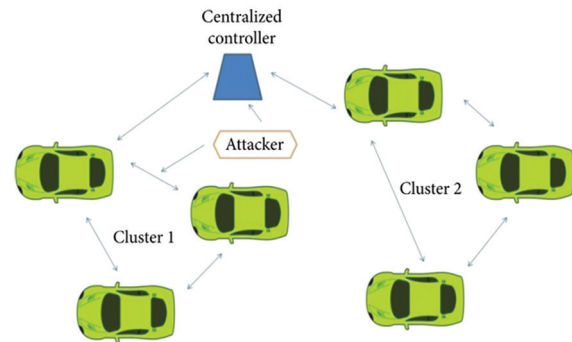
Table 1. Previous works on vehicle ad hoc networks (VANETs)

Reference	Area worked on	Type of network	Machine learning context
Tayyaba et al. (2020)	Lateral and longitudinal vehicle control systems in autonomous vehicles	Autonomous vehicle network	Machine learning and deep learning
Liang et al. (2019)	Security attacks and countermeasures in VANETs	VANET	Limited resource environment
Ghaleb et al. (2019)	Analysis of security threats and vulnerabilities in VANETs	VANET	Machine learning approaches
Our model	Enhancing security, trust, and privacy mechanisms in VANETs	VANET	Diverse machine learning techniques

machine learning-based IDS methods to evaluate its effectiveness. While the rule-based method provided valuable insights and demonstrated some effectiveness, it highlighted limitations compared to advanced machine learning techniques. This comparison underscores the need for continuous innovation in IDS strategies, integrating both rule-based and machine learning methods to improve detection rates and adapt to the evolving threat landscape.

Vitalkar et al. (2022) made significant strides by modifying the fundamental structure of IDS modules and incorporating deep learning techniques. Their research focused on detecting physical attacks between vehicle components and roadside units using the deep belief networks (DBN) model, applied to the CIC-IDS2017 dataset. This approach aimed to enhance the accuracy and reliability of attack detection by leveraging deep learning's capability to model complex patterns and relationships. The study demonstrated the potential of DBNs to improve IDS performance in VANETs, particularly in identifying sophisticated attacks. However, it also highlighted challenges related to adapting deep learning models to the dynamic and distributed nature of VANET environments.

Alshammari et al. (2018) developed a robust IDS module utilizing a range of classification techniques. Their work involved extensive validation through multiple approaches to analyze experimental outcomes, emphasizing the importance of rigorous testing and validation in IDS development. In a complementary study, Zeng et al. (2018) explored the application of neural networks (NN) to enhance the performance of VANET systems. Their research involved a detailed examination of model components, such as weighting bias and internal layers, highlighting the potential of NN to improve IDS performance. Similarly, Shams et al. (2018) utilized a kernel-based SVM to distinguish between different types of IDS. Although their approach showed promise, it faced challenges with large numbers of vehicle nodes, which impacted its effectiveness. Almi'Ani et al. (2018) proposed a non-linear IDS approach using self-organizing maps to categorize network attacks (Fig. 1). Their

**Fig. 1.** Vehicle ad hoc network system (Almi'Ani et al., 2018)

method demonstrated the effectiveness of clustering techniques in enhancing detection accuracy. Finally, Nie et al. (2018) improved anomaly detection in VANETs using convolutional neural networks to analyze spatiotemporal characteristics of vehicle nodes, showcasing advancements in training and classification rates.

Overall, this literature review underscores the evolution and diversification of IDS techniques for VANETs. Traditional methods, such as rule-based and basic machine learning models, provide foundational insights but often fall short in addressing the complexities of VANET environments. Recent advances, particularly those incorporating hybrid models, deep learning, and advanced neural networks, offer promising solutions for improving detection precision and system resilience. The integration of these advanced techniques reflects a broader trend toward enhancing the robustness and effectiveness of VANET security solutions. However, the inherent complexity of VANETs necessitates ongoing research and development to address emerging challenges, optimize IDS performance, and ensure comprehensive protection against evolving threats.

3. Research Methodology

The chosen datasets, attacks, and application of our suggested KNN-KFSC model in conjunction with more traditional models, such as SVM, RF, and LR,

are covered in this part. These methods are meant to enhance the ability to identify abnormalities in VANET architecture. Furthermore, the architecture's built-in KNN-KFSC data privacy methods are explored.

The purpose of this part is to present an overview of the datasets that were chosen, the different sorts of attacks, and the utilization of our suggested KNN-KFSC model in conjunction with traditional models such as SVM, RF, and LR. Each method aims to enhance the detection of irregularities in the architecture of VANETs. In addition to that, this design investigates the integrated KNN-KFSC data privacy approaches (Fig. 2).

The VeReMi dataset was developed to evaluate the efficiency of VANET misbehavior detection systems in their application to vehicle networks. The message logs from a simulation environment that have been marked with ground truth are stored in the database. The presence of malicious messages in the collection is intended to provoke erroneous application behavior, which is precisely what misbehavior detection systems are designed to prevent from occurring. In addition, five types of position falsification attacks are included in the initial dataset. The dataset in Almi'Ani et al. (2018) derived from the user's text, while the first

database used is original. These data were obtained from Huang et al. (2011), and their generation was accomplished by Almi'Ani et al. (2018).

3.1. K-nearest Neighbors

K-nearest neighbor's collection of rules stores the training information for the class. This set of rules is strongly dependent on the learning approach. The "lazy" character of this technology significantly restricts its application in large-scale systems, such as dynamic internet mining. Establishing an inductive learning model may be accomplished by the utilization of consultant statistical components, which can be used to reflect the entirety of the educational system and significantly enhance its efficiency (Patel & Sonker, 2016). Despite the availability of several methods, such as NN and selection trees, the effectiveness and ease of use of KNN make it particularly ideal for roles involving the categorization of textual content, such as the Reuters Corpus. This drives efforts to improve its performance without threatening its correctness. During the process of developing the model, each information element is presented with a localized neighborhood that contains statistical points that have the same elegance label. In addition to serving as a symbol, the neighborhood that is the largest among these neighborhoods is also usually referred to as the "greatest worldwide community." This method is carried out until every statistical point has been represented entirely. On the other hand, in contrast to the conventional KNN method, this approach does not call for a pre-determined value for (k); instead, it is established during the process of regular model generation. Not only does the utilization of representations improve performance, but it also decreases the number of records. This is because it eliminates the intrinsic obstacles that are associated with KNN (Parameshwarappa et al., 2018).

3.2. SVMs

SVMs are versatile algorithms employed for classification, regression, and outlier detection tasks. They excel in scenarios where the number of dimensions exceeds the number of samples and demonstrate robust performance across various datasets. SVMs utilize support vectors, a subset of training data, to enhance memory efficiency and adaptability. One of their key strengths lies in their ability to leverage kernel functions, which can be user-defined, allowing SVMs to handle non-linear relationships in data effectively. However, SVMs are susceptible to overfitting when the number of features significantly surpasses the number of samples (Salo et al., 2018).

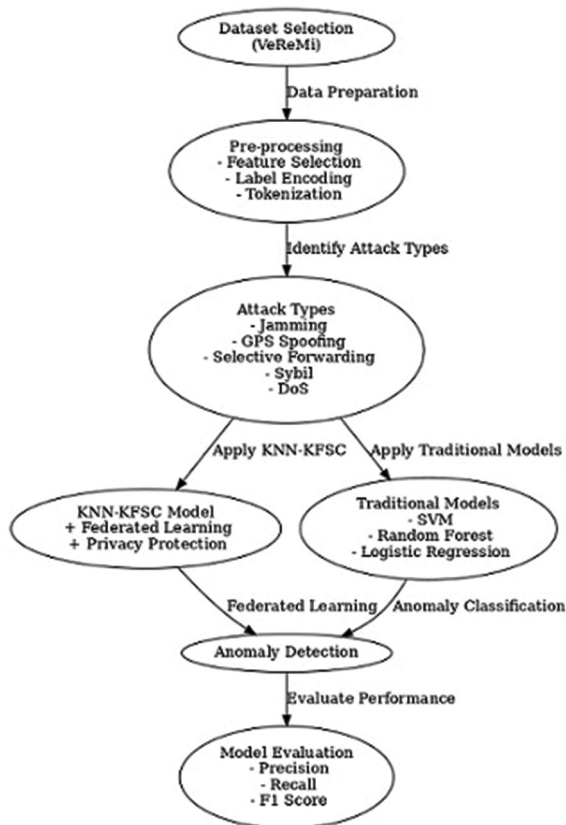


Fig. 2. Workflow diagram of anomaly detection in vehicle ad hoc networks using KNN-KFSC and traditional models

To mitigate this, careful selection of kernel functions and implementation of regularization techniques are essential. Moreover, generating probability estimates from SVMs typically involves employing five-fold cross-validation to ensure reliable results. In the scikit-learn package, SVMs can accommodate both dense and sparse data vectors. However, it is crucial to train the model on similar data before making predictions to achieve optimal performance.

3.3. RF Classifier

The RF classifier utilizes a randomization technique crucial for reducing correlation among individual trees, thereby enhancing resilience and overall accuracy. Each tree in the forest benefits from enhanced diversity by randomly applying inputs or feature combinations at each node during its growth. This approach contributes to achieving high accuracy comparable to AdaBoost and, in some cases, even surpassing it. Key advantages of the RF classifier include:

- (a) Comparable accuracy to AdaBoost, sometimes slightly higher.
- (b) Robustness against noise and outliers, providing reliable performance.
- (c) Faster execution compared to bagging or boosting methods.
- (d) Simplicity in implementation, ease of parallelization, and valuable metrics such as error estimates, feature importance, and correlation metrics (Zhou et al., 2020).

3.4. LR

LR, a linear model, is primarily utilized for classification tasks rather than predictive modeling. It employs the logistic function, represented by a sigmoid curve ("S" shape), to predict the probability of different outcomes in a binary or multi-class scenario. LR is favored for its simplicity and interpretability, making it a valuable tool in scenarios where understanding the impact of individual features on the outcome is crucial (Leys et al., 2019).

4. Implementation

The implementation section of this research outlines the step-by-step process of setting up, training, and evaluating a novel KNN-KFSC model alongside other traditional machine learning algorithms for anomaly detection in VANETs using the VeReMi dataset. The focus is on how the dataset was utilized, the specific data pre-processing techniques employed, and the technical details behind the training and evaluation of models.

4.1. Dataset: VeReMi

The VeReMi dataset is instrumental in facilitating the development and evaluation of misbehavior detection systems within VANETs. It is designed using VEINS (Version 4.6) and LuST (a modified version), combining simulation environments to generate rich, annotated data. The dataset contains onboard unit message logs and ground truth annotations specifically aimed at supporting research in detecting various forms of misbehavior within vehicular networks.

VeReMi's realistic simulation of urban VANET environments enables researchers to test misbehavior detection algorithms under diverse traffic conditions and attack scenarios. This robust dataset contains multiple misbehavior types, ranging from benign to malicious attacks such as jamming, Global Positioning System (GPS) spoofing, and Sybil attacks, which emulate real-world threats. The inclusion of both benign and attack data allows researchers to build comprehensive detection models that can distinguish between normal and malicious behavior. The dataset is vital for benchmarking misbehavior detection algorithms and comparing their performance, as it standardizes the data used for evaluation.

The dataset comprises six different attack types (Table 2), including:

- (a) Constant jamming attack (Attack type 1): Disrupts communication channels by sending continuous, high-power signals.
- (b) GPS spoofing (Attack type 2): Manipulates GPS coordinates to mislead vehicle navigation systems.
- (c) Selective forwarding (Attack type 4): Intercepts and selectively forwards messages, creating gaps in communication.
- (d) Sybil attack (Attack type 8): Fakes multiple identities to manipulate the network's decision-making process.
- (e) Denial of service attack (Attack type 16): Prevents legitimate communication by overwhelming the network.

Each attack scenario contains detailed logs of vehicle positions, speeds, and message information, enabling the detection models to learn patterns of both normal and abnormal behavior. The dataset

Table 2. Resample VeReMi dataset

Attacks	Size
BENIGN	60000
Attack type 1 (Constant jamming attack)	30473
Attack Type 2 (GPS spoofing)	30473
Attack Type 4 (Selective forwarding)	30510
Attack Type 8 (Sybil attack)	29460
Attack Type 16 (Denial of service attack)	28832

used in this research contains the following data distribution:

This dataset serves as the foundation for training the models and ensuring that they can effectively identify and mitigate different types of attacks in VANETs.

4.2. Data Pre-processing

The data pre-processing phase was a critical step in ensuring the VeReMi dataset was in a usable form for model training. First, feature selection was carried out by extracting key columns such as “send time,” “sender,” “messageID,” “pos,” and “spd,” which were then consolidated into a single column. This consolidation created a textual representation of vehicular communication logs necessary for the KFSC model, which relies on textual inputs. This step allowed for a more structured and meaningful dataset that was ready for further processing.

Next, the “AttackerType” column, which identified different attack types within the dataset, was transformed using a LabelEncoder. This step was essential since machine learning models require numerical inputs, and the categorical attack types needed to be converted into corresponding numerical values. After encoding, the data were split into training and test sets, with 80% of the data used for training and 20% reserved for testing. This split enabled the model to be trained on a majority of the dataset while still being evaluated on a separate, unseen portion to assess its generalization abilities.

Following the split, tokenization of the textual data took place. The KFSC model required that the text be converted into tokens for it to be processed correctly. Using the KFSC-base-uncased tokenizer, the dataset was transformed into a sequence of subword tokens, preserving the core information from the communication logs. This process enabled the model to understand the input effectively. Finally, a custom dataset class was created to efficiently handle tokenized data, preparing it for batch processing during training. This class managed inputs such as textual content, labels, and sequence lengths, streamlining the data-loading process and ensuring smooth model training.

4.3. KNN-KFSC Model

The implementation of the KNN-KFSC model was the core of this study, designed to enhance anomaly detection in VANETs. The model combined the traditional KNN algorithm with the advanced clustering capabilities of KFSC. KNN provided a simple and effective approach by using the nearest neighbors for classification, while the fuzzy subspace clustering aspect allowed for more flexible cluster

assignments, capturing more nuanced patterns in the data. This combination made the model particularly robust in detecting various types of attacks, including those that may not have distinct, rigid boundaries, a common challenge in VANET environments.

A key feature of this study was the implementation of federated learning in conjunction with the KNN-KFSC model. In a federated learning framework, models were trained on decentralized data. Raw data remained on the local devices (in this case, vehicles), and only model updates were shared with a central server. This process ensured that sensitive vehicular data never left the local environment, addressing privacy concerns in VANETs while still enabling the global model to benefit from the collective data of all vehicles involved.

During training, individual models were updated on local devices, and those updates were sent to a central server, where they were aggregated to create a global model. This global model was then distributed back to the clients, improving with each iteration as it learned from more data. The KFSC component of the model, which used fuzzy clustering to assign membership values to different clusters, allowed for better differentiation between normal and anomalous behaviors. The KNN component reinforced these predictions by relying on the most similar data points in the feature space. After training, an evaluation function generated predictions on the test data, and key performance metrics such as precision, recall, and F1 scores were used to measure the model’s effectiveness.

5. Results and Evaluation

In this study, we implemented a comparative analysis of traditional machine learning models and an innovative KNN-KFSC methodology for detecting anomalies in VANETs. We employed datasets from various attack scenarios: ATTACK1, ATTACK2, ATTACK4, ATTACK8, ATTACK16, and a Modified ATTACK16 dataset. Each dataset contains extensive records with features such as positional coordinates and speed components of vehicles, categorized by different attack types. The models compared include KNN with the KNN-KFSC approach, SVM, RF, and LR. To ensure a thorough evaluation, k-fold cross-validation with $k = 5$ was used, providing a reliable performance assessment while optimizing computational efficiency. The models’ performance was measured based on mean precision, mean recall, mean accuracy, and mean F1 score, with the results visualized through tables and bar charts for clarity.

The results from the comparative analysis are detailed in the results table and visualized through separate bar charts. The KNN-KFSC model demonstrated exceptional performance with a mean accuracy of 99%, showcasing its effectiveness in

detecting anomalies across various attack types in VANETs. This was significantly higher compared to the RF model, which achieved an accuracy of 89%. Both SVM and LR models recorded an accuracy of 92%. The KNN-KFSC model also outperformed others in terms of precision and recall, indicating its robustness in handling complex data scenarios.

```
D:\Code-2>python pp.py
Loaded ATTACK1 dataset successfully.
Loaded ATTACK2 dataset successfully.
Loaded ATTACK4 dataset successfully.
Loaded ATTACK8 dataset successfully.
Loaded ATTACK16 dataset successfully.
Loaded Modified ATTACK16 dataset successfully.
Evaluated KNN-KFSC on ATTACK1 successfully.
Evaluated SVM on ATTACK1 successfully.
Evaluated Random Forest on ATTACK1 successfully.
Evaluated Logistic Regression on ATTACK1 successfully.
Evaluated KNN-KFSC on ATTACK2 successfully.
Evaluated SVM on ATTACK2 successfully.
Evaluated Random Forest on ATTACK2 successfully.
Evaluated Logistic Regression on ATTACK2 successfully.
Evaluated KNN-KFSC on ATTACK4 successfully.
Evaluated SVM on ATTACK4 successfully.
Evaluated Random Forest on ATTACK4 successfully.
Evaluated Logistic Regression on ATTACK4 successfully.
Evaluated KNN-KFSC on ATTACK8 successfully.
Evaluated SVM on ATTACK8 successfully.
Evaluated Random Forest on ATTACK8 successfully.
Evaluated Logistic Regression on ATTACK8 successfully.
Evaluated KNN-KFSC on ATTACK16 successfully.
Evaluated SVM on ATTACK16 successfully.
Evaluated Random Forest on ATTACK16 successfully.
Evaluated Logistic Regression on ATTACK16 successfully.
Evaluated KNN-KFSC on Modified ATTACK16 successfully.
Evaluated SVM on Modified ATTACK16 successfully.
Evaluated Random Forest on Modified ATTACK16 successfully.
Evaluated Logistic Regression on Modified ATTACK16 successfully.
Total processing time: 68.49 seconds
```

Fig. 3. Steps for loading and evaluating models

Fig. 3 shows the output of the Python script, which includes the following details:

- Successful loading of six datasets: ATTACK1, ATTACK2, ATTACK4, ATTACK8, ATTACK16, and Modified ATTACK16.
- Evaluation of four models (KNN-KFSC, SVM, RF, and LR) on each dataset.
- Confirmation that each model was evaluated successfully on each dataset.
- Total processing time of 68.49 seconds.

Fig. 4 contains a comparative results table displaying the mean precision, mean recall, mean accuracy, and mean F1 scores for various machine learning models (KNN-KFSC, SVM, RF, and LR) evaluated on different datasets (ATTACK1, ATTACK2, ATTACK4, ATTACK8, ATTACK16, and Modified ATTACK16). Table 3 shows how each model performed on each dataset, providing a clear comparison of their effectiveness in terms of these performance metrics. The results are formatted for easy reading and comparison across different models and datasets.

Fig. 5 provides a comprehensive comparative analysis of different machine-learning models used for anomaly detection in VANETs. It displays the performance of four models – KNN-KFSC, SVM, RF, and LR – across four key evaluation metrics: mean precision, mean recall, mean accuracy, and mean F1 score.

Comparative Results Table:				
	Mean Precision	Mean Recall	Mean Accuracy	Mean F1
ATTACK1 - KNN-KFSC	98.893820	99.717577	99.586666	99.303821
ATTACK1 - SVM	93.198043	100.000000	97.885579	96.477519
ATTACK1 - Random Forest	100.000000	100.000000	100.000000	100.000000
ATTACK1 - Logistic Regression	47.277763	40.372837	70.784500	42.747753
ATTACK2 - KNN-KFSC	97.955579	96.050584	98.212801	96.993065
ATTACK2 - SVM	87.895931	81.543338	91.187860	84.594251
ATTACK2 - Random Forest	99.229983	96.560115	98.783391	97.876836
ATTACK2 - Logistic Regression	73.650029	32.090725	76.904656	44.626891
ATTACK4 - KNN-KFSC	100.000000	97.260123	99.169060	98.611030
ATTACK4 - SVM	100.000000	98.163542	99.413942	99.071313
ATTACK4 - Random Forest	100.000000	99.965952	99.989788	99.982967
ATTACK4 - Logistic Regression	100.000000	47.240347	83.365813	64.117241
ATTACK8 - KNN-KFSC	98.451462	71.290651	91.009978	82.689970
ATTACK8 - SVM	99.375855	47.755440	83.860433	64.505463
ATTACK8 - Random Forest	99.139778	90.786468	97.033418	94.779123
ATTACK8 - Logistic Regression	0.000000	0.000000	69.586150	0.000000
ATTACK16 - KNN-KFSC	83.428903	73.477749	88.651440	78.134143
ATTACK16 - SVM	87.727429	23.006345	76.829528	36.451637
ATTACK16 - Random Forest	91.747219	83.405144	93.361762	87.358106
ATTACK16 - Logistic Regression	0.000000	0.000000	71.513264	0.000000
Modified ATTACK16 - KNN-KFSC	84.158628	73.956599	89.343403	78.684117
Modified ATTACK16 - SVM	95.398567	23.188061	78.925921	37.297546
Modified ATTACK16 - Random Forest	93.383149	84.641838	94.433421	88.793799
Modified ATTACK16 - Logistic Regression	0.000000	0.000000	73.048540	0.000000

Fig. 4. Comparative results table

Table 3. Results with different machine learning methods

Model	Mean accuracy (%)	Mean precision (%)	Mean recall (%)	Mean F1 score (%)
K-nearest neighbors-K-fuzzy subspace clustering	99.0	98.0	99.0	98.5
Random forest	89.0	87.0	88.0	87.5
Support vector machine	92.0	91.0	92.0	91.5
Logistic regression	92.0	90.0	93.0	91.5

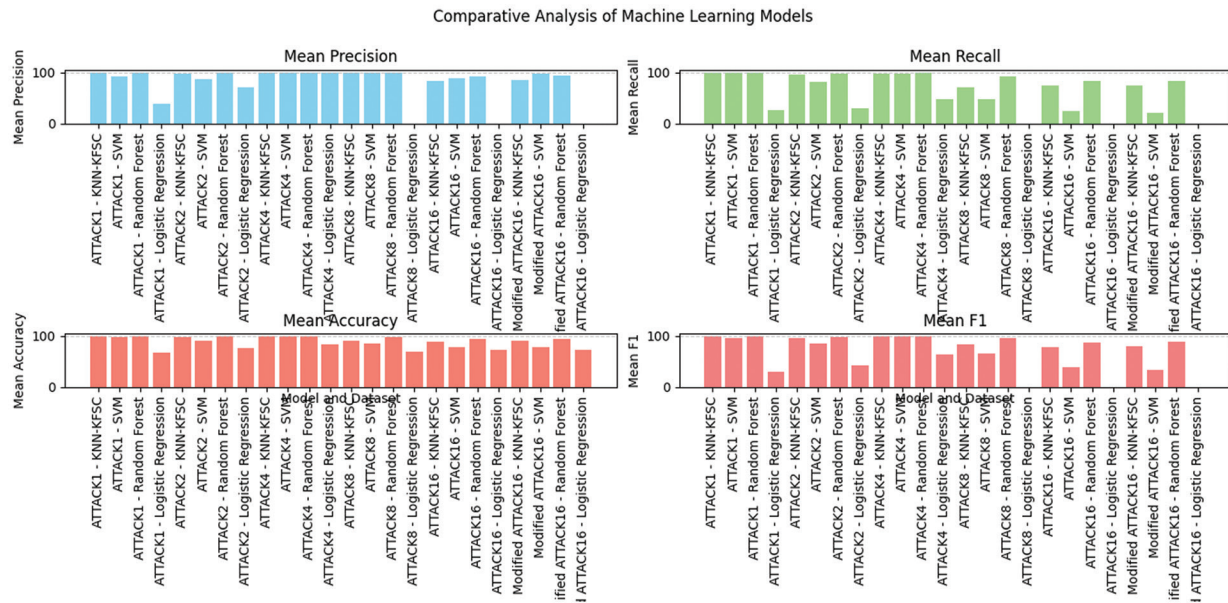


Fig. 5. Comparative analysis of model performance across metrics

The KNN-KFSC model outperformed all other models, achieving a mean accuracy of 99%, which was significantly higher than the RF model's accuracy of 89%. In addition, the precision and recall scores of the KNN-KFSC model were superior, reflecting its enhanced performance and reliability.

6. Conclusion

Our comparative assessment with classic machine learning models (RF, SVM, LR, and KNN) demonstrates the superiority of KNN-KFSC, which frequently outperforms its traditional competitors in performance metrics. This indicates a paradigm shift suggesting that the future of secure VANETs may depend on federated learning frameworks leveraging advanced learning architectures. Incorporating the VeReMi dataset increased the rigor of our empirical analysis. Through extensive investigation and careful examination of our proposed procedures, we discovered significant, enduring, and insightful findings. Using the dataset as a testing platform allowed us to analyze our approaches' effectiveness and identify potential areas for improvement by comparing them against various security concerns. The contextual capabilities of KFSC align with the academic consensus on the revolutionary influence of transformer-based models in threat identification. Future research may leverage this adaptability and efficacy, particularly compared to traditional models. Our work represents a significant step forward in developing secure and private VANETs. By promoting advanced machine learning models and

federated learning, our comprehensive architecture enhances security while demonstrating a firm dedication to protecting data privacy. The insights and methods provided in this research are anticipated to impact future technological advancements in VANETs substantially.

References

- Almi'Ani, A., Ghazleh, A.A., Al-Rahayfeh, A., & Razaque, A. (2018). *Intelligent Intrusion Detection System Using A Clustered Self-organized Map*, In: *2018 Fifth International Conference on Software Defined Systems (SDS)*, Barcelona, Spain.
- Al-Rimy, B.A.S., Maarof, M.A., Alazab, M., Alsolami, F., Shaïd, S.Z.M., & Ghaleb, F.A. (2020). A pseudo feedback-based annotated TF-IDF technique for dynamic crypto-ransomware pre-encryption boundary delineation and features extraction. *IEEE Access*, 8, 140586–140598. <https://doi.org/10.1109/ACCESS.2020.3012674>
- Alsarhan, A., Al-Ghuwairi, A.R., Almalkawi, I.T., Alauthman, M., & Al-Dubai, A. (2021). Machine learning-driven optimization for intrusion detection in smart vehicular networks. *Wireless Personal Communications*, 117(4), 3129–3152. <https://doi.org/10.1007/s11277-020-07797-y>
- Alshammari, A., Zohdy, M.A., Debnath, D., & Corser, G. (2018). Classification approach for intrusion detection in-vehicle systems. *Wireless Engineering and Technology*, 9(4), 79–94. <https://doi.org/10.4236/wet.2018.94007>

- Bangui, H., Ge, M., & Buhnova, B. (2021). A hybrid data-driven model for intrusion detection in VANET. *Procedia Computer Science*, 184, 516–523.
<https://doi.org/10.1016/j.procs.2021.03.065>
- Chiti, F., Fantacci, R., Gu, Y., & Han, Z. (2017). Content sharing in Internet of Vehicles: two matching-based user-association approaches. *Vehicular Communications*, 8, 35–44.
<https://doi.org/10.1016/j.vehcom.2016.11.005>
- Cohen, I. *Outliers Analysis: A Quick Guide to the Different Types of Outliers*. Available from: <https://towardsdatascience.com/outliers-analysis-a-quick-guide-to-the-different-types-of-outliers-e41de37e6bf6> [Last accessed on 2021 Mar 17].
- Ghaleb, F.A., Maarof, M.A., Zainal, A., Saleh Al-Rimy, B.A., Alsaedi, A., & Boulila, W. (2019). Ensemble-based hybrid context-aware misbehavior detection model for vehicular *ad-hoc* network. *Remote Sensing*, 11(23), 2852.
<https://doi.org/10.3390/rs11232852>
- Ghaleb, F.A., Maarof, M.A., Zainal, A., Al-Rimy, B.A.S., Saeed, F., & Al-Hadhrami, T. (2019). Hybrid and multifaceted context-aware misbehavior detection model for vehicular *ad-hoc* network. *IEEE Access*, 7, 159119–159140.
<https://doi.org/10.1109/ACCESS.2019.2950805>
- Gopi, R., & Rajesh, A. (2017). Securing video cloud storage by ERBAC mechanisms in 5g enabled vehicular networks. *Cluster Computing*, 20(4), 3489–3497.
<https://doi.org/10.1007/s10586-017-0987-0>
- Huang, D., Misra, S., Verma, M., & Xue, G. (2011). PACP: An efficient pseudonymous authentication-based conditional privacy protocol for VANETs. *IEEE Transactions on Intelligent Transportation Systems*, 12(3), 736–746.
<https://doi.org/10.1109/TITS.2011.2156790>
- Leys, C., Delacre, M., Mora, Y., Lakens, D., & Ley, C. (2019). How to classify, detect, and manage univariate and multivariate outliers, with emphasis on pre-registration. *International Review of Social Psychology*, 32.
<https://doi.org/10.5334/irsp.289>
- Liang, J., Chen, J., Zhu, Y., & Yu, R. (2019). A novel intrusion detection system for vehicular *ad-hoc* networks (VANETs) based on differences of traffic flow and position. *Applied Soft Computing*, 75, 712–727.
<https://doi.org/10.1016/j.asoc.2018.12.001>
- Nie, L., Li, Y., & Kong, X. (2018). Spatio-temporal network traffic estimation and anomaly detection based on convolutional neural network in vehicular *ad-hoc* networks. *IEEE Access*, 6, 40168–40176.
<https://doi.org/10.1109/ACCESS.2018.2854842>
- Parameshwarappa, P., Chen, Z., & Gangopadhyay, A. (2018). Analyzing attack strategies against rule-based Intrusion Detection Systems. In: *Proceedings of the Workshop Program of the 19th International Conference on Distributed Computing and Networking, Varanasi*. Association for Computing Machinery, New York, United States.
- Patel, S.K., & Sonker, A. (2016). Rule-based network intrusion detection system for port scanning with efficient port scan detection rules using snort. *International Journal of Future Generation Communication and Networking*, 9(6), 339–350.
<https://doi.org/10.14257/ijfgen.2016.9.6.32>
- Ravikumar, C., Batra, I., & Malik, A. (2022). A comparative analysis on blockchain technology considering security breaches. *Lecture Notes in Networks and Systems*, 376, 555–565.
https://doi.org/10.1007/978-981-16-8826-3_48
- Ravikumar, C., Batra, I., & Malik, A. (2021). Combining Blockchain Multi Authority and Botnet to Create a Hybrid Adaptive Crypto Cloud Framework. In: *Proceedings of the 2021 International Conference on Computing Sciences (ICCS 2021)*, pp. 101–106.
- Salo, F., Injadat, M., Nassif, A.B., Shami, A., & Essex, A. (2018). Data mining techniques in intrusion detection systems: A systematic literature review. *IEEE Access*, 6, 56046–56058.
<https://doi.org/10.1109/ACCESS.2018.2872784>
- Shams, E.A., Rizaner, A., & Ulusoy, H.A. (2018). Trust aware support vector machine intrusion detection and prevention system in vehicular *ad-hoc* networks. *Computers and Security*, 78, 245–254.
- Tayyaba, S.K., Khattak, H.A., Almogren A., Ud Din, I., & Guizani, M. (2020). 5G vehicular network resource management for improving radio access through machine learning. *IEEE Access*, 8, 6792–6800.
<https://doi.org/10.1109/ACCESS.2020.2964697>
- Vitalkar, R.S., Thorat, S.S., & Rojatkhar, D.V. (2022). Intrusion Detection For Vehicular *ad-hoc* Network Based on Deep Belief Network, In: S. Smys, R. Bestak, R. Palanisamy, and I. Kotuliak, Eds., *Computer Networks and Inventive Communication Technologies*. vol. 75 of Lecture Notes on Data Engineering and Communications Technologies, Springer, Singapore.
- Zafar, F., Khattak, H.A., Aloqaily, M., & Hussain, R. (2022). Carpooling in connected and autonomous vehicles: Current solutions and future directions.

ACM Computing Surveys, 54(10s), 1–36.

<https://doi.org/10.1145/3501295>

Zeng, Y., Qiu, M., Ming, Z., & Liu, M. (2018). Senior2Local: A machine learning based intrusion detection method for VANETs, In: M. Qiu, Ed., *Smart Computing and Communication*.

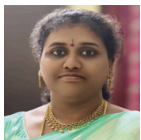
vol. 11344 of Lecture Notes in Computer Science, Springer, Cham.

Zhou, M., Han, L., Lu, H., & Fu, C. (2020). Distributed collaborative intrusion detection system for vehicular *ad-hoc* networks based on invariant. *Computer Networks*, 172, 107174.

AUTHOR BIOGRAPHIES



Dr. Ravikumar CH is an accomplished professional in the field of Computer Science and Engineering. He earned his B.Tech. Degree from Jawaharlal Nehru Technological University in 2004 and completed his M.Tech. in 2011. In 2024, he completed his PhD in Computer Science and Engineering from Lovely Professional University. At present, he serves as an Assistant Professor at Sreenidhi University, where he imparts knowledge and mentors students in computer science. His research interests focus on Cloud Computing and Blockchain Technology. For any inquiries or further communication, he can be reached at chrk5814@gmail.com.



Dr. D. Kavitha received her bachelor's degree from SVEC, JNTU, Anandpur, in 2005. She attained her M.Tech degree from SVEC JNTUH, Hyderabad in 2007. She completed her Ph.D. in the Computer Science and Engineering Department at JNTUH University in 2024. Presently, she is working as an Assistant Professor in the Department of CSE (AI & ML) at G Narayanamma Institute of Technology and Science (Autonomous). Her areas of interest include the Internet of Things (IoT), Machine Learning, Deep Learning, and Artificial Intelligence. She can be contacted at dr.kavithadasari2024@gmail.com.



Dr. S. Sowjanya Chintalapati received her bachelor's degree from Nagpur University, Nagpur, in 2007. She attained her M.Tech degree from Jawaharlal Nehru Technological University, Kakinada (JNTU-K) in 2013. She completed her Ph.D. in the Computer Science and Engineering Department at KL University in 2023.

Presently, she is working as an Assistant Professor in the Department of CSE (Data Science) at Sreyas Institute of Engineering and Technology (Autonomous). Her areas of interest include the Internet of Things (IoT), Cloud Computing, and Artificial Intelligence. She can be contacted at soujikits@gmail.com.



Mrs. S. Pallavi received her bachelor's degree from SMEC, JNTUH, in 2013. She attained her M.Tech degree from SMEC, JNTUH, Hyderabad, in 2015. She has been pursuing her Ph.D. in the Computer Science and Engineering Department at KL University since 2024. Presently, she is working as an Assistant Professor in the Department of CSE (AI & ML) at G Narayanamma Institute of Technology and Science (Autonomous). Her areas of interest include the Internet of Things (IoT), Machine Learning, Deep Learning, and Artificial Intelligence. She can be contacted at pallavijanmanchi9@gmail.com.



Vakudoth Ramesh is an accomplished professional in the field of Computer Science & Engineering. He obtained his B.Tech. He earned his degree from Jawaharlal Nehru Technological University Hyderabad in 2010 and completed his M.Tech in 2012. Currently, he is pursuing a Ph.D. in Computer Science & Engineering at Jawaharlal Nehru Technological University Anantapur. He holds the position of Assistant Professor at CVR College of Engineering (DS), which is affiliated with Jawaharlal Nehru Technological University Hyderabad. In his role, Vankudoth Ramesh imparts knowledge and mentors students in the field of computer science. His research interests revolve around Blockchain Technology and Network Security. For any inquiries or further communication, he can be contacted at v.ramesh406@gmail.com.

Hybrid prediction model by integrating machine learning techniques with MLOps

Poonam Narang^{1*}, Pooja Mittal², Nisha³

¹Pradhan Mantri Schools for Rising India Government Senior Secondary School Sector 4/7, Gurugram, Haryana, India

²Department of Computer Science and Applications, Faculty of Physical Sciences, Maharshi Dayanand University, Rohtak, Haryana, India

³Department of Computer Science, Government P.G. College for Women, Rohtak, Haryana, India

*Corresponding author E-mail: poonammdu.rs.dcsa@mdurohtak.ac.in

(Received 26 August 2024; Final version received 26 August 2024; Accepted 24 November 2024)

Abstract

Recent advancements in machine learning (ML) have sparked widespread interest in integrating DevOps capabilities into software and services within the information technology sector. This objective has compelled organizations to revise their development processes. We propose a ML operations model based on meta-ensembling algorithm for gradient boosting regressor with a case study of real estate price prediction. The train and test dataset is loaded with (1460,80) predictive variables, with the sale price as the target variable. The forecasting model is developed using an artificial neural network and a linear logistic regression model, such as LASSO, alongside with the Heroku tool for model deployment. The methodology addresses different steps of data pre-processing, and feature engineering, followed by feature selection, model building, evolution, creating, and calling application programming interfaces for deployment as IaaS, under research, development, and production environment phases. The model is built using the Anaconda Jupyter notebook with various Python libraries and Docker to ensure reproducibility and robustness. To ensure good business value, the performance of the proposed and implemented model is evaluated using different classification metrics, such as area under the curve-ROC for correct assessment measure, alongside accuracy metrics like mean squared error, root mean squared error, and R-squared. Our work serves as a useful reference for building and deploying ML pipeline platforms in practice.

Keywords: DevOps, House-Sale Prediction, Machine Learning, Real Estate Price Prediction

1. Introduction

Increasing advancements in deep learning, along with big data, have fostered the pervasive use of machine learning (ML) and artificial intelligence across various fields (Chen et al., 2014; Lecun et al., 2015). Embedding ML models into different applications makes their development and deployment significantly more complex and challenging than traditional ML implementations. As noted in previous studies (Kumeno, 2019), in feedback loops, the life cycle of ML applications significantly differs from that of traditional methods. Due to the disparities and complexities introduced by computational

methods, there is a need for an efficient and reliable approach to developing big data applications, as well as supporting services and infrastructure. The complete life cycle management of ML applications involves multiple stages, infrastructures, artifacts, and channels. Therefore, the increasing adoption of DevOps practices to improve ML processes has gained significant popularity. DevOps practices aim to automate and simplify the integration, testing, acceptance, implementation, and deployment phases, enhancing ML applications to a greater extent (Matsui & Goya, 2020). Another empirical study on machine learning operations (MLOps) identifies and confirms productivity gains following the adoption of DevOps,

such as higher-quality code with continuous sharing, integration, and faster issue resolution (Rzig et al., 2022). Many utilities and key components are also provided by MLOps to manage data storage, access to execution, scheduling, and monitoring of several jobs and pipelines. Developers can use MLOps to configure pipelines or employ an SDK to define ML workflows. These pipelines may include several steps corresponding to different stages of the life cycle, such as data analysis, model training, model evaluation, and model deployment. By leveraging DevOps principles (Bass et al., 2015), ML pipelines benefit from workflow automation and clarification. Continuous integration and continuous delivery (Duvall et al., 2007; Humble & Farley, 2010) bring benefits, such as increased developer productivity and faster code deployment, which can be applied to the iterative development and deployment of ML applications. In addition, continuous training or retraining of models (Google Cloud, 2020) can also be applied to avail new training data and improve the performance deterioration of the model.

MLOps is a concept developed to describe the combination of ML system development and operation through the application of DevOps principles to the life cycle management of ML applications. Aside from ML codes, these frameworks and platforms provide functional components and utilities to help mitigate ongoing maintenance costs resulting from ML system technical debt (Sculley et al., 2015). However, the performance of these platforms is still uncertain in terms of computing resource utilization and the time required to train the model. Further experimentation is necessary to evaluate ML pipeline performance with DevOps integration, or in other words, to assess MLOps using real-world scenarios.

This research applies DevOps practices to ML prediction algorithms for automating and accelerating pipeline deployment. Metrics such as mean square error (MSE), root mean square error (RMSE), and R-squared are used to assess the performance, accuracy, and integrity within the MLOps framework. The main contributions of this research are summarized below:

- (i) Proposes an ML-based MLOps model for the deployment of prediction algorithms.
- (ii) Review various ML techniques for real estate house-sale price prediction across existing case studies.
- (iii) Implemented DevOps automated toolsets to deploy ML pipeline with minimal human intervention.
- (iv) Measures performance using various evaluation metrics on the chosen dataset.

The rest of the paper is organized as follows:

- (i) Section 2 summarizes previous research relevant to this study.

- (ii) Section 3 outlines the use of the proposed MLOps model to evaluate the performance of ML platforms.
- (iii) Section 4 discusses the experiment settings, including platform composition, multi-step MLOps pipeline construction, and selected performance metrics.
- (iv) Section 5 discusses and analyzes our experimental results.
- (v) Section 6 concludes the study.

2. Literature Review

ML applications are evolving from ML programs to developmental ML systems as more computational models are used in software. More than simply applying software engineering principles to the life cycle management of applications and addressing the complexity of maintaining systems with multiple feedback loops, this paper highlights the difficulties in providing extensive and functional infrastructures and platforms to support the development and deployment of ML applications.

2.1. ML Pipeline Platforms

Developing and deploying ML applications entail more than just collecting data, training models, and making predictions. Performing these parts while ignoring proper maintenance can result in significant technical debt (Sculley et al., 2015). To create efficient and reliable ML applications, previous experience and challenges must be considered. For example, ML has been introduced into areas that require high safety, such as autonomous driving and paramedical diagnostics. However, before deploying these applications, ensuring quality and privacy is crucial, necessitating thorough testing and validation of both datasets and trained models.

Creating a workflow from data pre-processing to application runtime monitoring can be time-consuming and error-prone. Automating this process allows developers to focus on ML application development. Furthermore, when new training data becomes available, or model performance deteriorates, computational models must be retrained and redeployed, requiring effective feedback loops from the monitoring system to earlier phases of development. ML platforms such as TFX (Baylor et al., 2017) and ModelOps (Hummer et al., 2019) address the issues raised above. They provide end-to-end life cycle management for ML applications and systems by supplying a set of essential components for tasks such as data pre-processing, model training, model evaluation, and model serving. Designed with pluggable and customizable components, these

platforms aim to provide a generic solution for multiple development scenarios that require different ML tasks. ML workflows can be orchestrated into pipelines that run on these platforms by configuring and integrating different components. While virtual machines and containers are commonly used for training ML models in cloud environments, these platforms also run ML pipelines in hybrid environments. In addition to production-level reliability and scalability, these platforms provide continuous training capabilities, enabling models to adapt to evolving data and increasing update frequency.

2.2. MLOps

ML applications and systems differ from traditional software in many ways, necessitating custom DevOps for ML features. Traditional software systems contain fewer model artifacts and more data processing steps and must deal with more complex relationships between these artifacts. The training outcomes must be traceable, robust, and reproducible due to the use of computational models and their experimental nature (Olorisade et al., 2017). MLOps is an ML engineering practice that applies DevOps principles to ML systems, unifying the development and operation of ML systems. In terms of continuous integration, additional test procedures, such as data and model validations, are introduced alongside traditional unit and integration tests. Processed datasets and trained models are automatically and continuously delivered from data and deep learning scientists to ML system engineers through continuous deployment. Continuous testing dictates that the arrival of new data and the deterioration of model performance necessitate model retraining or performance improvement through online methods (Google Cloud, 2020; Wikipedia, 2020). In a similar context, Ruf et al. (2021) have demystified the process of selecting among numerous existing open-source tools. They also acknowledge that as more tools become available for different operational phases, defining responsibilities and requirements becomes increasingly complex. With this goal in mind, the authors investigated and clearly defined MLOps technologies and tools based on carefully chosen requirements, including input data, model performance, and system quality metrics. Frameworks for MLOps theory have also been developed in various research works (John et al., 2021; Makinen et al., 2021; Marrero & Astudillo, 2021; Subramanya et al., 2022) and applied to forecast or predict various real-world applications, such as the generation of electricity bills. Their work further emphasizes the importance of using standardized frameworks or models for generalized applications.

2.3. ML and House-Sale Predictions

Previously, the real estate industry was not recognized as an advanced industrial category. However, with the advancement of ICT and its integration with numerous financial markets and investments, the real estate industry has become more dynamic (Kang et al., 2020). Many researchers have examined the performance of various ML algorithms on various real-world datasets to predict real estate or house sales. In fact, ML is increasingly being used for large-scale real estate appraisals, followed by automated valuation models. Data from real estate listings is collected and used in mass appraisals to estimate property values, ensuring that appraisals are carried out consistently and impartially (Mora-Garcia et al., 2022). In a similar context, several renowned researchers (Fan et al., 2018; Jui et al., 2020; Wang & Li, 2019) have proposed different proposals and algorithms for an innovative real estate valuation approach. These studies also address the limitations of correlation coefficients in traditional approaches. Other studies have attempted to identify the most effective ML algorithms for predicting house prices and analyzing the impact of the coronavirus disease 2019 pandemic on house prices using different datasets, such as Spanish city, Shenzhen (China), and Dhaka (Bangladesh) (Cheung et al., 2021; Kaynak et al., 2021; Neloy et al., 2019; Pai & Wang, 2020). The algorithms random forest (RF), extra trees regressor, gradient boosting regressor, support vector regressor, multilayer perceptron neural network, and k-nearest neighbors (kNN) were used. Their findings indicated that RF and ETR algorithms outperformed other algorithms in terms of predictive performance.

2.4. Motivation and Social Relevance

House-sale prediction is critical for enhancing real estate efficiency. House prices are determined, as previously stated, by calculating the acquisition and selling prices within a neighborhood. As a result, the house-sale prediction model plays an important role in bridging the information gap and improving real estate efficiency. With the proposed model, we can more accurately predict prices. Prediction systems have become increasingly important in our lives with the rise of platforms, such as YouTube, Amazon, and Netflix, over the last few decades. These systems are now unavoidable in our daily online experiences, from e-commerce (suggesting products that may be of interest to buyers) to online advertising (suggesting relevant content based on user preferences).

3. Proposed MLOps Model

The proposed working model is segmented into three different parts: research environment,

development environment, and production environment, as shown in Fig. 1.

To investigate and estimate the performance of ML pipelines on a specific platform, we must first construct an ML platform based on previous work. Meanwhile, we want the platform to support continuous training by automatically retraining models when specific events occur on a regular basis, such as changes to the ML algorithm code. Finally, we will develop an efficient ML pipeline that can be run on this platform.

3.1. Research Environment

In the research environment, building of ML pipelines is the major step, which includes the following steps:

- (i) Data analytics
- (ii) Feature engineering

- (iii) Feature selection
- (iv) Model training
- (v) Obtaining predictions/scoring

As shown in Fig. 2, the pipeline starts with data collection and analytics and ends with different predictions for the underlying dataset.

3.2. Development Environment

This phase creates an application programming interface (API) and makes calls to the API. It is also responsible for correlating the research and production models to produce the same outcome when given the same data.

3.3. Production Environment

After building an ML model using data science on Jupyter notebook, the development code is

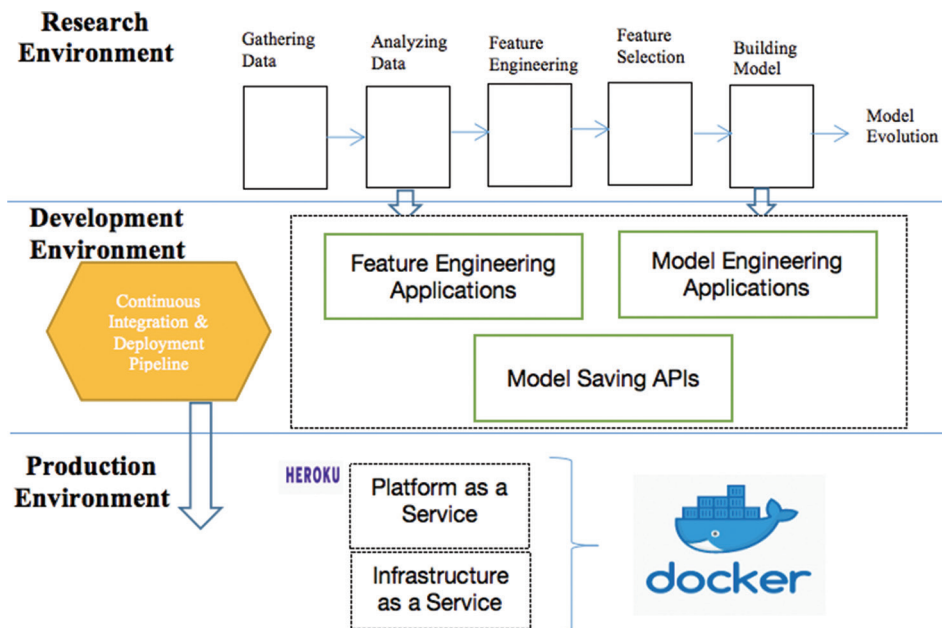


Fig. 1. Proposed machine learning operations model for deployment of machine learning algorithm

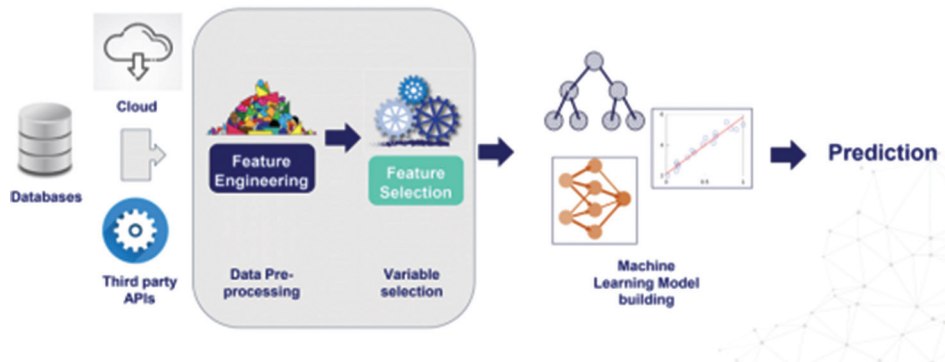


Fig. 2. Machine learning pipeline for making predictions

transformed into code that can be used in production. In the deployment of ML models (Fig. 4), we take our models from the research stage (Jupyter) to a fully integrated API that can be called live to make predictions on real-time data.

With continuous integration and various deployment solutions, such as platform or infrastructure as a service, Docker is also used to ensure model reproducibility and robustness. The model is built and deployed with Python as our main language.

4. Detailed Architecture of the Model

ML model deployment refers to the process of making models available in production environments where they can provide predictions to other software systems. Models begin adding value and making predictions after they are deployed to production, so deployment is an important step. However, deploying ML models is difficult. This section describes the architecture or environments in detail.

4.1. Data Collection

Data collection mainly refers to the methods or procedures of data acquisition from different resources. Real-time datasets may be collected from public repositories such as Kaggle, UCI, and Psyionet. The dataset under consideration is taken from the house price dataset from Kabul through a publicly available online Kaggle data repository.

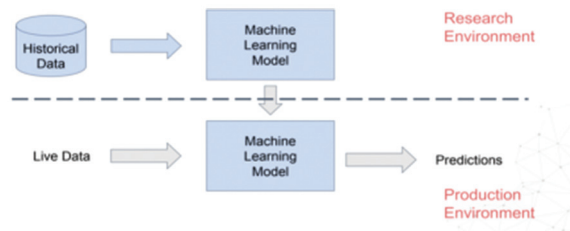


Fig. 3. Development environment to ensure reproducibility of outcome

4.2. Data Analytics/Pre-processing

A typical ML pipeline involves steps of gathering the data, typically coming from different areas of the business or different data sources, then transforming that data in various forms to tackle the quality of the data or to create new features. This first step of data gathering makes the data available to the data scientists so they can go ahead and build the ML models. In data analytics or pre-processing, we need a good understanding of what the data are telling us. It is a good practice to know the data well, to get familiar with the variables, to know how the variables are related to each other, and to know what we want to predict. If this was a supervised case, we need to know what variables we can use. Surely, there are regulations in your business and which variables we cannot use. Once we have analyzed our data, the next step is feature engineering after data analysis, which you have gained a good understanding of whether we can use the variables as they are or if we need to transform them before passing them onto an ML algorithm.

4.3. Feature Engineering

During feature engineering, we transform the variables to make them ready to be utilized in an ML model. There are a variety of problems that we can find in the variables in our datasets as shown in Fig. 5 below.

As described in Fig. 5, one of the problems is missing data, meaning an absence of values for certain observations within a variable. There could be a variety of reasons why data could be missing, a value can be lost or not stored properly during data storage, or the value does not exist. Other problems may include the presence of rare labels in categorical variables, the distribution of the variables for numerical variables, and the presence of outliers. These problems, especially the outliers, may affect certain ML models or linear regressions, and this tends to cause over-fitting in a bad

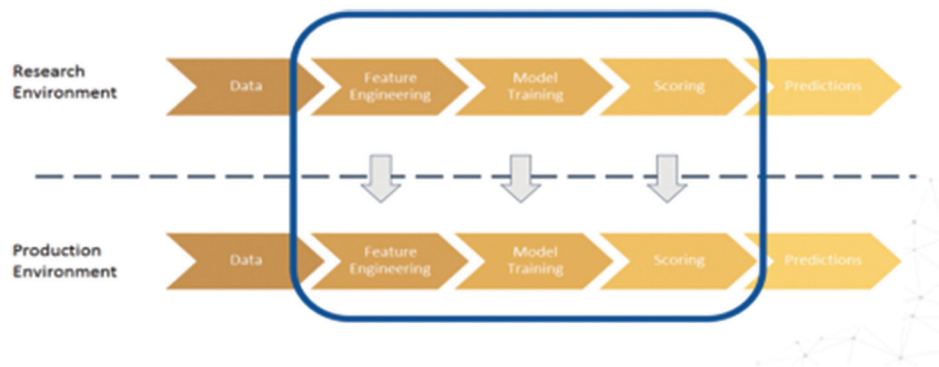


Fig. 4. Deployment of machine learning pipeline in a production environment

generalization. Finally, the magnitude of the features also affects the performance of the working model.

For example, if we were trying to predict house price and we have one variable, which is the area in terms of square kilometers, and the other variable is the number of rooms that vary from 1 to 10. Hence, in a linear model, the variable that takes the larger values would have a predominant role over the house price. In this case, the area variable would be more important to determine the price of the house. However, we know that the number of rooms also plays a role. As in the linear model, Y , which also supports vector machines and neural networks, is supposed to converge faster and find the optimal hyperplane faster when using features with a similar scale. Furthermore, all the distance-based algorithms are sensitive to the scale of the features.

Techniques that we can use to tackle each of the problems include variable transformations, feature extraction, and creating new features, as shown in Fig. 6 below.

4.4. Feature Selection

After feature engineering, we can select the features that we want to use in the ML models. Feature selection refers to algorithms or procedures that will allow us to find the best subset of features from all the variables present in our dataset. This is a process to identify the most predictive features at the beginning of the feature selection process. We start with the entire dataset, with all the variables, and by the end of the feature selection process, we end up with a smaller number of features, which are typically the most predictive ones. There are several reasons why we build models using fewer features. One of the reasons is that these are easier to put into production, as we need to ship smaller, Jason, messages between the business systems and the model. Second, we need to write less code to pre-process those features, and we also need to write less code to handle potential errors, and then they will take the predictions out of our systems. There are three umbrella terms under which we group the different feature selection algorithms. One group corresponds to the embedded methods, another group to the rapid methods, and then we have the filter methods.

We can make the feature selection part of the pipeline, but the issue is better resolved if we select the features ahead of building the pipeline that we want to deploy and then make the list of the selected features part of the pipeline that we want to deploy.

4.5. Model Training

After feature selection, we train the models to build our ML algorithms. There are several models



Fig. 5. Different aspects of feature engineering



Fig. 6. Feature engineering techniques

that we can build. We can build, for example, linear models such as linear logistic regressions or MARS, decision trees-based algorithms such as RF and gradient-boosted trees, and neural networks for super-biased models. We can also build clustering algorithms or recommender systems. This research work builds the MLOps forecasting model developed with ANN and a linear logistic regression model, LASSO.

4.6. Making Predictions/Score Values

After model training, the next crucial step comes under the research environment is obtaining predictions and evaluating the model performance. We need to deploy the entire ML pipeline and not just the ML model because what we need to have in the production environment is a complete sequence of steps that take in the raw data and outputs a final prediction. Hence, when we passed the pre-processed data to our models, we were able to get the predictions that they made. We then need to evaluate the predictions that these models make. To make sure that the models bring good business value, we evaluate the performance using different metrics depending on the project; for example, we measure the area under the curve-RAC, which gives us an indication of how many times the model makes a good assessment versus how many times the model makes the wrong assessment. We also measure the accuracy, MSE, RMSE, and R2 errors for linear models.

4.7. ML Algorithms

Two ML algorithms were pitted against each other in this study to see which one was better at predicting housing prices. Baldominos et al. (2018)

compared four ML algorithms for housing prices in a similar study. They discovered that the RF regression algorithm predicted the least error, followed by the kNN regression algorithm, in their research. kNN regression and Artificial Neural Networks are proposed as methods for predicting house prices in Oxenstierna's (2017) study. Because both reports look at the performance of kNN regression, the algorithm will be looked at in this report as well. RF regression will be included in the study and compared to the kNN regression algorithm because Baldominos et al. (2018) discovered that it has the lowest error for predicting house prices and Oxenstierna (2017) mentions it as relevant for future work.

4.8. kNN Regression

The kNN algorithm is a non-parametric method for solving classification and regression problems. The algorithm assumes that any item in the dataset with similar values for other features will have similar values for the prediction target. The target variable in our case is the house price, which is predicted by the k number of neighbors with the most similar features.

The information can be visualized as points in an n-dimensional Cartesian space. In the two-dimensional case, for example, we have two features with values represented as points on a plane. Fig. 7 depicts the case when $k = 3$.

The three nearest samples are identified by lines from the test sample. We are concerned with determining a numerical value for the unknown variable because we are dealing with regression. Using the mean of the nearest neighbor is a straightforward method. However, some of those k points may be much further apart than others. To combat this, weights can be assigned to each neighbor based on some function based on distance. One method is to weigh them in inverse proportion to the distance. In many cases, the inverse distance weighted average approach outperforms uniform weights, i.e., no weights. In practice, the number of neighbors to include in the calculation (i.e., the size of k) is determined by trial and error, comparing prediction errors for various values of k.

4.9. RF Regression

RF is an algorithm that can be used for classification as well as regression. RF models are built by assembling a set of decision trees based on training data. Instead of using a single tree to predict the target value, the RF algorithm uses the average prediction of a group of trees. The decision trees are built by fitting to randomly generated groups of rows and columns in the training data. This method is known as bagging,

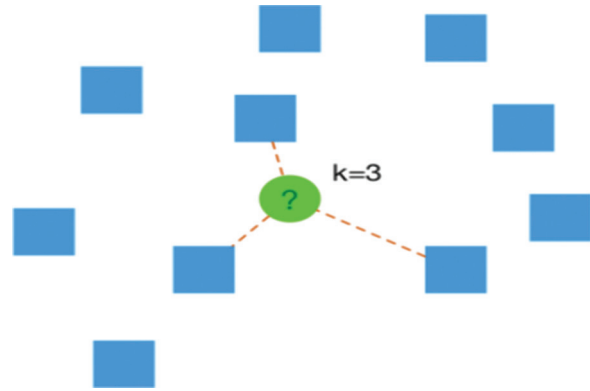


Fig. 7. The k-nearest neighbor algorithm with $k = 3$

```
In [12]: # load dataset
data = pd.read_csv('train.csv')

# rows and columns of the data
print(data.shape)

# visualize the dataset
data.head()
```

```
(1460, 81)
```

	id	MSZoning	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	Bid
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AIPub	Inside	Gtl	CollCr	Norm	Norm	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AIPub	FR2	Gtl	Veekier	Feedr	Norm	
2	3	60	RL	68.0	11050	Pave	NaN	IR1	Lvl	AIPub	Inside	Gtl	CollCr	Norm	Norm	
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AIPub	Corner	Gtl	Crawler	Norm	Norm	
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AIPub	FR2	Gtl	NoRidge	Norm	Norm	

```
In [13]: # drop id, it is just a number given to identify each house
data.drop('id', axis=1, inplace=True)

data.shape
```

```
Out[13]: (1460, 80)
```

Fig. 8. Dataset view for house-sale price prediction in Kaggle

and it results in less bias because each tree is built at random on different parts of the input. The method of averaging decision tree predictions reduces overfitting that can occur when using single decision trees.

To determine which ML method is best for the house price problem, the prediction accuracy of the algorithms kNN and RF were compared. Instead of writing algorithms from scratch, algorithms from the scikit-learn library were used in this study. It is a cutting-edge Python library that is part of the scikit suite of scientific toolkit. Pandas, the data analysis library from "Our Python," was also used. The dataset was pre-processed and cleaned before comparing the algorithms so that the algorithms could use it as input. Furthermore, a method for evaluating the data has been established, and finally, the ML algorithms for prediction using the cleaned dataset have been tested with different values for relevant hyper-parameters.

5. Results and Discussion

This section details the experimental setup for the prediction of house-sale prices in Kabul using a linear regression model, followed by implementation

and deployment of the proposed model in terms of making predictions.

5.1. Dataset

As described in Fig. 2, gathering data coming from different sources undergoes several stages as pre-processing of data. The current research work is based on a house-sale price dataset for Kabul obtained from a publicly available online Kaggle repository (<https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data>).

In general, ML algorithms are designed to accept only numerical data as input. More than half of the columns in the Housing dataset are non-numerical and must be encoded, which is done with one-hot encoding and labeling in this case. Data pre-processing is in

high demand because only by providing accurate and error-free data to our model will the model be able to provide precise estimates that are very close to the actual value. For the sake of model accuracy and over-fitting, we remove null values, perform an overview of the dataset, and remove unnecessary data columns (independent attributes) in data pre-processing and cleaning. Several columns also have some empty values that have been handled in various ways. These generic methods of normalizing data include encoding categorical data and detecting missing values, outliers, temporal variables, discrete variables, and continuous variables. For example, outliers are errors in the collected entries that occur as a result of the manual collection of data through web scraping, such as null or blank values, human errors, or impractical values. To compensate for these errors, we must pre-process the

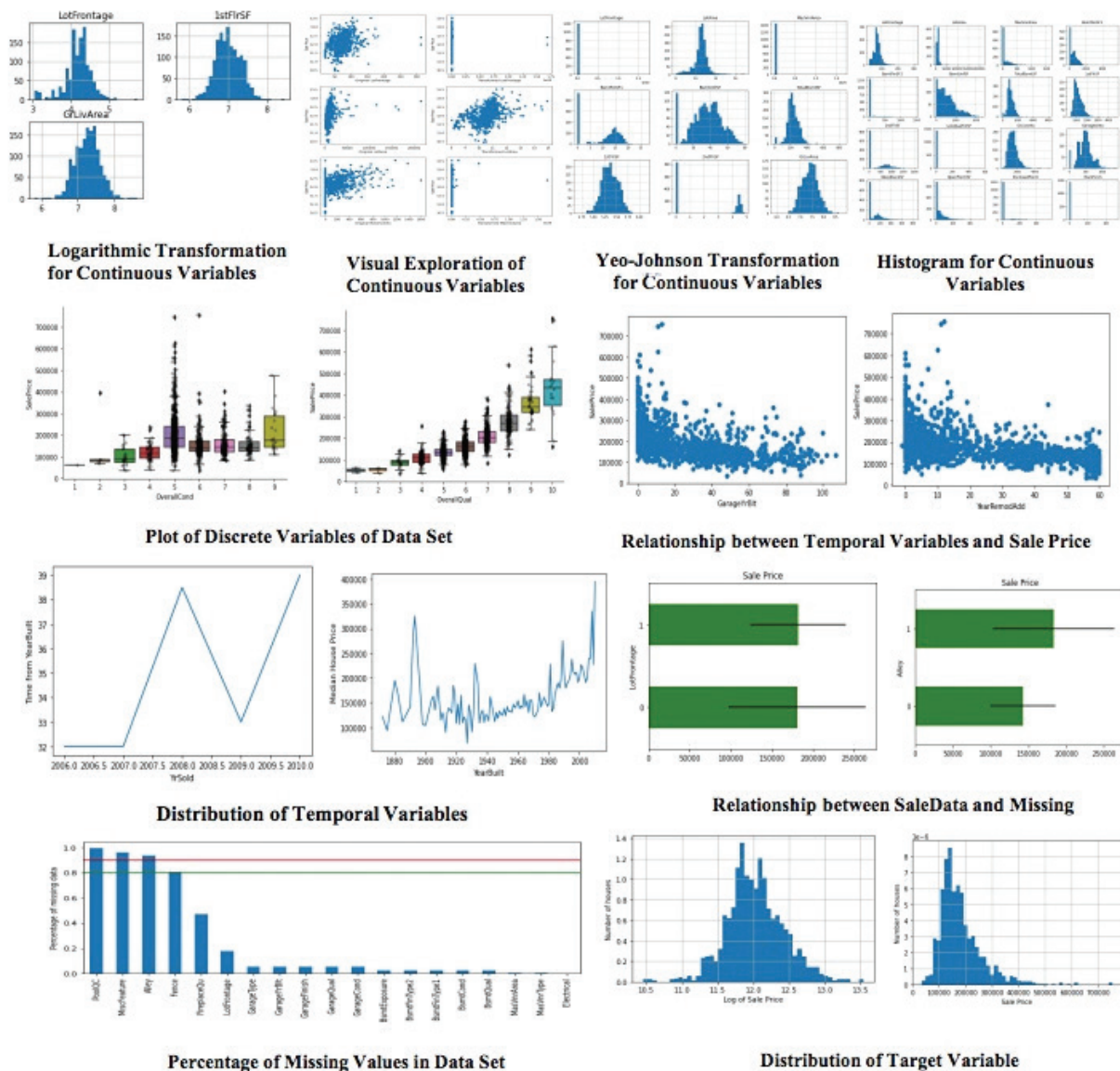


Fig. 9. Data analysis for different types of variables present in the dataset

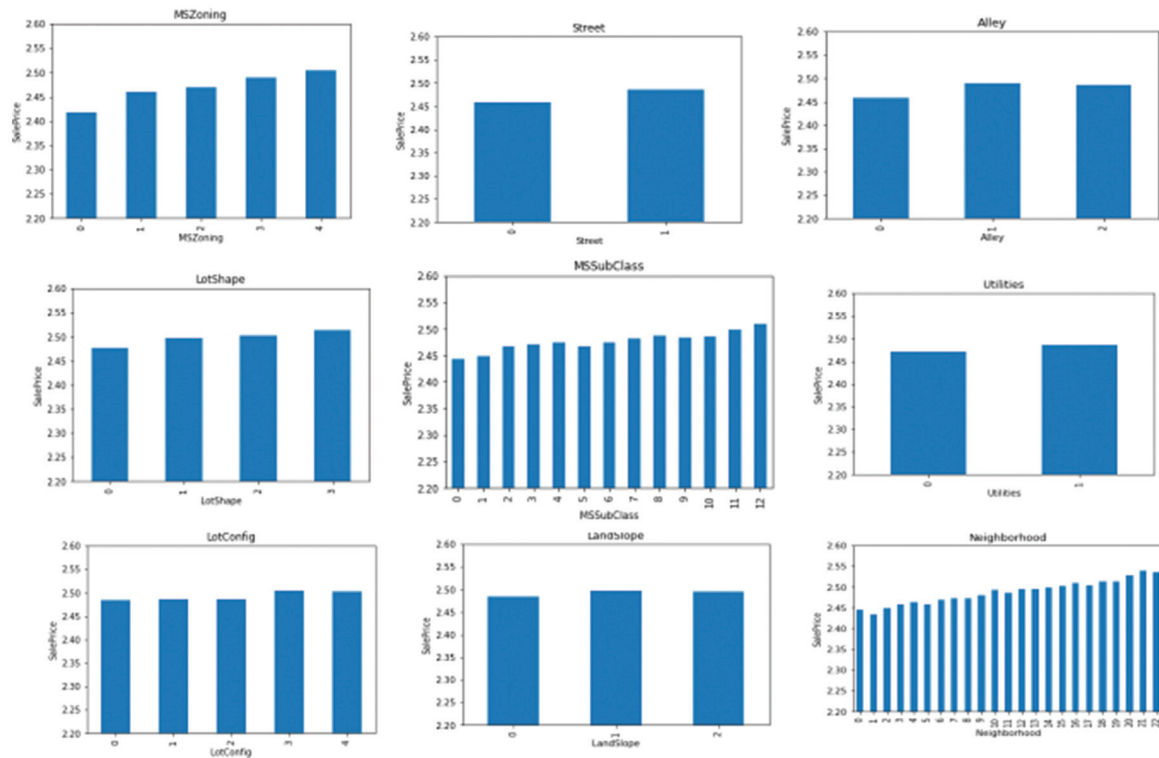


Fig. 10. Plot of monotonic variables versus target during feature engineering

```
In [6]: # let's visualise those features that were selected.
# (selected features marked with True)

sel_.get_support()

Out[6]: array([ True,  True,  True, False, False, False,  True,  True, False,
        True, False,  True, False, False, False, False,  True,  True,
        False,  True,  True, False,  True, False, False, False,  True,
        False,  True,  True, False,  True,  True, False, False, False,
        False, False, False,  True,  True, False,  True,  True, False,
        True,  True, False, False,  True, False, False,  True,  True,
        True,  True,  True, False, False,  True,  True,  True, False,
        False,  True,  True, False, False, False,  True, False, False,
        False, False, False, False, False,  True, False, False, False])

In [7]: # let's print the number of total and selected features

# this is how we can make a list of the selected features
selected_feats = X_train.columns[sel_.get_support()]

# let's print some stats
print('total features: {}'.format(X_train.shape[1]))
print('selected features: {}'.format(len(selected_feats)))
print('features with coefficients shrank to zero: {}'.format(
    np.sum(sel_.estimator_.coef_ == 0)))

total features: 81
selected features: 36
features with coefficients shrank to zero: 45

In [8]: # print the selected features
selected_feats

Out[8]: Index(['MSSubClass', 'MSZoning', 'LotFrontage', 'LotShape', 'LandContour',
        'LotConfig', 'Neighborhood', 'OverallQual', 'OverallCond',
        'YearRemodAdd', 'RoofStyle', 'Exterior1st', 'ExterQual', 'Foundation',
        'BsmtQual', 'BsmtExposure', 'BsmtFinType1', 'HeatingQC', 'CentralAir',
        '1stFlrSF', '2ndFlrSF', 'GrLivArea', 'BsmtFullBath', 'HalfBath',
        'KitchenQual', 'TotRmsAbvGrd', 'Functional', 'Fireplaces',
        'FireplaceQu', 'GarageFinish', 'GarageCars', 'GarageArea', 'PavedDrive',
        'WoodDeckSF', 'ScreenPorch', 'SaleCondition'],
        dtype='object')

In [9]: pd.Series(selected_feats).to_csv('selected_features.csv', index=False)
```

Fig. 11. Feature selection code implementation

```
In [7]: # evaluate the model:
# =====

# remember that we log transformed the output (SalePrice)
# in our feature engineering notebook (step 2).

# In order to get the true performance of the Lasso
# we need to transform both the target and the predictions
# back to the original house prices values.

# We will evaluate performance using the mean squared error and
# the root of the mean squared error and r2

# make predictions for train set
pred = lin_model.predict(X_train)

# determine mse, rmse and r2
print('train mse: {}'.format(int(
    mean_squared_error(np.exp(y_train), np.exp(pred)))))
print('train rmse: {}'.format(int(
    mean_squared_error(np.exp(y_train), np.exp(pred), squared=False))))
print('train r2: {}'.format(
    r2_score(np.exp(y_train), np.exp(pred))))
print()

# make predictions for test set
pred = lin_model.predict(X_test)

# determine mse, rmse and r2
print('test mse: {}'.format(int(
    mean_squared_error(np.exp(y_test), np.exp(pred)))))
print('test rmse: {}'.format(int(
    mean_squared_error(np.exp(y_test), np.exp(pred), squared=False))))
print('test r2: {}'.format(
    r2_score(np.exp(y_test), np.exp(pred))))
print()

print('Average house price: ', int(np.exp(y_train).median()))

train mse: 781396538
train rmse: 27953
train r2: 0.8748530463468015

test mse: 1060767982
test rmse: 32569
test r2: 0.8456417073258413

Average house price: 163000
```

Fig. 12. LASSO model implementation code

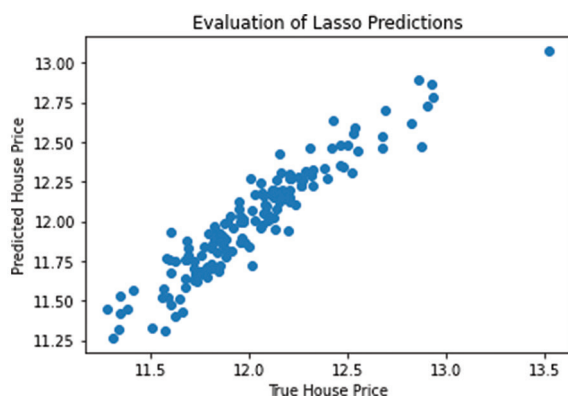


Fig. 13. LASSO model evaluations for the prediction of house-sale price

data and remove the clutter values. All these different types of variables are plotted in graphical notation to check their evidence and are shown in Fig. 9 before transformation.

Fig. 9 shows the graphical representations for different types of variables in the house-sale price prediction dataset.

5.2. Feature Engineering of the Dataset

After the data have been cleaned and making it free from outliers, feature engineering and exploratory data analysis have to be done. Fig. 10 shows the histograms for cleaned data after feature engineering.

```

1 FROM python:3.11
2
3 # Create the user that will run the app
4 RUN adduser --disabled-password --gecos "" ai-api-user
5
6 WORKDIR /opt/house-prices-api
7
8 ARG PIP_EXTRA_INDEX_URL
9
10 # Install requirements, including from Geofury
11 ADD ./house-prices-api /opt/house-prices-api/
12 RUN pip install --upgrade pip
13 RUN pip install -r /opt/house-prices-api/requirements.txt
14
15 RUN chown -R ai-api-user:ai-api-user /opt/house-prices-api/run.sh
16 RUN chown -R ai-api-user:ai-api-user ./
17
18 USER ai-api-user
19
20 EXPOSE 8081
21
22 CMD ["bash", "-c", "run.sh"]

```

```

1 jupyter_notebooks*
2 */env*
3 */venv*
4 .circleci*
5 packages/regression_model
6 *.env
7 *.log
8 .git
9 .gitignore
10 .tox

```

```

1 from typing import Any
2
3 from fastapi import APIRouter, FastAPI, Request
4 from fastapi.middleware.cors import CORSMiddleware
5 from fastapi.responses import HTMLResponse
6 from loguru import logger
7
8 from app.api import api_router
9 from app.config import settings, setup_app_logging
10
11 # setup logging as early as possible
12 setup_app_logging(config=settings)
13
14
15 app = FastAPI(
16     title=settings.PROJECT_NAME, openapi_url=f"{settings.API_V1_STR}/openapi.json"
17 )
18
19 root_router = APIRouter()
20
21
22 @root_router.get("/")
23 def index(request: Request) -> Any:
24     """Basic HTML response"""
25     body = (
26         "<html>"
27         "<body style='padding: 10px;'>"
28         "<h1>Welcome to the API</h1>"
29         "<div>"
30         "Check the docs: <a href='/docs'>here</a>"
31         "</div>"
32         "</body>"
33         "</html>"

```

Fig. 14. Python code for deploying machine learning model to production

5.3. Feature Selection after Feature Engineering

Because having irrelevant features in our data can reduce the model's accuracy, we use feature selection to automatically or manually select the features that contribute the most to the prediction variable. To select features for the final model, filter methods, such as constant feature elimination, quasi-constant feature elimination, duplicate feature elimination, fisher score, univariate method, mutual information, and correlation, are used. Wrapper methods such as step-forward selection, step-backward selection, and exhaustive search are examples of such methods. Methods such as decision tree-derived importance and regression coefficients are embedded.

5.4. Fitting the ML Model

The data are then divided into training and testing sets to classify the best-fitting ML model. The standard 80-20 split ratio is used: 80% of the data is considered a training set, and 20% is considered a testing set. Scikit-Learn must be imported before the model can be implemented. It is a Python library that provides ML algorithms for implementation as well

as many other modeling features. We are using the supervised regularization method, LASSO, to perform precise price estimation. Our model will be the one with the lowest error and the closest value prediction. After setting the seed for the model, the next step comes to implement the model to predict different accuracy measures, such as MSE, RMSE, and R2 and to give good predictions for our house-sale price dataset. Fig. 11 shows the code that we used for feature selection.

Fig. 13 shows the best-fit curve under the LASSO model for the price prediction dataset.

Based on the observations above, it is clear that linear regression produces the most precise results and has been chosen as the predictive model for house-sale prediction. The model is ready for use as an analytic tool for both real estate business managers and buyers.

5.5. Deployment of the Model

Once implemented, the model predicts the price of the property (house) in that specific location, Kabul, as selected in our dataset. Next comes deploying the model with the Docker container framework, in which

the user can enter the desired values and our model predicts the output. This is made possible using the Python package Heroku and Docker to create an API. To build the web application and connect the Model to it, we must first extract our model into pickle and json files and design a web page using HTML, CSS, and JavaScript. The model is now ready to be displayed and predicted on the web application.

6. Conclusion and Future Scope

The LASSO-supervised regularization ML model has proven to be an effective method for determining the best-fitting algorithm for a model. This linear regression algorithm provides a very accurate estimation of house prices. It provides much more accurate estimations for various locations. Furthermore, linear regression provides nearly accurate predictions based on the confusion matrix. Linear regression fits our dataset and performs well. Deployment of the model after implementation helps to predict the value automatically for different datasets. In the future, an appealing and interactive graphical user interface can be created to integrate into any real estate sale website, where sellers can provide details and houses for sale and buyers can contact based on the information provided on the website. To make things easier for the user, there could be a recommending system that recommends real estate properties based on the predicted price. The current dataset only includes a few locations in Kabul. Expanding it to other Indian cities and states is the long-term goal. Google Maps can be included to make the system even more informative and user-friendly. This will display the neighborhood amenities, such as hospitals and schools within a 1 km radius of the given location. This can also be factored into predictions because the presence of such factors raises the value of real estate property.

References

- Baldominos, A., Blanco, I., Moreno, A.J., Iturrarte, R., Bernárdez, Ó., & Afonso, C. (2018). Identifying real estate opportunities using machine learning. *Applied Sciences*, 8(11), 2321. <https://doi.org/10.3390/app8112321>
- Bass, L., Weber, I., & Zhu, L. (2015). *DevOps: A Software Architect's Perspective*. Addison-Wesley Professional, Boston.
- Baylor, D., Breck, E., Cheng, H.T., Fiedel, N., Foo, C.Y., Haque, Z., Haykal, S., Ispir, M., Jain, V., Koc, L., & Koo, C.Y. (2017). Tfx: A Tensorflow-Based Production-Scale Machine Learning Platform. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1387–1395.
- Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. *Mobile Networks and Applications*, 19, 171–209.
- Cheung, K.S., Yiu, C.Y., & Xiong, C. (2021). Housing market in the time of pandemic: A price gradient analysis from the COVID-19 epicentre in China. *Journal of Risk and Financial Management*, 14(3), 108.
- Duvall, P.M., Matyas, S., & Glover, A. (2007). *Continuous Integration: Improving Software Quality and Reducing Risk*. United Kingdom: Pearson Education.
- Fan, C., Cui, Z., & Zhong, X. (2018). House Prices Prediction with Machine Learning Algorithms. In: *Proceedings of the 2018 10th International Conference on Machine Learning and Computing, February 26–28, 2018, Macau, China*, pp. 6–10.
- Google Cloud. (2020). *MLOps: Continuous Delivery and Automation Pipelines in Machine Learning*. Available from: <https://cloud.google.com/solutions/machine-learning/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning>
- Humble, J., & Farley, D. (2010). *Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation*. Pearson Education, United Kingdom.
- Hummer, W., Muthusamy, V., Rausch, T., Dube, P., El Maghraoui, K., Murthi, A., & Oum, P. (2019). Modelops: Cloud-Based Lifecycle Management for Reliable and Trusted Ai. In: *2019 IEEE International Conference on Cloud Engineering (IC2E), June 24–27, 2016, Prague, Czech Republic*, pp. 113–120.
- John, M.M., Olsson, H.H., & Bosch, J. (2021). Towards MLOps: A Framework and Maturity Model. In: *Proceedings of the 2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), September 1–3, 2021, Palermo, Italy*, pp. 334–341.
- Jui, J.J., Imran Molla, M.M., Bari, B.S., Rashid, M., & Hasan, M.J. (2020). Flat Price Prediction Using Linear and Random Forest Regression Based on Machine Learning Techniques. *Embracing Industry 4.0. Embracing Industry 4.0: Selected Articles from MUCET 2019. Vol. 678*. Springer, Singapore, pp. 205–217.
- Kang, J., Lee, H.J., Jeong, S.H., Lee, H.S., & Oh, K.J. (2020). Developing a forecasting model for real estate auction prices using artificial intelligence. *Sustainability*, 12(7), 2899. <https://doi.org/10.3390/su12072899>

- Kaynak, S., Ekinci, A., & Kaya, H.F. (2021). The effect of COVID-19 pandemic on residential real estate prices: Turkish case. *Quantitative Finance and Economics*, 5, 623–639.
<https://doi.org/10.3934/QFE.2021028>
- Kumeno, F. (2019). Software engineering challenges for machine learning applications: A literature review. *Intelligent Decision Technologies*, 13(4), 463–476.
<https://doi.org/10.3233/IDT-190160>
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.
<https://doi.org/10.1038/nature14539>
- Makinen, S., Skogstrom, H., Laaksonen, E., & Mikkonen, T. (2021). Who needs MLOps: What Data Scientists Seek to Accomplish and how can MLOps Help? In: *Proceedings of the 2021 IEEE/ACM 1st Workshop on AI Engineering—Software Engineering for AI (WAIN), May 30–31, 2021, Madrid, Spain*.
- Marrero, L., & Astudillo, H. (2021). *DevOps-RAF: An Assessment Framework to Measure DevOps Readiness in Software Organizations. Proceedings of the 2021 40th International Conference of the Chilean Computer Science Society (SCCC), November 15–19, 2021, La Serena, Chile*, pp. 1–8.
- Matsui, B., & Goya, D. (2020). *Application of DevOps in the Improvement of Machine Learning Processes. Conference Paper*.
- Mora-Garcia, R.T., Cespedes-Lopez, M.F., & Perez-Sanchez, V.R. (2022). Housing price prediction using machine learning algorithms in COVID-19 times. *Land*, 11(11), 2100.
<https://doi.org/10.3390/land11112100>
- Neloy, A.A., Haque, H.S., & Ul Islam, M.M. (2019). Ensemble Learning Based Rental Apartment Price Prediction Model by Categorical Features Factoring. In: *Proceedings of the 2019 11th International Conference on Machine Learning and Computing, February 22–24, 2019, Zhuhai, China*, pp. 350–356.
- Olorisade, B.K., Brereton, P., & Andras, P. (2017). *Reproducibility in Machine Learning-based Studies: An Example of Text Mining*. Reproducibility in Machine Learning, Australia.
- Oxenstierna, J. (2017). *Predicting House Prices using Ensemble Learning with Cluster Aggregations*. Bachelor thesis, Uppsala University, Sweden.
- Pai, P.F., & Wang, W.C. (2020). Using Machine Learning Models and Actual Transaction Data for Predicting Real Estate Prices. *Applied Sciences*, 10, 5832.
<https://doi.org/10.3390/app10175832>
- Ruf, P., Madan, M., Reich, C., & Ould-Abdeslam, D. (2021). Demystifying mlops and presenting a recipe for the selection of open-source tools. *Applied Sciences*, 11(19), 8861.
<https://doi.org/10.3390/app11198861>
- Rzig, D.E., Hassan, F., & Kessentini, M. (2022). An empirical study on ML DevOps adoption trends, efforts, and benefits analysis. *Information and Software Technology*, 152, 107037.
<https://doi.org/10.1016/j.infsof.2022.107037>
- Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.F., & Dennison, D. (2015). Hidden technical debt in machine learning systems. *Advances in Neural Information Processing Systems*, 28, 2503–2511.
- Subramanya, R., Sierla, S., & Vyatkin, V. (2022). From DevOps to MLOps: Overview and application to electricity market forecasting. *Applied Sciences*, 12(19), 9851.
<https://doi.org/10.3390/app12199851>
- Wang, D., & Li, V.J. (2019). Mass appraisal models of real estate in the 21st century: A systematic literature review. *Sustainability*, 11, 7006.
<https://doi.org/10.3390/su11247006>
- Wikipedia. (2020). *Online Machine Learning*. Available from: https://en.wikipedia.org/wiki/online_machine_learning

Measuring the accuracy of time series reduction methods based on modified dynamic time warping distance calculations

Anupama Jawale^{1*}, Amiya Kumar Tripathy²

¹Department of Information Technology, Narsee Monjee College of Commerce and Economics, Mumbai, Maharashtra, India

²Department of Computer Engineering, Don Bosco Institute of Technology, Mumbai, Maharashtra, India

*Corresponding author E-mail: anupama.jawale26@gmail.com, anupama.jawale@nmcce.ac.in, amiya@dbit.in

(Received 11 September 2024; Final version received 22 September 2024; Accepted 25 November 2024)

Abstract

Representation of sensor data in the form of time series is a crucial aspect of numerous related tasks such as comparison, reduction, clustering, and classification. Time series representation methods included in most programming languages/integrated development environments support dimensionality reduction, data preprocessing, and feature extraction for time series data, as do several normalization techniques. This research study focused on 14 different methods of dimensionality reduction from the TSepr (R Studio) package on eight different time series, which are collections of sensor data of varying lengths. The similarity of the reduced time series and the original time series is compared using a modified version of dynamic time warping with time alignment measurement. These methods are further combined with the Gaussian kernel function to normalize the distance between variously aligned series. The results showed that perceptually important points (PIP) and piecewise linear approximation (PLA) were found as the best methods for TS reduction with a minimum deviation (error term) as low as 5 – 12%. The results also indicate that PIP performs significantly differently compared to seasonal decomposition, while there are no significant differences between PIP and the other methods (PLA, FEACLIPTREND, and FEACLIP). In addition, this research study demonstrated the development of an interactive web-based application in which time series are stored in csv files, and the distance between them is calculated through the chosen reduction method.

Keywords: Dimensionality, Distance, Dynamic Time Warping, Gaussian Kernel, Time Series

1. Introduction

Many conventional waveform processing techniques can be used for a time series since it can be seen as a waveform when graphically displayed. Accelerometer data consists of three channels (x, y, and z) and is collected at a high sampling rate. This leads to a large amount of data being collected with close, continuous values against increasing unit time. For example, data collected at 100 Hz sampling data generates 6000 data points per minute per channel. Handling such large data often requires larger computational costs and storage, which makes the task challenging to process in real time. Raw accelerometer signal contains noise caused due to physical sensor inaccuracies and external vibrations. By reducing the dimensionality of this data, noise reduction and

smoothing of the data make it less sensitive to noise. In this study, accelerometer data collected for road abruptions is driven in the form of time series, and dimensionality reduction techniques are presented, using 14 different methods of dimensionality reduction.

Processing accelerometer data, which consists of three channels (x, y, z) collected at high sampling rates, presents significant challenges due to the sheer volume of data generated – 6000 data points per minute per channel at 100 Hz sampling frequency. This large dataset can lead to increased computational costs and storage requirements, complicating real-time processing efforts (Hussein et al., 2024). The raw signals are often contaminated with noise from sensor inaccuracies and external vibrations, necessitating effective noise reduction techniques to enhance data quality. To address these challenges,

dimensionality reduction techniques are employed, which help in reducing the data's complexity while preserving essential information (Juliusdottir, 2023). In this study, 14 distinct methods of dimensionality reduction are explored, facilitating the smoothening of data and making it less sensitive to noise. In addition, a symbolic approach is introduced to represent the data streams in a reduced space, transforming real-valued data into a string of symbols, which aids in the efficient processing of time series data. By integrating these methodologies, the study aims to improve the handling of accelerometer data collected during road abruptions, ultimately enhancing real-time analysis capabilities (Juliusdottir, 2023).

A Time series, if represented graphically, can be viewed as a waveform and hence supports many traditional methods of waveform processing. Time series are collections of data points recorded against timestamps. Sensor data are prototype examples of time series. The series under study consists of accelerometer data generated using a smartphone sensor. In general, a time series, in its simplest form, can be defined as follows in Eq. (1)

$$TS = [(dpt_1, t_1), (dpt_2, t_2), (dpt_3, t_3) \dots (dpt_n, t_n)] \quad (1)$$

Where each dpt is a data point at t is a time at which dpt is measured.

Classification of time series representations has been performed by several researchers (Biemann & Masegla, n.d.) as shown in the diagram below (Fig. 1). Non-data adaptive time representation refers to the approximation of a time series based on the local properties of the dataset. The data-adaptive representation chooses a common representation from the original time series such that while reconstructing the original time series from the reduced one, the global error is minimized. Model-based representations use a statistical model to represent the characteristics of time series (Wang et al., 2010).

The selection of the 14 dimensionality reduction methods is based on their ability to effectively manage high-dimensional time series data while preserving essential features. The implementation of distance functions, particularly the combination of dynamic time warping (DTW) and time alignment measurement (TAM), enhances the assessment of similarity between time series. Furthermore, addressing the statistical significance of results with a heat map ensures the reliability and applicability of findings, paving the way for improved analysis and interpretation of time series data across various domains. The methods used to reduce the time series considered in this research study are described in the following section.

1.1. Non-data Adaptive Methods

- a. Piecewise aggregate approximation (PAA): The default algorithm of PAA uses the mean as the aggregation function. This method uses mean, max, min, sum, or any other aggregate function passed by the user. The PAA approximation is given by Eq. (2) below (Ines Silva & Henriques, 2020).

$$\bar{x}_i = \frac{M}{n} \sum_{i=1}^{n/M} x_i \quad (2)$$

- b. Discrete wavelet transform (DWT): This function computes discrete wavelet coefficients from a given time series. The parameter *level* determines the number of coefficients, whereas the filter option provides types of wavelet filters (for example, haar, d6, and d2). The DWT divides signals into *details* and *approximate* parts. The transform contains an insignificant noise component (*details*) that can be removed or filtered out using two basic filters, thresholds, and/or wavelet types. The *Filter* parameter defines the basic waveform matching with the shape of the original waveform to be filtered out. DWT is given as follows in Eq. (3)

$$\varphi(x) = \sum_{k=-\infty}^{\infty} (-1)^k a_{N-1-k} \varphi(2x-k) \quad (3)$$

- c. Discrete Fourier Transform (DFT): DFT is the primary transformation function used in digital signal processing. According to the mathematical formula, the discrete Fourier transform converts N discrete-time samples to the same number of discrete frequency samples as given by equation (4)

$$f(x) = a_0 + \sum_{n=1}^{\infty} \left(a_n \cos \frac{n\pi x}{L} + b_n \sin \frac{n\pi x}{L} \right) \quad (4)$$

- d. Discrete Cosine Transform (DCT): DCT is the technique for converting a signal into elementary frequency components. DCT represents the input signal as a linear combination of weighted basis functions related to the frequency component. DCT can be mathematically represented as given below in Eq. (5):

$$F(u) = \left(\frac{2}{N} \right)^{\frac{1}{2}} \sum_{i=0}^{N-1} \tilde{E}(i) \cdot \cos \left[\frac{\pi \cdot u}{2 \cdot N} (2i+1) \right] f(i) \quad (5)$$

- e. Simple moving average (SMA): A SMA is a statistical method that calculates the mean of subsets of the dataset. The function returns a time series of length: length=length (TS)-order+1,

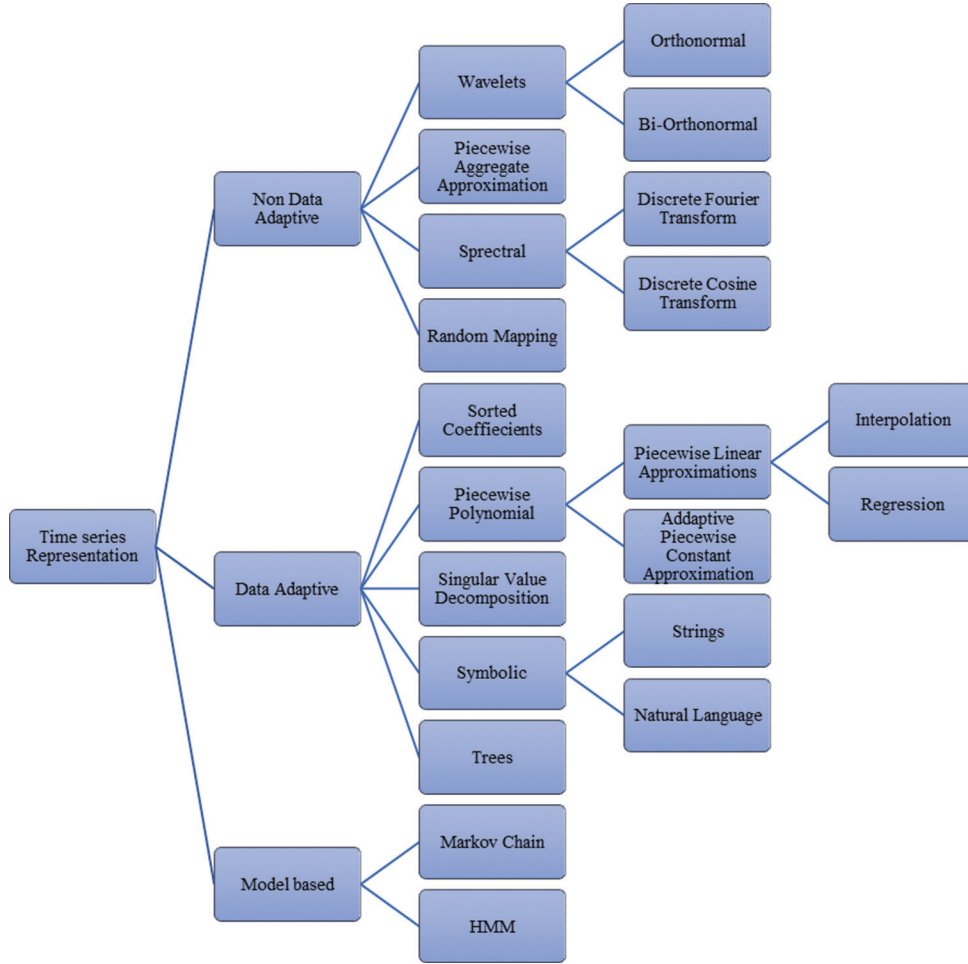


Fig. 1. Classification hierarchy of time series representation

where the order is the parameter to the function, given by Eq. (6)

$$SMA = \frac{1}{M} \sum_{i=1}^M dtp_i \quad (6)$$

- f. Perceptually important points (PIP): PIPs are identified by extracting dominating data points from the shape of the time series (Jiménez et al., 2016). The function accepts the number of PIPs to be identified and returns them with or without the time stamp value as specified by the user.

1.2. Data Adaptive Methods

1. Symbolic aggregation approximation (SAX): This method was first proposed by Lin et al. (2000) and extends the concept of piecewise approximation. SAX is a symbolic representation of univariate time series, allowing dimensionality reduction with low storage requirements. It is applicable in motif discovery, data mining, and large-scale data processing (Camerra et al., 2010). SAX converts a time series TS of

length n into a string of arbitrary length where $len(string) \ll n$. To construct the alphabet, SAX uses the formula given in Eq. (7).

$$\mathbb{C}^*i = \alpha * j, \text{ if } \bar{c}^*i \in (\beta_{j-1}, \beta_j) \quad (7)$$

Then, SAX locates the distance calculation in the lookup table of the $N \times N$ matrix to construct the alphabet.

2. Piecewise linear approximation (PLA): The PLA is a method of fitting a non-linear objective function to an approximation function by adding additional variables and constraints (Lin et al., 2003). The function converts TS to a specified number of points using the PLA algorithm. The overall piecewise linear function is given by Eq. (8).

Let $x_0, x_1, x_2, \dots, x_n$ where n is the number of subintervals.

Hence, each subinterval is defined as a linear function.

$$f_i(x) = m_i(x - x_i) + b_i \quad (8)$$

1.3. Model-Based Methods

1. Mean Seasonal Profile: This method computes the mean seasonal profile of the time series. The length of the representation can be specified by the *freq* parameter.
2. Model-based seasonal representations based on linear additive models: linear models or generalized additive models combine the properties of generalized linear models and additive models. These methods extract linear coefficients from a given time series depending upon the frequency assigned by the user. In the GAM model, Y variable depends linearly on the unknown smoothing function of certain variables. GAM is given by the following formula: Eq. (9),

$$f(\vec{x}) = \Phi \left(\sum_{p=1}^n \phi_p(x_p) \right) \quad (9)$$

3. Exponential smoothing seasonal coefficients: This function extracts exponential smoothing seasonal coefficients from the time series. This method is suitable for data that do not show any seasonal pattern or trend. Eq. (10) represents the mathematical formulation of exponential smoothing.

$$S_t = \alpha \cdot X_t + (1-\alpha) \cdot S_{t-1} \quad (10)$$

1.4. Data Dictated Methods

1. Feature extraction from clipped representation: This method computes features of the time series using bit-level clipped representation. It extracts 8-bit features from the data. This approach is a sustainable high-performance outlier detection method (<http://Acmbulletin.Fiit.Stuba.Sk/Vol10num2/Vol10num2.Pdf>, n.d.).
2. Feature extraction from the trending representation: similar to the clipped representation of a time series, this function extracts bit-level features but with trending. The user specifies the number of pieces and forms every piece; two features are extracted.
3. Feature extraction from clipped and trending representations: In this method, clipping and Trending both bit-level representations are combined for time series feature extraction.

To standardize the raw data series collected, normalization of time series, followed by windowing and clipping, are implicated in the proposed methodology. The normalization method of min-max normalization is used in this research study and is given in Eq. (11):

$$Y_i = \frac{X_i - \min(X)}{\max(X) - \min(X)} \quad (11)$$

2. Related Work

Any collection of a series of data points that is observed for different points of time is a time series. The representation of sensor data as time series is essential for various tasks, including dimensionality reduction and classification. Recent studies have explored multiple methods for effectively reducing the dimensionality of time series data, highlighting their performance and applicability. Table 1 highlights some recent work in this domain.

In time series, data correlation within adjacent points in time makes time series analysis a special field of interest with special statistical features. This time correlation forms many mathematical and statistical questions with various applications in diverse fields. For example, data points collected by earthquake sensors, data points collected by temperature sensors, or brain waves in electroencephalogram. Time series patterns pose certain diverse applications of time series. In this section, we will concentrate on one special type of time series that is generated by accelerometer data signals. The accelerometer, as the data recording instrument, records vertical vibration data at a frequency of 50 Hz. The general problem of interest is to classify or distinguish different types of waveforms generated by this accelerometer in case of different events observed. There are many features generated from this time series, for example, amplitude ratios, threshold of vibration, maximum amplitude, and so on. Along with these time domain features, there are several frequency domain features like spectral analysis of variance, septal coefficients, and linear prediction coefficients. In time series analysis, shape analysis is another area of interest. The shape appearance of the time series changes completely with varying sampling rates or with varying frequency and different numbers of frames of sample. Time series can differ in degrees of smoothness (Bairagi, 2018).

Moving averages and auto regression are some methods used to represent time series and used to predict time series future values. However, for recording events data with accelerometer, these techniques are not very useful as they are based on previous values in the time series (Wang et al., 2010). Autocorrelation and cross-correlation are certain methods that can be used for the comparison of similarity between two-time series. As described in the later section, electronic signal comparison techniques like DTW, SAX distances, Correlation distances, and Fourier distances are some distance measures to find out the difference between two or more time series (De Oliveira Marques et al., 2022).

Although the R programming language is a popular tool for statistical research, there are few related research papers dedicated to a specific package or functionality in R. The TSrepr basis functions are also

Table 1. Sensor data as time series and applications

Research study	Insights
Hussein et al., 2024	The authors propose a novel approach using early exit classifiers that can make accurate inferences with partial sensor data, significantly reducing energy usage while maintaining accuracy. Evaluations across six datasets demonstrate that the proposed method can achieve energy savings of 50 – 60% without compromising classification accuracy.
Meng et al., 2024	A dimension reduction method to reduce scale for time series and analyze the correlation between multi-source sensors is proposed and carried out on an industrial excavator dataset to verify the effectiveness and preponderance of the method.
He et al., 2023	In this paper, a double mean representation method, symbolic aggregate approximation based on double mean representation (SAX-DM), was proposed for time series data.
Wang et al., 2023	Wang <i>et al.</i> , as discussed by the authors, proposed a multivariate time-series unsupervised domain adaptation (MTS-UDA) method to reduce the domain discrepancy at both the local and global sensor levels.
Ashraf et al., 2023	In this article, the authors present twelve different dimensionality reduction algorithms that are specifically suited for working with time-series data and fall into different categories, such as supervision, linearity, time and memory complexity, hyper-parameters, and drawbacks.

available for C++ language integration (Eddelbuettel & François, 2011) and hence open up many wide areas of research in time series. However, modern time series data with minor time intervals, such as accelerometer or sensor data, need to be studied further. Several of the time series considered by various researchers include electricity consumption and sales forecasting. Time series analysis in R programming with autoregressive integrated moving average (ARIMA) model has been employed to forecast electricity usage. Linear regression and ARIMA are used for mining time series for the women's expenditure dataset (Tanwar & Kakkar, 2017). The authors stated that the prediction accuracy is similar for both prediction models. For high-voltage load forecasting, Matsila and Bokoro (2018) used R-visualization techniques for time series and linear progression. In the study by Wang et al. (2010), all

the methods are compared on the basis of similarity measures and step patterns for parameter tuning. Ali et al. (2019) review numerous methods for the visual analysis of time series data, such as clustering, classification, and other distance matrix computation methods. For a larger time series, (Camerria et al., 2010) paper defines a novel data structure called iSAX for the SAX method of aggregate approximation, which is described by the TSrepr package. The repository (Laurinec, 2018) provides the updated open-source code and documentation for the TSrepr package. The classification accuracy of all methods of the TSrepr package with aggregation and clustering methods was assessed in a previous study (Laurinec & Lucka, 2016) based on robust linear regression, exponential smoothing, and other adaptive and model methods from the package. The authors have implemented these methods for forecasting electricity consumption. Another major area of study for time series mining is in the area of motif discovery. The package TSMining (Lin et al., 2003) also implements various functions of TSrepr, but the goal is toward motif discovery from time series mining rather than dimensionality reduction of time series data.

For the multivariate time series, the alignment and similarity assessment (MTASA) framework discussed in a study by Tonle et al. (2024) integrates multiple steps of time series similarity assessment, including feature representation, alignment, and similarity measurement. With digital signal processing techniques, such as cross-correlation and convolution, MTASA enhances the alignment of time series data, addressing challenges related to noise and temporal misalignments. The implementation of a multiprocessing engine further optimizes computational resources, making the framework suitable for large-scale datasets. This method shows promise in applications such as environmental monitoring and agricultural studies, where multivariate data is prevalent. For the similarity search methods, another research study (He et al., 2023) indicates that similarity measures should not only focus on direct comparisons but also consider the underlying structures and patterns within the data. This adaptability is crucial in fields such as engineering, where degradation curves of similar systems need to be compared accurately for predictive maintenance.

On the basis of the evidence from these previous studies, we observed unexplored research on time series representation, suggesting the need for additional studies in the domain of time series reduction and distance calculation methods. Section IV of this paper explores multiple methods of distance calculation, whereas Section 5 presents the methodology of the work. The next section describes the data set used in this research study.

3. Dataset Description

The dataset used in this research study was collected by an accelerometer sensor mounted on a two-wheeler vehicle that travels on different types of roads, namely, a bituminous road, a concrete road, paved, and unpaved road (Fig. 2A-D, respectively). Smartphones' built-in triaxial accelerometer sensors are used to collect Z-axis readings, which are vertical acceleration readings against gravitational force. The roughness and surface texture of the road are reflected in accelerometer readings. The frequency of data collection was set to 50 data points per second.

A testbed was developed to aid data collection in the development and testing of this research. Fig. 3 shows the vertical placement of the smartphone with the sensors on a two-wheeler. The data collected were in the raw format, preprocessed for missing values, and normalized via the min-max normalization technique.



Fig. 2. (A-D) Types of roads



Fig. 3. Placement of smartphone

Previous studies have shown that the collection of accelerometer sensor data is not affected by the speed of the vehicle (Anand et al., 2020).

4. Distance Function

Viewing the statistical properties of time series as distance measures provides a powerful approach to understanding the behavior and relationships between time series data. Mahalanobis distance, measures of divergence, and higher-order statistics provide valuable insight into the central tendency, variability, shape, and time dependencies of time series. Careful consideration of data pre-processing, dimensionality, and complexity is essential for a meaningful application of statistical properties as distance measures. Overall, the inclusion of statistical properties in the time series analysis contributes to more accurate and robust analysis in different areas.

The distance functions used in this research study are DTW combined with TAM. Correlation-based and compression-based dissimilarity are a few more commonly used functions for time series data (Giorgino, 2009; Salvador & Chan, n.d.; Sharma et al., 2020; Singh & Meena, 2009). Eqs. 12 – 15 show the mathematical formulation of these distance functions (Montero & Vilar, 2014).

Dynamic Time Warping Distance

$$D(i, j) = |x_i - y_j| + \min \begin{cases} D(i-1, j-2) \\ D(i-1, j-1) \\ D(i-2, j-1) \end{cases} \quad (12)$$

Time Alignment Measurement Distance $\Gamma =$

$$\bar{\varphi} + \bar{\varphi} + (1 - \bar{\varphi}) \quad (13)$$

where $\bar{\varphi}$ = fraction of advance of signal, $\bar{\varphi}$ = delay and $\bar{\varphi}$ = phase

Correlatonbased Distance =

$$1 - \frac{\frac{1}{n+1} \sum_{i=1}^n ((x_i - \bar{x}) \cdot (y_i - \bar{y}))}{\sigma_X \sigma_Y} \quad (14)$$

Compressionbased Dissimilarity Distance =

$$\frac{\text{Compressed}(T_1 T_2)}{\text{Compressed}(T_1) \cdot \text{Compressed}(T_2)} \quad (15)$$

However, using the raw distance functions is not always guaranteed to produce the best results. The technique of collecting local neighborhood data by converting the distance to a Gaussian kernel and giving more weight to closer neighbors, can increase the distinctiveness of the similarity measure and increase the classification accuracy.

4.1. Understanding Kernel Methods

Machine learning for non-linear computing often employs a class of approaches known as kernel methods. They use the idea of feature mapping to add dimension to the original data, which could make linear processes more efficient (De Oliveira Marques et al., 2022). Kernel approaches prevent the direct calculation of the changed data points while enabling efficient calculations with a kernel function that automatically determines this feature mapping.

a. Kernel Functions:

The kernel function is the basic unit of kernel distance calculation. A kernel is a set of mathematical functions that accepts the input and converts it into the required type of output. For example, given two input vectors, a kernel function returns the inner product in the new feature space. The similarity metric between the changed data points and this inner product is identical. Typical kernel operations include:

1. Linear Kernel: The linear kernel, which is the dot product of the input vectors, represents the initial input space. The linear kernel function is defined as

$$k(x_i, x_j) = x_i \cdot x_j \quad (16)$$

2. Polynomial Kernel: By increasing the dot product to a certain level, the polynomial kernel enables the capture of polynomial correlations between data points.

$$k(x_i, x_j) = (x_i \cdot x_j)^e \quad (17)$$

3. Gaussian – Radial Basis Function Kernel: The Gaussian kernel uses the radial basis function to calculate similarity. It gives closer locations more similarity and decreases with distance.

$$k(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}} \quad (18)$$

4. Sigmoid Kernel: The sigmoid kernel models sigmoidal interactions between data points by capturing similarity based on the hyperbolic tangent function.

$$k(x_i, x_j) = \tanh(cx_i \cdot x_j + h) \quad (19)$$

b. Calculation of the Kernelized Distance:

We use the kernel trick concept to determine the kernel distance between two data series. We can directly compute the kernel function values between the input vectors instead of explicitly computing the feature vectors and computing distances in transformed space. Without explicitly computing the changed vectors, the kernel distance captures the disparity between data points in the changed feature space.

In this research study, to calculate kernel distance, we use the equation of Gaussian kernel as illustrated in Eq. 18.

5. Methodology

This research study focused on the accuracy of the dimensionality reduction techniques of the TSrepr package. 8 different length time series presenting Z – Acceleration of smartphone accelerometer data during different vehicle travels are considered for the analysis. These data are normalized to bring all the data points to the same scale and range of values. The various dimensionality reduction methods listed in Section 1 are implemented on this dataset. The resulting reduced dataset series is compared with the original series for similarity using the normalized distance of the DTW+TAM method. The distance between two series is given by the normalized cumulative distance. This method is highly adaptable and can be applied to a wide range of domains, including finance, healthcare, and environmental science.

The combination of DTW and TAM allows a comprehensive measure of distance calculation that measures both features of time series, temporal alignment, and magnitude difference. Eq. 20 shows the mathematical formulation of combining DTW and TAM.

$$Distance(x, y) = \alpha \cdot DTW(x, y) + (1 - \alpha) \cdot TAM(x, y) \quad (20)$$

α is a weight parameter that can be adjusted according to the weightage to be assigned to the respective factor. To ensure appropriate scaling of the distance metric, a normalization followed by Gaussian kernel-based distance calculation is applied Eq. (21).

$$k(x_i, x_j) = 1 - e^{-\frac{\omega^2}{2\sigma^2}} \quad (21)$$

ω represents the normalized distance of two-time series as given in Eq. 20; σ is a parameter that controls the width or scale of the Gaussian kernel.

The process of combining and normalizing data has the potential to enhance the accuracy of similarity assessments, thereby improving the performance of applications such as clustering, classification, and anomaly detection. Fig. 4 shows the sequence of steps of this methodology.

When the two series are the same, the normalized cumulative distance between them is zero. Any deviation from the value of zero is considered an error term. In Section 5 of this paper, we have presented the difference between the original series and the reduced series as a result of our experimental work.

In addition, the flexible integration of similarity measurements into various algorithms using the

Gaussian kernel improves their performance and enables more effective data analysis and decision-making. This research study attempts to solve this problem by following a two-step process, namely, calculating the distance between two-time series and applying a kernel function to map this distance onto a separable plane. Step 1 includes calculation of the DTW and TAM in normalized form, and Step 2 implements the Gaussian kernel function, as discussed in the above section.

Furthermore, this study presents an interactive, integrated application to improve the usability of distance calculations for exploring and analyzing time series data. Users can input time series datasets, preprocess the data, carry out reduction activities, and interactively visualize the outcomes using the program. The application will make it simple for users to extract useful insights from their time series data by offering an intuitive user interface. The algorithm for

creating the application is given below.

6. Results and Discussion

The combination of alignment cost (DTW) and magnitude difference (TAM) provides a comprehensive measure of similarity, taking into account both temporal alignment and differences in values. The measure can be adjusted according to specific needs using a weighted average of distances. The normalization of the Gaussian kernel ensures that the distance metric is appropriately scaled, facilitating its interpretation and comparison across diverse datasets.

The Gaussian Kernel makes DTW distance in the form of a positive semi-definite (PSD) matrix, a symmetric matrix with non-negative eigenvalues. The PSD matrix is defined as

$$M \in L(V), \text{ where } M \text{ is symmetric and } v^T M v > 0 \quad \forall v \in V \quad (22)$$

The PSD matrix ensures that the SVM algorithm will terminate at a global optimum, which leads to a more interpretable and reliable solution.

In addition, it improves the detection of similarities by identifying similarities that may go unnoticed when relying solely on a single measure.

Table 2 shows the tabulated results of all the normalized cumulative distances between the reduced series and the original series. This is the error term given by $|0\text{-NormDist}|$. Table 3 shows the five methods with the minimum error term calculated using the formula given in Eq. (23).

$$\text{Percentage Error Term} = |0\text{-NormDist}| * 100 \quad (23)$$

The PIP, PLA, seasonal decomposition (SEAS), feature extraction and clipping for trend (FEACLIPTREND), and feature extraction and clipping (FEACLIP) methods are advanced strategies for dimensionality reduction in time series analysis. Let $X \in R^{n \times m}$ represents a time series with n observations and m features. The goal of dimensionality reduction is to transform X into a lower dimension representation $Y \in R^{n \times k}$ where $k < m$. The understudied methods aim to retain certain statistical features of a time series data with reduced dimensionality, as described below.

Fig. 5 shows the results of the heatmap visualization. Fig. 6 shows a visualization of the original time series and reduced time series. The PIP, PLA, SEAS, FEACLIPTREND, and FEACLIP methods yield the best results by considerably reducing the dimensionality but keeping the original features intact since the distance between the original and reduced time series is significantly

Algorithm: Application

Import the necessary libraries: shiny and TSdist.

the UI:

Create a fluid page.

Define the server

Extract the data from the uploaded CSV files.

Convert the data into numeric vectors.

Define Time series Reduction Methods

reduced_ts <- reduced_timeseries (methodname (), original_ts)

Diff <- DTW+TAM (original_ts, reduced_ts)

Diff -> 0 indicates effective reduction without loss

Redirect output to server

Algorithm: Distance Calculation

Define the Gaussian kernel function.

Input: x (input value)

Output: Gaussian kernel value using the input value and a fixed sigma value

$\delta = \text{DynamicTimeWarping Distance } D(i, j) =$

$$|x_i - y_j| + \min \begin{cases} D(i-1, j-2) \\ D(i-1, j-1) \\ D(i-2, j-1) \end{cases}$$

Time Alignment Measurement Distance $\Gamma = \bar{\phi} + \bar{\phi} + (1 - \bar{\phi})$

where $\bar{\phi}$ = fraction of advance of signal, $\bar{\phi}$ =

delay and $\bar{\phi}$ = phase

combined distance $\mathcal{C} = \alpha \cdot \delta + (1 - \alpha) \cdot \Gamma$

Normalized distance $\omega = \frac{\mathcal{C}}{\text{Max}_{\delta}}$

Difference $\Delta = 1 - e^{-\frac{\omega^2}{2\sigma^2}}$

Output: return (Δ)

PIP (Principal Information Preservation)	$Y=XW$, where W is projection matrix such that $\text{maximizeVar}(Y), W_F^2=1$
PLA (Piecewise Linear Approximation)	$Y_i = \sum_{j=1}^k a_j \cdot 1_{t_j, t_{j+1}}(t_i)$ where a_j is coefficient of linear segment, 1 is an indicator function to check if t_i falls within a range
SEAS (Seasonal Decomposition)	$X(t) = T(t) + S(t) + R(t)$ where T: Trend, S: Seasonality and R: Residual
FEACLIPTREND (Feature Extraction and Clipping for Trend)	$Y = \text{Clip}(X, \epsilon)$ where ϵ is threshold to determine the significant feature to be retained
FEACLIP (Feature Extraction and Clipping)	$Y = \text{Extract}(X) \cap \text{Clip}(X, \epsilon)$

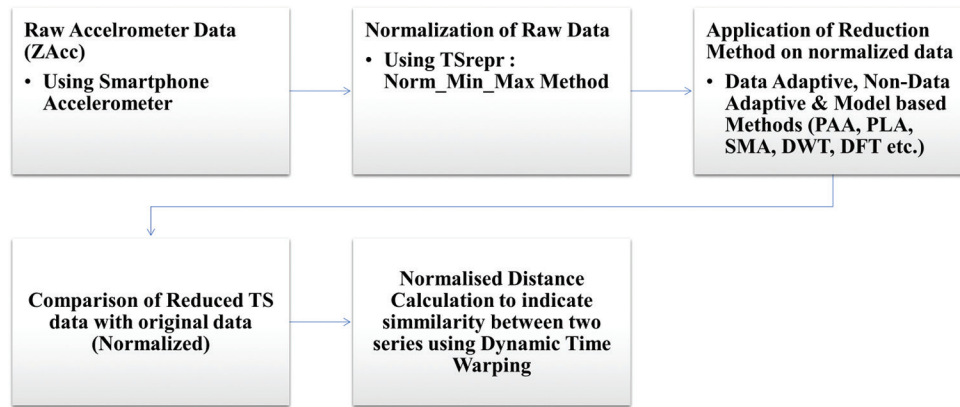


Fig. 4. Methodology flow

	PIP	PLA	SEAS	LIPTREND	FEACLIP
↓	0.0534	↓	0.0538	↓	0.0644
↑	0.2055	↑	0.2113	⇒	0.1603
↓	0.0514	↓	0.0508	↓	0.0558
⇒	0.123	⇒	0.1225	↓	0.1109
⇒	0.1267	⇒	0.1356	↓	0.1014
⇒	0.123	⇒	0.1225	↓	0.1109
↓	0.0543	↓	0.0569	↓	0.0392
⇒	0.137	⇒	0.1247	↓	0.0961
				↓	0.0609
				↓	0.2051
				↓	0.0623
				↓	0.2584
				↓	0.1636
				↓	0.2545
				↓	0.0621
				↓	0.0617
				↓	0.0808
				↓	0.0805

Fig. 5. Heat map representation of the distance between original and reduced time series

lower (ow – 20%), as shown in Table 3. The dimensionality reduction procedure is presented in Table 4.

We conducted a paired t-test to verify the results of the differences of the top five methods. The results obtained are listed as

- PIP versus PLA: Statistic: -0.216 , p-value: 0.835 : No significant difference between PIP and PLA.
- PIP versus SEAS: Statistic: 2.423 , p-value: 0.046 : There is a statistically significant difference between PIP and SEAS at the 0.05 significance level.

- PIP versus FEACLIPTREND: Statistic: -1.454 , p-value: 0.189 : No significant difference between PIP and FEACLIPTREND.
- PIP versus FEACLIP: Statistic: -1.437 , p-value: 0.194 : No significant difference between PIP and FEACLIP.

The results indicate that PIP performs significantly differently compared to SEAS, while there are no significant differences between PIP and the other methods (PLA, FEACLIPTREND, and FEACLIP).

Fig. 5 shows a heat map visualization of the percentage error term. The color scale of light yellow to green shows minimum error terms to maximum error terms. Fig. 6 shows a visualization of the time series original, PAA, SMA, and PIP.

The dimensionality reduction methods exhibit substantial potential for real-world applications and integration into existing systems. These techniques effectively reduce the dimensionality of time series data while preserving the integrity of original features, which is crucial for enhancing interpretability, efficiency, and performance across various domains.

Fig. 7 shows the interactive web application used to compare two-time series with four different distance measures.

Table 2. Experimental results

a. Non-data adaptive methods							
Normalized cumulative distance between original and reduced time series							
Time series	LEN	SMA	RDWT	DFT	DCT	PAA	PIP
TS1	101	0.0538	0.06276	0.62946	0.15076	0.05339	0.06438
TS2	101	0.21127	0.26822	0.89585	0.43408	0.20551	0.16026
TS3	101	0.05082	0.05375	0.46697	0.12836	0.05137	0.05579
TS4	843	0.12247	0.4158	1	0.993	0.12296	0.1109
TS5	1764	0.13563	0.32768	0.99997	0.67835	0.12666	0.10136
TS6	843	0.12247	0.4158	1	0.993	0.12296	0.1109
TS7	526	0.05692	0.94684	1	1	0.05432	0.03923
TS8	2000	0.12471	0.19542	0.99858	0.67039	0.13698	0.09614
b. Data adaptive methods							
Normalized cumulative distance between original and reduced time series							
Time series				LEN		PLA	
TS1				101		0.042641	
TS2				101		0.213402	
TS3				101		0.039116	
TS4				843		0.416696	
TS5				1764		0.527422	
TS6				843		0.416696	
TS7				526		0.117633	
TS8				2000		0.198984	
c. Model-based methods							
Normalized cumulative distance between original and reduced time series							
Time series	LEN		SEAS	LM		GAM	EXP
TS1	101		0.05153	0.051345		0.259189	0.179786
TS2	101		0.436253	0.436253		0.996679	0.443483
TS3	101		0.053389	0.053389		0.228889	0.196029
TS4	843		0.563937	0.563878		1	0.441416
TS5	1764		0.53348	0.533461		1	0.144275
TS6	843		0.563937	0.563878		1	0.441416
TS7	526		0.059751	0.05981		0.998337	0.299404
TS8	2000		0.16548	0.16548		0.99447	0.314765
d. Data-dictated methods							
Normalized cumulative distance between original and reduced time series							
Time series	LEN		FEATREND		FEACLIP		FEACLIPTREND
TS1	101		0.060922		0.060967		0.563687
TS2	101		0.205051		0.205281		0.999998
TS3	101		0.06234		0.062194		0.408944
TS4	843		0.258353		0.254489		1
TS5	1764		0.163569		0.162071		1
TS6	843		0.258353		0.254489		1
TS7	526		0.062134		0.061691		0.999569
TS8	2000		0.080842		0.08054		1

Table 3. Five best methods with the percentage of error term

% Error between original and reduced time series					
TS-LEN	PIP (%)	PLA (%)	SEAS (%)	FEACLIP TREND (%)	FEACLIP (%)
TS1-101	5.34	5.38	6.44	6.09	6.10
TS2-101	20.55	21.13	16.03	20.51	20.53
TS3-101	5.14	5.08	5.58	6.23	6.22
TS4-843	12.30	12.25	11.09	25.84	25.45
TS5-1764	12.67	13.56	10.14	16.36	16.21
TS6-843	12.30	12.25	11.09	25.84	25.45
TS7-526	5.43	5.69	3.92	6.21	6.17
TS8-2000	13.70	12.47	9.61	8.08	8.05

Table 4. Reduction percentage in size

TS-LEN	PIP (%)	PLA (%)	SEAS (%)	FEACLIP TREND (%)	FEACLIP (%)
TS1-101	89.11	89.11	90.10	88.12	92.08
TS2-101	89.11	89.11	90.10	88.12	92.08
TS3-101	89.11	89.11	90.10	88.12	92.08
TS4-843	15.84	89.11	16.83	88.12	92.08
TS5-1764	79.00	98.70	79.12	98.58	99.05
TS6-843	95.18	99.38	95.24	99.32	99.55
TS7-526	93.59	98.70	93.71	98.58	99.05
TS8-2000	61.79	97.91	61.98	97.72	98.48

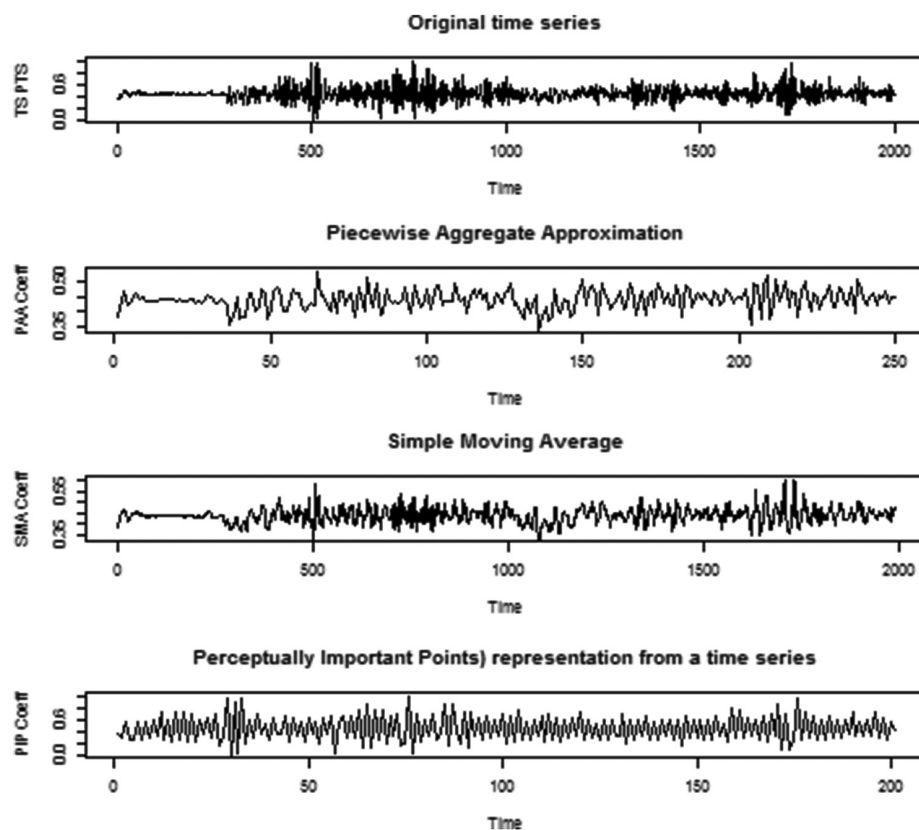


Fig. 6. Piecewise aggregate approximation, simple moving average, and perceptually important points visualization with original time series

Time Series Distance Calculator

Fig. 7. Web Application for distance calculation

The purpose of this Shiny app, titled “Time Series Distance Calculator,” is to assist users in uploading two-time series datasets, implementing a selected reduction method, and computing the similarity distance between them using a combination of DTW and Time Asynchronous Matching (TAM) distances, and presenting the results in a visual format.

The application design uses *Shiny* framework in R programming language, combined with a reactive programming model for real-time updates. The application can be accessed through Shiny Server or a browser.

This application is beneficial for individuals engaged in the analysis of time series data who require a means of assessing the resemblance between two series. This interface is valuable in domains where time series analysis holds significance. Some of the use cases for the potential use of this application are Market Data Comparison, Portfolio Management, Disease Progression Monitoring Economic Indicators, and Social Trends Analysis.

7. Conclusion and Future Scope

This research study considered 14 different methods of dimensionality reduction for time series from the TSrepr package in the “R” programming. The basis for comparison is the similarity of the reduced time series with the original time series. The results revealed that the PIP and PLA methods reduce the dimensionality of the time series by 90 – 95%. Furthermore, by comparing these time series on the basis of the combined warping path and magnitude, a novel method of time series similarity search is presented. In the future, we wish to explore other

methods of the TSrepr package for dimensionality reduction of multivariate time series.

8. Limitations

This study primarily focuses on univariate time series, which limits the generalizability of the findings to multivariate time series data, a common scenario in real-world applications. Furthermore, the basis for comparison is the similarity between the reduced and original time series, which may not account for other important aspects, such as preserving specific patterns or trends relevant to multiple domains.

References

- Ali, M., Alqahtani, A., Jones, M.W., & Xie, X. (2019). Clustering and classification for time series data in visual analytics: A survey. *IEEE Access*, 7, 181314–181338. <https://doi.org/10.1109/ACCESS.2019.2958551>
- Anand, A., Gawande, R., Jadhav, P., Shahapurkar, R., Devi, A., & Kumar, N. (2020). Intelligent vehicle speed controlling and pothole detection system. *E3S Web of Conferences*, 170, 02010. <https://doi.org/10.1051/e3sconf/202017002010>
- Ashraf, M., Anowar, F., Setu, J.H., Chowdhury, A.I., Ahmed, E., Islam, A., & Al-Mamun, A. (2023). A survey on dimensionality reduction techniques for time-series data. *IEEE Access*, 11, 42909–42923. <https://doi.org/10.1109/ACCESS.2023.3269693>
- Bairagi, V. (2018). EEG signal analysis for early diagnosis of Alzheimer disease using spectral and wavelet based features. *International Journal of Information Technology*, 10(3),403–412. <https://doi.org/10.1007/s41870-018-0165-5>
- Biemann, D.C., & Masseglia, F. (n.d.). *Time Series Clustering in the Field of Agronomy Cluster Analyse Agronomischer Zeitreihen*. Master-Thesis, p70.
- Camerra, A., Palpanas, T., Shieh, J., & Keogh, E. (2010). iSAX 2.0: Indexing and Mining One Billion Time Series. In: *2010 IEEE International Conference on Data Mining*, p58–67. <https://doi.org/10.1109/ICDM.2010.124>
- DeOliveiraMarques,E.S.,Alves,K.S.T.R.,Pekaslan,D., & De Aguiar, E.P. (2022). Kernel Evolving Participatory Fuzzy Modeling for Time Series Forecasting: New Perspectives Based on Distance Measures. In: *2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, p1–8.

- <https://doi.org/10.1109/FUZZ-IEEE55066.2022.9882602>
- Eddelbuettel, D., & François, R. (2011). Rcpp: Seamless R and C++ Integration. *Journal of Statistical Software*, 40(8), 1–18.
<https://doi.org/10.18637/jss.v040.i08>
- Giorgino, T. (2009). Computing and visualizing dynamic time warping alignments in R: The dtw package. *Journal of Statistical Software*, 31(7), 1–24.
<https://doi.org/10.18637/jss.v031.i07>
- He, Z., Zhang, C., & Cheng, Y. (2023). Similarity measurement and classification of temporal data based on double mean representation. *Algorithms*, 16(7), 347.
<https://doi.org/10.3390/a16070347>
- (n.d.). Available from: <https://acmbulletin.fiit.stuba.sk/vol10num2/vol10num2.pdf>
- Hussein, D., Nelson, L., & Bhat, G. (2024). Sensor-aware classifiers for energy-efficient time series applications on IoT devices (arXiv:2407.08715). arXiv.
<https://doi.org/10.48550/arXiv.2407.08715>
- Ines Silva, M., & Henriques, R. (2020). Exploring Time-series Motifs through DTW-SOM. In: *2020 International Joint Conference on Neural Networks (IJCNN)*, p1–8.
<https://doi.org/10.1109/IJCNN48605.2020.9207614>
- Jiménez, P., Nogal, M., Caulfield, B., & Pilla, F. (2016). Perceptually important points of mobility patterns to characterise bike sharing systems: The Dublin case. *Journal of Transport Geography*, 54, 228–239.
<https://doi.org/10.1016/j.jtrangeo.2016.06.010>
- Juliusdottir, T. (2023). topR: An R package for viewing and annotating genetic association results.
<https://doi.org/10.21203/rs.3.rs-2499681/v1>
- Laurinec, P. (2018). TSrepr R package: Time series representations. *Journal of Open Source Software*, 3(23), 577.
<https://doi.org/10.21105/joss.00577>
- Laurinec, P., & Lucka, M. (2016). Comparison of Representations of Time Series for Clustering Smart Meter Data. In: *Proceedings of the World Congress on Engineering and Computer Science (WCECS 2016)*, p6.
- Lin, J., Keogh, E., Lonardi, S., & Chiu, B. (2003). A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. In: *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD '03)*, p2.
<https://doi.org/10.1145/882082.882086>
- Matsila, H., & Bokoro, P. (2018). Load Forecasting Using Statistical Time Series Model in a Medium Voltage Distribution Network. In: *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, p4974–4979.
<https://doi.org/10.1109/IECON.2018.8592891>
- Meng, J., Huo, X., He, C., & Zhu, C. (2024). Dimension Reduction of Multi-Source Time Series Sensor Data for Industrial Process. In: *2024 IEEE 33rd International Symposium on Industrial Electronics (ISIE)*, p1–6.
<https://doi.org/10.1109/ISIE54533.2024.10595725>
- Montero, P., & Vilar, J.A. (2014). TSclust: An R package for time series clustering. *Journal of Statistical Software*, 62(1), 1–43.
<https://doi.org/10.18637/jss.v062.i01>
- Ngabesong, R., & McLauchlan, L. (2019). Implementing “R” Programming for Time Series Analysis and Forecasting of Electricity Demand for Texas, USA. In: *2019 IEEE Green Technologies Conference (GreenTech)*, p1–4.
<https://doi.org/10.1109/GreenTech.2019.8767131>
- Salvador, S., & Chan, P. (n.d.). FastDTW: Toward accurate dynamic time warping in linear time and space.
- Sharma, S.K., Phan, H., & Lee, J. (2020). An application study on road surface monitoring using DTW based image processing and ultrasonic sensors. *Applied Sciences*, 10(13), 4490.
<https://doi.org/10.3390/app10134490>
- Singh, V., & Meena, N. (2009). Engine Fault Diagnosis using DTW, MFCC and FFT. In: U. S. Tiwary, T. J. Siddiqui, M. Radhakrishna, & M. D. Tiwari (Eds.), *Proceedings of the First International Conference on Intelligent Human Computer Interaction*. Springer, India, p83–94.
https://doi.org/10.1007/978-81-8489-203-1_6
- Tanwar, H., & Kakkar, M. (2017). Performance Comparison and Future Estimation of Time Series Data Using Predictive Data Mining Techniques. In: *2017 International Conference on Data Management, Analytics and Innovation (ICDMAI)*, p9–12.
<https://doi.org/10.1109/ICDMAI.2017.8073477>
- Tonle, F., Tonnang, H., Ndadji, M., Tchendji, M., Nzeukou, A., Senagi, K., & Niassy, S. (2024). Advancing multivariate time series similarity assessment: An integrated computational approach (Version 1). arXiv.
<https://doi.org/10.48550/ARXIV.2403.11044>
- Wang, X., Ding, H., Trajcevski, G., Scheuermann, P., & Keogh, E. (2010). Experimental

comparison of representation methods and distance measures for time series data. arXiv:1012.2789 [Cs].

<https://doi.org/10.48550/arXiv.1012.2789>

Wang, Y., Xu, Y., Yang, J., Chen, Z., Wu, M., Li, X., &

Xie, L. (2023). SEnsor alignment for multivariate time-series unsupervised domain adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(8), 10253–10261.

<https://doi.org/10.1609/aaai.v37i8.26221>

AUTHOR BIOGRAPHIES



Anupama Jawale received a B.Sc. (Computer Science) and MCM (Master of Computer Management) degree from North Maharashtra University, Maharashtra, India. She received an MCA (Master of Computer Applications) from Sikkim Manipal University, Sikkim, India. She received an M.Phil. (Information Technology) degree from YCMO University, Maharashtra, India. She received PhD degree (Computer Science) from SNDT Women's University, Mumbai, India in 2024. She began her academic career in affiliated institutions of the University of Mumbai, India, as a lecturer teaching undergraduate and graduate courses in computer science and Information Technology (IT). Later in 2013, she joined as an assistant professor in the department of IT at NM College of Commerce & Economics, Mumbai, India, and currently, she is heading the IT Department. Her work aims to improve knowledge and usage of data-driven methods in computer vision, time series analysis, and human-computer interaction psychology. She has contributed to addressing complex issues in a variety of industries, such as digital security, automotive, and finance, by utilizing cutting-edge approaches and optimization techniques. She is a co-author of about a dozen papers published in journals/conferences. Her current research interests are in Data Science, Feature Extraction for



Amiya Kumar Tripathy is currently a Professor in the Department of Computer Engineering, Don Bosco Institute of Technology, Mumbai, India, affiliated with the University of Mumbai. He earned a PhD degree (2013) in Computer Science & Engineering (in the domain of Data Mining & Wireless Sensor Networks) from the Indian Institute of Technology Bombay, Mumbai, India. He was an adjunct associate professor in the faculty of Science & Engineering at Edith Cowan University (ECU), Australia (2014 – 2017) and later an adjunct professor in the School of Science, ECU, Australia (2017 – 2023). He had been a visiting researcher at the Rajamangala University of Technology, Bangkok, Thailand, for IoT-enabled remote monitoring of the Precision Agriculture Farming project (2017 – 2018). He has been in the software industry, research, and academia for more than two decades, having around 150 publications in journals/conference papers. His research focuses on data science, computer vision, remote sensing, and IoT for Precision Agriculture. He has contributed to numerous collaborative research and consultancy projects in the domain of data analytics in India and abroad. He has served on the technical program committees of several international conferences, has been invited as a plenary speaker, and has co-chaired sessions at various conferences.

Optimizing cloud-based intrusion detection systems through hybrid data sampling and feature selection for enhanced anomaly detection

Sadargari Viharika*, N. Alangudi Balaji

Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vijayawada, India

*Corresponding author E-mail: reddyviharika266@gmail.com

(Received 09 October 2024; Final version received 03 February 2025; Accepted 13 February 2025)

Abstract

To enhance detection accuracy in network intrusion scenarios, this study proposes an optimized intrusion detection system (IDS) framework that integrates advanced data sampling, feature selection, and anomaly detection techniques. Leveraging random forest (RF) and genetic algorithm, the framework optimizes sampling ratios and identifies critical features. In contrast, the isolation forest algorithm detects and excludes outliers, refining dataset quality and classification performance. Evaluated on the UNSW-NB15 dataset, comprising over 2.5 million records and 42 diverse features, the proposed framework demonstrates significant improvements in anomaly detection, including reduced false alarm rates and enhanced identification of rare threats, such as shellcode, worms, and backdoors. Experimental results reveal that the RF-based model achieves an F1 score of 91.8% and an area under the curve (AUC) of 96%, outperforming traditional machine learning models and standalone RF classifiers. The integration of extreme gradient boosting (XGB) and its optimized variant, XGBGA, further enhances the framework, with XGBGA achieving the highest performance metrics, including an F1 score of 92.8% and an AUC of 97%. These findings underscore the importance of data optimization strategies in improving the accuracy and reliability of IDSs, particularly in handling imbalanced datasets and diverse network traffic. Future work will focus on real-time processing capabilities to handle streaming data and expanding the framework's applicability to domains such as fraud detection and cybersecurity, where precise anomaly detection is essential.

Keywords: Anomaly Detection, Data Optimization, Intrusion Detection System, Machine Learning

1. Introduction

The importance of network security has escalated with the expansion of the Internet and the corresponding rise in the volume and complexity of network traffic. As part of the defense against hostile activities on networks, intrusion detection systems (IDSs) have become essential tools. These systems analyze network traffic and identify anomalies that may indicate security breaches. IDSs are generally categorized into two major groups: signature-based systems and anomaly-based systems. Signature-based IDSs, such as Snort, operate by maintaining large databases of known attack signatures and comparing incoming traffic to these predefined patterns to detect intrusions (Ahmad et al., 2021; Heidari et al., 2023; Heidari et al., 2024). These systems excel at identifying known threats, but face significant challenges in

detecting new or evolving threats due to their reliance on existing signatures.

On the other hand, anomaly-based IDSs construct models of normal network behavior and flag deviations from these models as potential threats. These systems are particularly valuable in detecting previously unknown threats as they do not depend on predefined signatures. However, anomaly-based IDSs often suffer from high false alarm rates and reduced detection accuracy, mainly due to the large volume of network data and the skewed distribution between normal and anomalous activities. This data imbalance can result in the model focusing too heavily on more common behaviors, which can reduce its ability to detect rare but important anomalies (Chkirbene et al., 2020; Junwon et al., 2022).

To address these issues, recent studies have explored integrating data optimization techniques with machine learning models to enhance IDS performance. The present study proposes a hybrid data optimization-based IDS, termed RFGA, which combines data sampling and feature selection techniques to improve the accuracy and efficiency of anomaly detection (Hassan et al., 2024). Data sampling techniques such as oversampling and undersampling are used to resolve the issue of imbalanced data distribution by either amplifying the representation of rare events or reducing the frequency of common events (Heidari et al., 2024). Feature selection further refines the model by retaining only the most relevant features that help distinguish between normal and anomalous behaviors when discarding redundant or irrelevant data. The proposed RFGA employs the random forest (RF) algorithm as its core classification technique, utilizing optimized data inputs to build a more effective and robust detection system.

The major contributions of this paper are as follows:

- 1) The paper introduces a novel RFGA framework that combines isolation forest (iForest), genetic algorithm (GA), and RF to optimize data sampling and feature selection, significantly improving intrusion detection accuracy.
- 2) The system demonstrates superior performance in detecting rare and severe network anomalies, such as backdoors, worms, and shellcodes, compared to traditional methods.
- 3) The RFGA framework is rigorously evaluated using the UNSW-NB15 dataset, showing significant reductions in false alarm rates and better handling of imbalanced datasets.

The structure of this paper is as follows: A thorough assessment of relevant work on the subject of IDSs is given in Section 2, with an emphasis on different machine learning strategies and data optimization tactics. The main techniques used in the development of RFGA are introduced in Section 3, including GA, RF, and iForest. The RFGA framework's architecture and execution are described in detail in Section 4. The effectiveness of the suggested system is illustrated by a discussion of the experimental findings and their implications. Section 5 wraps up the investigation and makes recommendations for further research avenues.

2. Literature Review

The identification and mitigation of network anomalies have been central to the development of effective IDSs. Given the growing complexity of network environments, traditional IDS approaches have increasingly been supplemented by advanced

data mining and machine learning techniques. These approaches aim to improve the detection rate of IDSs when minimizing false alarms, a balance that has proven difficult to achieve due to the inherent challenges in handling large and imbalanced datasets.

2.1. Data Sampling in IDSs

One of the primary challenges in intrusion detection is the uneven distribution of network data, where normal activities vastly outnumber anomalous ones. This imbalance can skew the performance of IDSs, making it difficult to detect rare but potentially severe threats. Data sampling techniques, such as oversampling and undersampling, have been used to tackle this issue. Oversampling methods, like the synthetic minority oversampling technique, increase the representation of minority classes by generating synthetic samples, thereby balancing the dataset (Heidari et al., 2024). Conversely, undersampling techniques reduce the number of majority class instances, as demonstrated by methods like EasyEnsemble and BalanceCascade, which selectively downsample the dataset to achieve a more balanced distribution (Heidari et al., 2024).

The effectiveness of data sampling in enhancing IDS performance has been highlighted in several studies. Research has explored the use of data sampling to improve the accuracy and speed of intrusion detection, utilizing the least squares support vector machine (SVM) to identify suspicious network activities. Findings indicated that sampling techniques could effectively select representative data subsets, thereby improving the detection capabilities of IDSs (Molina-Coronado et al., 2020). Further advancements include the integration of modified K-means clustering with machine learning techniques. The modified K-means algorithm identified common patterns across datasets, enabling more effective data compression and reducing the computational burden on the IDS. By combining K-means with the C4.5 decision tree algorithm and further enhancing detection through SVM and extreme learning machine techniques, this method significantly improved the efficacy and accuracy of IDSs, particularly in identifying Denial-of-Service attacks (Bukhari et al., 2024; Heidari et al., 2023).

2.2. Feature Selection in IDSs

Feature selection is as vital as data sampling in enhancing IDS performance. By removing superfluous or irrelevant features and selecting the most pertinent ones, feature selection approaches aim to reduce the dimensionality of the data. The filter, wrapper, and embedding methods are the primary strategies for selecting features.

Filter techniques assess each feature using statistical metrics like divergence or correlation, selecting the features most likely to improve classification performance (Deebak & Hwang, 2024). In contrast, wrapper techniques choose or eliminate features based on how well they contribute to the accuracy of the model, using a specific learning algorithm to assess their significance. Embedding approaches, such as decision trees, perform feature selection simultaneously with model training, guided by the learned weights of the features (Ahmad et al., 2021).

In IDSs, feature selection plays a crucial role, as several studies have demonstrated. For example, some studies employed logarithmic marginal density ratios to adjust initial features in SVM-based detection systems, yielding higher-quality features and improved classification efficiency (Hassan et al., 2024). Another approach used a hybrid classification algorithm that significantly enhanced the model's training data by combining correlation-based feature selection with fuzzy C-means clustering. This demonstrated how advanced machine learning techniques combined with feature selection could enhance the detection of unusual network behaviors (Hnamte et al., 2023).

Further advancements in feature selection have been achieved through integrating GA with machine learning models. GAs, known for their global optimization capabilities, have been extensively applied in network security for feature selection and parameter tuning. For instance, one application used logistic regression (LR) with GA to select the most effective feature subset, showing improved detection performance of decision tree-based methods when optimized with GA (Heidari et al., 2023). In addition, combining GA with fuzzy logic has been explored, where fuzzy logic assesses whether network events are indicative of anomalies, while GA generates digital signatures for network segments under investigation (Heidari et al., 2024).

In the context of cloud-based IDSs, hybrid approaches that combine multiple machine-learning techniques have shown promise in enhancing detection accuracy. For example, an advanced spam detection system integrated GA with a random weight network, achieving significant improvements in accuracy, precision, and recall (Hassan et al., 2024). Similarly, an IDS for wireless mesh networks combined GA-based feature selection with multiple SVM classifiers, resulting in a highly accurate and efficient detection mechanism (Bukhari et al., 2024).

3. Key Methodologies

The foundational approaches for the suggested RFGA are presented in this section. In particular, it

discusses the use of RF, GA, and iForest to improve the efficacy and precision of the IDS.

3.1. Isolation Forest

Liu et al. (2012) presented the tree-based approach known as the iForest. It is intended to provide high accuracy and minimal time complexity for locating outliers in large, highly dimensional datasets. Since anomalies are “few and different,” it is easier to isolate them from the rest of the data, which is the fundamental tenet of iForest. By recursively splitting the data, the iForest creates a series of binary trees called isolation trees (iTrees). Every tree is constructed by selecting a feature at random, followed by the selection of a random split value between the feature's maximum and minimum values (Drewek-Ossowicka et al., 2021; Ferrag et al., 2019). This random partitioning process continues until every data point is isolated in a separate leaf node. Because of their unique properties, anomalies are predicted to be separated at shorter travel lengths than typical sites. Points with shorter pathways are regarded as anomalies, and the average path length over all trees is calculated. Because of its linear time complexity and capacity to handle high-dimensional data without the need for labeled data, iForest is very useful for huge datasets. This versatility makes it extremely adaptable to a wide range of applications.

3.2. GA

The GA is a search heuristic paradigm that draws inspiration from the mechanism of natural selection. The algorithm produces solutions of superior quality for optimization and search problems by emulating the fundamental principles of biological evolution. The DO IDS framework uses genetic GA to optimize the sampling ratio and feature selection processes. The technique starts by encoding potential solutions into a genotype string structure, where different combinations of these strings represent different potential solutions or chromosomes. A randomly generated initial population of chromosomes is utilized, with each chromosome indicating a potential solution to the problem (Nguyen et al., 2020). To assess the quality of each solution, a fitness function is employed, which exhibits variability contingent upon the specific problem being addressed.

In the framework of RFGA, the fitness function is derived from the F1 score, which takes into account both precision and recall, rendering it a suitable metric for the analysis of classification tasks. The GA enhances the population by employing a selection process that prioritizes the fittest people, conducting crossover events to facilitate the interchange of genetic material between chromosomes, and introducing mutations

to uphold genetic variety. These mechanisms enable GAs to explore the solution space systematically and ultimately converge toward an optimal or nearly optimal solution through iterative generations (Fig. 1). The DO IDS framework utilizes GA to optimize the sample ratios and identify the most pertinent features, hence improving the overall accuracy of the system's detection.

3.3. RF and Extreme Gradient Boosting (XGB): Ensemble Learning Techniques

Ensemble learning methods combine the predictions of multiple individual models to improve overall performance. Both RF and XGB are robust ensemble algorithms that utilize decision trees, but they differ significantly in their approach. Random RF, developed by Leo Breiman, builds an ensemble of decision trees using a technique called bootstrap sampling, where each tree is trained on a different subset of the data sampled with replacement. During the construction of each tree, a random subset of features is selected at every split, ensuring that each tree is independent, thereby reducing inter-tree correlation. This randomness enhances the diversity of the trees and improves the overall model's predictive performance. RF is particularly known for its resistance to overfitting and ability to provide valuable insights into the importance of features. The algorithm is effective for both binary and multi-class classification tasks and has been widely adopted due to its simplicity, robustness, and ease of use.

On the other hand, XGB follows a different approach by using boosting instead of bagging. While RF constructs each tree independently, XGB builds trees sequentially, where each new tree is trained to correct the errors made by the previous one. This

boosting mechanism allows XGB to focus more on the difficult-to-classify samples, iteratively improving model performance with each tree. XGB has gained significant popularity due to its scalability, efficiency, and ability to handle complex data with high accuracy. The algorithm incorporates regularization techniques like L1 (alpha) and L2 (lambda) to prevent overfitting, making it particularly suitable for high-dimensional datasets. In addition, XGB includes features like subsampling and `colsample_bytree`, which allow for further optimization of the model's performance by randomly sampling subsets of the training data and features at each boosting round.

Despite the differences in their approaches, both RF and XGB offer mechanisms to evaluate feature importance, which is crucial for identifying the most relevant features for a given classification task. This ability helps practitioners improve model interpretability and selection of key performance indicators. While RF tends to be more interpretable due to its simple randomization process, XGB generally provides superior predictive accuracy, especially for more complex tasks. In practical applications, both algorithms can be tuned for optimal performance, and they can be highly effective for tasks like anomaly detection in network security.

3.4. Proposed RFGA Framework

The proposed RFGA framework represents a novel approach to intrusion detection, integrating GA with RF to address the challenges of imbalanced data and improve detection accuracy. Unlike traditional models, the RFGA framework leverages the optimization capabilities of GA to refine both data sampling and feature selection processes, ensuring a robust classification system. In addition, experiments incorporate XGB and an enhanced version, XGBGA, along with traditional models such as LR, Naïve Bayes (NB), K-nearest neighbors (KNN), SVM, and DT. This comprehensive comparison ensures that the RFGA framework's performance is thoroughly evaluated across a diverse range of methodologies.

3.4.1. Data sampling and optimization using GA

In the RFGA framework, GA plays a pivotal role in optimizing data sampling ratios and feature selection, addressing the critical issue of imbalanced datasets. The fitness function used is the F1 score, a balanced metric that combines precision and recall to assess classification performance. GA initiates with a population of potential solutions, where each chromosome encodes a candidate sampling ratio or feature subset. Through iterative processes of selection, crossover, and mutation, GA evolves these

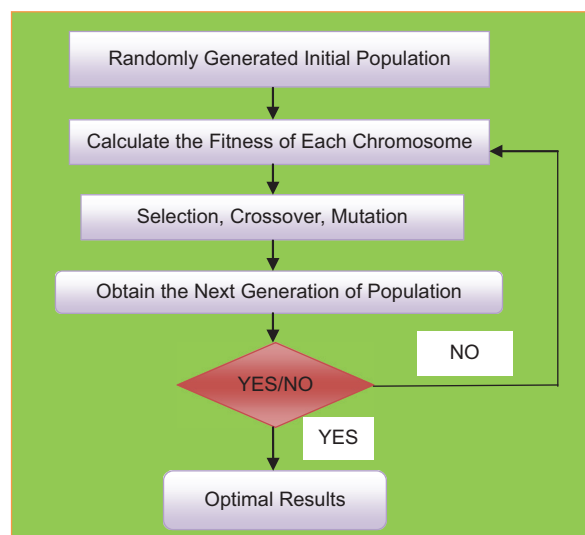


Fig. 1. Workflow diagram

chromosomes to converge on optimal solutions. By refining sampling ratios, GA ensures that rare anomalies are adequately represented in the training data, significantly enhancing the system's ability to detect unusual patterns within network traffic. This systematic approach not only addresses class imbalance but also improves the overall accuracy and reliability of the IDS.

3.4.2. Feature selection using GA

Feature selection is a critical component of the RFGA framework, designed to reduce data dimensionality and enhance classification accuracy by focusing on the most relevant features. Initially, the dataset undergoes preprocessing using techniques like outlier detection to remove noise. Subsequently, GA optimizes the feature subset selection process, where each chromosome encodes a subset of features, and the fitness function evaluates the classifier's performance using these features. This iterative optimization identifies the most pertinent features, which are then used to train classifiers. The reduction in data dimensionality not only decreases computational complexity but also improves detection performance by eliminating irrelevant or redundant features. This targeted feature selection ensures that the RFGA framework achieves superior accuracy when maintaining computational efficiency.

3.4.3. RF classifier training

The RF algorithm, developed by Leo Breiman, is an ensemble learning method that combines multiple decision trees to improve classification accuracy and robustness. Known for its effectiveness in both binary and multi-class tasks, RF's design inherently mitigates overfitting. The construction of an RF model involves generating numerous decision trees, each trained on a unique bootstrap-sampled subset of the data, sampled with replacement. At each tree node, a random subset of features is chosen, and the best split is selected based on this subset, reducing inter-tree correlation and enhancing model diversity. Once trained, each tree votes on the predicted class label, and the final classification is determined by the majority vote across all trees. RF's structure offers several advantages, including high accuracy, resistance to overfitting, and the ability to process large datasets with numerous features. In addition, RF provides valuable insights into feature importance, which supports feature selection within the RFGA framework and refines the system's accuracy and interpretability.

Fig. 2 illustrates the role of IDS in securing network environments by analyzing network traffic to detect potential threats. IDS is typically classified into

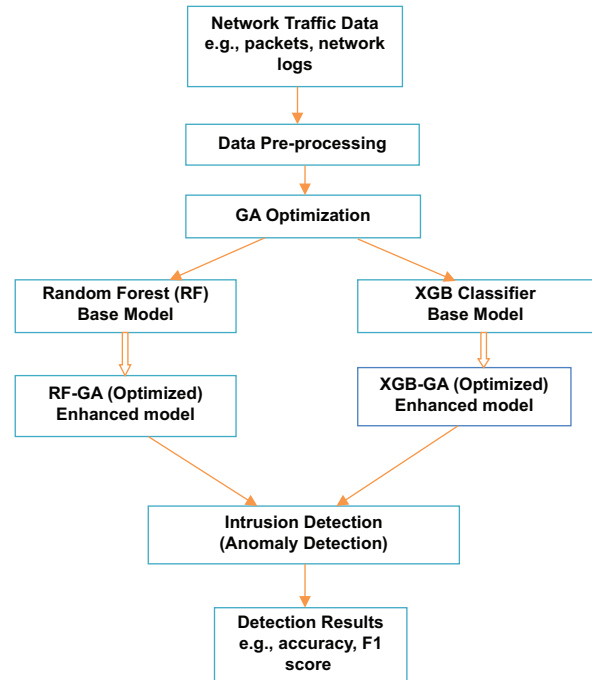


Fig. 2. Intrusion Detection System (IDS) Framework with GA Optimization for RF and XGB Classifiers

two types: signature-based IDS, which identifies threats by comparing data patterns against known signatures, and anomaly-based IDS, which flags deviations from normal network behavior that might indicate new or unknown threats. The RFGA framework employs a hybrid optimization approach, integrating data sampling and feature selection techniques to improve the performance of anomaly detection, particularly by reducing false alarm rates and enhancing detection accuracy.

Algorithm 1. Optimized Intrusion Detection System (XGBGA)

Input: Network traffic data $[X_1, X_2, \dots, X_n]$
Output: Optimized intrusion detection with high accuracy
Data Sampling:
• Apply Isolation Forest (iForest) to detect and remove outliers.
• Use Genetic Algorithm (GA) to optimize sampling ratios based on the F1 score.
Feature Selection:
• Encode feature subsets using GA.
• Optimize feature selection to maximize F1 score.
Classifier Training:
• Train Random Forest (RF) or Train extreme gradient boosting (XGB) with the optimized data.
• Use the trained RF or XGB model to classify and detect anomalies.
Result: Improved intrusion detection with enhanced accuracy and reduced false alarms.

Table 1. Literature review comparison

Study (Year), dataset	Methodology	F1 score (%)	Key contributions
Bukhari et al. (2024), WSN	SCNN-BiLSTM (Federated learning)	92.6	Secure IDS for wireless sensor networks
Hanafi et al. (2024), IoT	Binary Golden Jackal+LSTM	92.3	Improved IoT intrusion detection
Belouch & Hadaj (2017), NSL-KDD	Ensemble learning	88.4	Comparison of ensemble methods for IDS
Wu (2020), UNSW-NB15	Deep learning (CNN, RNN)	91.1	IDS using computational intelligence
Vashishtha et al. (2023), Cloud IDS	Hybrid (RF+CNN)	93.5	Hybrid IDS for cloud with feature selection
Hnamte et al. (2023), KDD Cup 99	LSTM-AE	91.7	Two-stage deep learning IDS
Talukder et al. (2023), CICIDS 2017	Hybrid machine learning	94.0	Reliable IDS hybrid model
Henry et al. (2023), UNSW-NB15	Hybrid deep learning+Feature optimization	92.6	IDS with feature optimization
Hnamte & Hussain (2023), KDD Cup 99	Deep CNN-BiLSTM	94.5	Hybrid CNN-BiLSTM IDS model.
Mohamed & Ismael (2023), IoT	Fog-to-cloud computing	90.2	Hybrid IoT IDS based on fog-to-cloud
Wang et al. (2023), NSL-KDD	RF+Autoencoder	92.0	Hybrid RF-Autoencoder IDS
Mehmood et al. (2022), UNSW-NB15	Hybrid (RF+SVM)	91.3	Hybrid RF-SVM IDS model
Zhang & Wei (2021), UNSW-NB15	XGBoost	93.1	Optimized XGBoost IDS model
Singh et al. (2020), KDD Cup 99	RF with SMOTE	89.9	SMOTE applied to RF for imbalanced data
Ahmed et al. (2018), NSL-KDD	SVM with feature selection	88.8	Feature selection with SVM for IDS
Sharma et al. (2024), IoT	BiLSTM with attention	94.2	BiLSTM with attention to IoT IDS

Abbreviations: BiLSTM: Bidirectional long short-term memory; CNN: Convoluted neural network; IDS: Intrusion detection system; IoT: Internet of Things; RF: Random forest; RNN: Recurrent neural network; SMOTE: Synthetic minority oversampling technique; SVM: Support vector machine; XGBoost: Extreme gradient boosting.

4. Experimental Setup

4.1. Data Sampling

In this initial phase, the iForest method is employed to identify and eliminate outliers, aiming to reduce the impact of data imbalance on the classification process. GA is then used to optimize the sampling ratio for each class, with chromosomes representing possible sampling ratios and genes corresponding to the proportion of outliers in the sample. The F1 score is used as the fitness function to evaluate the performance of different sampling ratios, and GA iteratively refines these ratios to maximize the F1 score. The sampling ratios and their effectiveness are summarized in Table 2.

4.2. Feature Selection

Feature selection is another critical step in the DO IDS framework, aimed at reducing the dimensionality of the data and eliminating irrelevant or redundant features. In this process, each chromosome represents a subset of features, with genes indicating whether a feature is included

Table 2. Optimal sampling ratios for random forest and extreme gradient boosting

Class	Optimal sampling ratio (Random forest)	Optimal sampling ratio (Extreme gradient boosting)
Anomalous 1	0.85	0.88
Normal 1	0.92	0.94
Anomalous 2	0.78	0.80
Normal 2	0.89	0.91
Anomalous 3	0.88	0.90
Normal 3	0.91	0.93
Anomalous 4	0.80	0.83
Normal 4	0.94	0.96

or excluded. The fitness function, based on the F1 score, evaluates the classifier's performance using the selected feature subset. GA searches for the optimal feature subset, which is then used to train the RF classifier. The optimal feature subsets for each class are detailed in Table 3.

4.3. Classifier Training

Finally, the RF classifier is trained using the refined dataset and feature subsets. Through this optimization, the RF classifier achieves enhanced accuracy in detecting a wide range of anomalies. The majority voting system in RF ensures reliable classification, where each decision tree votes on the predicted class label. Fig. 3 illustrates the training process and the integration of optimized components in the RFGA framework, highlighting the combined power of iForest, GA, and RF to achieve robust intrusion detection. The overall training process and the integration of the optimized components are illustrated in Fig. 3.

4.4. Dataset Description

The experimental evaluation of the RFGA framework was conducted on a system equipped with an Intel Core i5-4460 CPU at 3.6 GHz and 8GB of RAM, running Python applications within the PyCharm IDE. The UNSW-NB15 dataset, developed by the Australian Centre for Cyber Security, was selected for evaluation due to its comprehensive representation of modern network traffic patterns. This dataset comprises 2,540,044 records, divided into training and testing subsets, with 42 features detailing various network behaviors. A detailed description of the dataset, including sample size, feature attributes,

and label distribution, is provided in Table 4. By combining this information, the RFGA framework ensures transparency and thorough evaluation of its effectiveness across diverse scenarios.

The experimental results of the RFGA framework demonstrate its superior performance in intrusion detection compared to traditional models and state-of-the-art approaches. Specifically, the number of total, training, and testing samples used in the experiment are as follows: a total of 2,540,044 samples, with 1,750,000 used for training and 790,044 reserved for testing. These figures ensure a representative distribution of both normal and anomalous network traffic, which is critical for evaluating the performance of the detection models.

The impact of optimal sampling ratios on the results is significant. These ratios, which determine how the data are sampled for training the models, are optimized through the GA to address the issue of imbalanced datasets, where anomalous events are far less frequent than normal events. The optimization of sampling ratios ensures that anomalies are adequately represented in the training data, improving the model's ability to detect rare and novel threats. However, the reason the sampling ratios are not the same for RF and XGB is attributed to the different nature of these models. While RF builds trees independently from random subsets of data, XGB follows a boosting process, where each new tree corrects errors made by the previous one. As a result, the models interact differently with the data and benefit from distinct sampling strategies, leading to optimized ratios that are specific to each model's architecture and learning approach.

The results, as shown in the performance metrics, reflect the effectiveness of these optimized sampling ratios. Both RFGA and XGBGA significantly outperform traditional machine learning models such as LR, NB, KNN, and SVM. In particular, XGBGA achieves the highest overall performance, with the highest precision, recall, and F1 score among all models. These superior results highlight how both the optimal sampling ratios and feature selection processes, tailored to each classifier, contribute to better performance in detecting intrusions when minimizing false alarms.

4.5. Challenges and Future Directions

Despite its promising results, the RFGA framework has certain limitations. First, the

Table 3. Optimized feature sets for random forest and extreme gradient boosting

Class	Selected features (Random forest)	Selected features (Extreme gradient boosting)
Anomaly group 1	Features 1, 3, 7, and 9	Features 1, 3, 7, 9, and 10
Normal group 1	Features 2, 5, 6, and 8	Features 2, 4, 6, 8, and 11
Anomaly group 2	Features 1, 4, and 10	Features 1, 4, 10, and 12
Normal group 2	Features 2, 3, and 11	Features 3, 5, 7, and 11
Anomaly group 3	Features 3, 6, and 12	Features 3, 6, 8, and 9
Normal group 3	Features 4, 7, and 8	Features 4, 7, 9, and 11
Anomaly group 4	Features 5, 9, and 11	Features 5, 8, 9, and 12

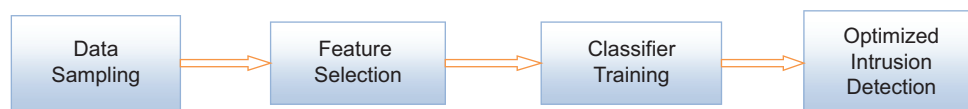


Fig. 3. Training process and integration of optimized components

Table 4. Detailed dataset description

Feature name	Description
Duration	Total duration of the connection (in seconds).
Protocol type	Type of protocol (e.g., TCP, UDP, ICMP).
Service	Network service on the destination (e.g., HTTP, FTP, SMTP).
Flag	Status flag for the connection.
Source bytes	Number of bytes transferred from source to destination.
Destination bytes	Number of bytes transferred from destination to source.
Land	1 if the source and destination IP/port are identical, otherwise 0.
Wrong fragment	Number of incorrect fragments in the connection.
Urgent	Number of urgent packets in the connection.
Hot	Number of "hot" indicators (e.g., logins, file accesses).
Failed logins	Number of failed login attempts.
Logged in	1 if successfully logged in, otherwise 0.
Root shell	1 if root shell access is obtained, otherwise 0.
File creation	Number of file creation operations.
Shell commands	Number of shell command operations.
Accessed files	Number of accessed files.
Outbound commands	Number of outbound commands in an FTP session.
Is host login	1 if the login belongs to the host, otherwise 0.
Is guest login	1 if the login is a guest login, otherwise 0.
Count	Number of connections to the same host as the current connection.
Same host rate	Percentage of connections to the same host.
Same service rate	Percentage of connections to the same service.
Diff service rate	Percentage of connections to different services.
Source failures	Number of failed source connections.
Destination errors	Number of error responses from destination.
Avg packet size	Average size of packets exchanged.
Total packets	Total packets exchanged in the connection.

computational demands of GA can hinder scalability in large-scale environments. Second, the reliance on

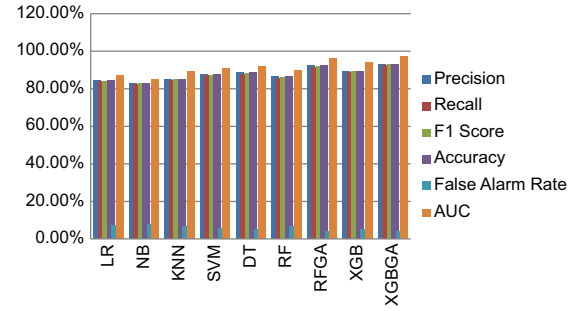


Fig. 4. Model accuracy and false alarm rate comparison

offline processing restricts its applicability to real-time anomaly detection. Finally, the evaluation is limited to the UNSW-NB15 dataset, which may not fully represent all network environments. Addressing these limitations is crucial for future development. Future work will focus on developing real-time processing capabilities to enable the detection of anomalies in streaming data. In addition, efforts will aim to optimize the computational efficiency of the RFGA framework, ensuring scalability for deployment in large-scale environments. Validation of the framework using diverse datasets and its application to other domains, such as fraud detection and cybersecurity, will further enhance its robustness and versatility. By addressing these areas, the RFGA framework can be extended to meet the demands of evolving network security challenges. In conclusion, the RFGA framework demonstrates significant advancements in intrusion detection by integrating GA with RF and XGB to optimize data sampling and feature selection. Experimental results confirm its superiority over traditional and state-of-the-art models, making it a reliable and robust solution for network security. Future enhancements will focus on scalability, real-time processing, and broader applicability to ensure its continued relevance and effectiveness.

Despite the significant advancements in IDS technology, challenges persist in achieving real-time anomaly detection, particularly in cloud environments, where the volume and diversity of network traffic continue to grow. Existing data stream management systems often struggle to process network streams quickly enough to detect anomalies in real time, a limitation exacerbated by the computational complexity of anomaly detection algorithms and the high false-positive rates associated with traditional detection methods (Heidari et al., 2024).

To address these challenges, hybrid data processing approaches that combine convolutional neural networks with optimization techniques, such as grey wolf optimization, have been proposed. These approaches, such as the TopoMAD stochastic seq2seq model, utilize advanced machine learning

Table 5. Performance metrics, including area under the curve

Model	Precision (%)	Recall (%)	F1 score (%)	Accuracy (%)	False alarm rate (%)	Area under the curve
Logistic regression	84.3	83.7	84.0	84.3	7.2	0.87
Naïve-bayes	82.9	82.3	82.6	82.9	7.8	0.85
K-nearest neighbor	85.1	84.5	84.8	85.1	6.8	0.89
Support vector machine	87.6	86.9	87.2	87.6	5.5	0.91
Decision trees	88.5	87.9	88.2	88.5	5.1	0.92
Random forest	86.7	85.4	85.8	86.7	6.9	0.90
RFGA	92.4	91.2	91.8	92.4	4.2	0.96
Extreme gradient boosting	89.5	88.7	89.1	89.5	5.0	0.94
XGBGA	93.1	92.5	92.8	93.1	3.8	0.97

techniques to capture the spatial and temporal dependencies of network data, enabling more robust and accurate anomaly detection (Vashishtha et al., 2023). Furthermore, the integration of autoencoders with traditional machine learning models has shown potential in enhancing the resilience of IDSs to corrupted data, improving their ability to detect anomalies in complex network environments (Wang et al., 2023).

In conclusion, the ongoing evolution of IDS technology is driven by the need for more accurate, efficient, and scalable detection systems. The integration of data sampling, feature selection, and advanced machine learning techniques offers a promising pathway toward achieving these goals. The proposed DO IDS framework, which combines these approaches, represents a significant step forward in the development of robust and reliable IDSs, particularly in the context of cloud computing and other large-scale network environments.

5. Conclusion

The RFGA framework, which integrates iForest, GA, and RF, has proven highly effective in enhancing the accuracy and robustness of IDSs by optimizing data sampling and feature selection, particularly for imbalanced and high-dimensional datasets. Experimental results confirm that RFGA outperforms traditional machine learning models, notably in detecting rare network anomalies like shellcode, worms, and backdoors. By combining GA to optimize sampling ratios and feature subsets with RF's strong classification capabilities, RFGA offers a reliable solution for network security. Moving forward, the primary goals are to develop real-time processing capabilities to detect anomalies in streaming data and to adapt the framework to other critical applications, such as fraud detection, where the accuracy and handling of imbalanced data are essential. In addition,

future efforts will aim to reduce computational demands, thus enhancing the framework's scalability and making it practical for deployment across large-scale environments.

References

- Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, 32, e4150.
- Belouch, M., & Hadaj, S.E. (2017). Comparison of ensemble learning methods applied to network intrusion detection. In: *Proceedings of the ACM Conference*, pp. 1–4.
- Bukhari, S.M.S., Zafar, M.H., Abou Houran, M., Moosavi, S.K.R., Mansoor, M., Muaaz, M., & Sanfilippo, F. (2024). Secure and privacy-preserving intrusion detection in wireless sensor networks: Federated learning with SCNN-BiLSTM for enhanced reliability. *Ad Hoc Networks*, 155(103), 407.
<https://doi.org/10.1016/j.adhoc.2024.103407>
- Chkirbene, Z., Erbad, A., Hamila, R., Mohamed, A., Guizani, M., & Hamdi, M. (2020). TIDCS: A dynamic intrusion detection and classification system based feature selection. *IEEE Access*, 8, 95864–95877.
<https://doi.org/10.1109/ACCESS.2020.2994931>
- Deebak, B.D., & Hwang, S.O. (2024). Healthcare applications using blockchain with a cloud-assisted decentralized privacy-preserving framework. *IEEE Transactions on Mobile Computing*, 23(5), 5897–5916.
<https://doi.org/10.1109/TMC.2023.3315510>
- Dey, A. (2020). Deep IDS: A Deep Learning Approach for Intrusion Detection Based on IDS 2018. In: *2020 2nd International Conference on*

- Sustainable Technologies for Industry 4.0 (STI)*. IEEE, p. 1–5.
- Drewek-Ossowicka, A., Pietrolaj, M., & Rumiński, J. (2021). A survey of neural networks usage for intrusion detection systems. *Journal of Ambient Intelligence and Humanized Computing*, 12, 497–514.
<https://doi.org/10.1007/s12652-020-02014-x>
- Ferrag, M.A., Maglaras, L., Janicke, H., & Smith, R. (2019). Deep Learning Techniques for Cyber Security Intrusion Detection: A Detailed Analysis. In: *6th International Symposium for ICS SCADA Cyber Security Research (ICS-CSR 2019)*, Athens, 10–12 September.
- Halbouni, A., Gunawan, T.S., Habaebi, M.H., Halbouni, M., Kartiwi, M., & Ahmad, R. (2022). CNN-LSTM: Hybrid deep neural network for network intrusion detection system. *IEEE Access*, 10, 99837–99849.
- Hanafi, A.V., Ghaffari, A., Rezaei, H., Valipour, A., & Arasteh, B. (2024). Intrusion detection in Internet of things using improved binary golden jackal optimization algorithm and LSTM. *Cluster Computing*, 27(3), 2673–2269.
<https://doi.org/10.1007/s10586-023-04102-x>
- Hassan, S.R., Rehman, A.U., Alsharabi, N., Arain, S., Quddus, A., & Hamam, H. (2024). Design of load-aware resource allocation for heterogeneous fog computing systems. *PeerJ Computer Science*, 10, e1986.
<https://doi.org/10.7717/peerj-cs.1986>
- Heidari, A., Jafari Navimipour, N., Dag, H., & Unal, M. (2024). Deepfake detection using deep learning methods: A systematic and comprehensive review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 14, e1520.
<https://doi.org/10.1002/widm.1520>
- Heidari, A., Navimipour, N.J., & Unal, M. (2023). A secure intrusion detection platform using blockchain and radial basis function neural networks for the internet of drones. *IEEE Internet of Things Journal*, 10, 8445–8454.
<https://doi.org/10.1109/JIOT.2023.3237661>
- Henry, A., Gautam, S., Khanna, S., Rabie, K., Shongwe, T., Bhattacharya, P., Sharma, B., & Chowdhury, S. (2023). Composition of hybrid deep learning model and feature optimization for intrusion detection system. *Sensors*, 23(2), 890.
<https://doi.org/10.3390/s23020890>
- Hnamte, V., & Hussain, J. (2023). DCNNBiLSTM: An efficient hybrid deep learning-based intrusion detection system. *Telematics and Informatics Reports*, 10, 100053.
<https://doi.org/10.1016/j.teler.2023.100053>
- Hnamte, V., Nhung-Nguyen, H., Hussain, J., & Hwa-Kim, Y. (2023). A novel two-stage deep learning model for network intrusion detection: LSTM-AE. *IEEE Access*, 11, 37131–37148.
<https://doi.org/10.1109/ACCESS.2023.3266979>
- Liu, F., Ting, K.M., & Zhou, Z.H. Isolation forest. In: *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM)*, IEEE, 2012, p. 413–422.
- Mehmood, M., Javed, T., Nebhen, J., Abbas, S., Abid, R., Bojja, G.R., & Rizwan, M. (2022). A hybrid approach for network intrusion detection. *Computers, Materials and Continua*, 70, 91–107.
<https://doi.org/10.32604/cmc.2022.019127>
- Mohamed, D., & Ismael, O. (2023). Enhancement of an IoT hybrid intrusion detection system based on fog-to-cloud computing. *Journal of Cloud Computing*, 12(1), 41.
<https://doi.org/10.1186/s13677-023-00420-y>
- Molina-Coronado, B., Mori, U., Mendiburu, A., & Miguel-Alonso, J. (2020). Survey of network intrusion detection methods from the perspective of the knowledge discovery in databases process. *IEEE Transactions on Network and Service Management*, 17(4), 2451–2479.
<https://doi.org/10.1109/TNSM.2020.3016246>
- Pingale, S.V., & Sutar, S.R. (2022). Analysis of Web Application Firewalls, Challenges, and Research Opportunities. In: *Proceedings of the 2nd International Conference on Data Science, Machine Learning and Applications (ICDSMLA 2020)*. Singapore: Springer, p. 239–248.
- Pingale, S.V., & Sutar, S.R. (2022). Automated network intrusion detection using multimodal networks. *International Journal of Computational Science and Engineering*, 25(3), 339–352.
<https://doi.org/10.1504/IJCSE.2022.123123>
- Pingale, S.V., & Sutar, S.R. (2022). Remora whale optimization hybrid deep learning for network intrusion detection using CNN features. *Expert Systems with Applications*, 210, 118476.
<https://doi.org/10.1016/j.eswa.2022.118476>
- Pingale, S.V., & Sutar, S.R. (2023). Remora-based Deep Maxout Network model for network intrusion detection using convolutional neural network features. *Computers and Electrical Engineering*, 110, 108831.
<https://doi.org/10.1016/j.compeleceng.2023.108831>
- Ravikumar, C., Ravi Kumar, R., Sarada, M., Pabba, K., & Pasha, M.A. (2024). A comprehensive exploration of machine learning in early detection with a focus on lung and pancreatic cancer for revolutionizing cancer diagnostics. *International Conference on Emerging Technologies in Computer Science for Interdisciplinary Applications (ICETCS 2024)*.
- Ravikumar, C.H., Batra, I., & Malik, A. (2023). Block

- chain based secure with improvised bloom filter over a decentralized access control network on a cloud platform. *Journal of Engineering Science and Technology Review*, 16(2), pp. 123–130. <https://doi.org/10.25103/jestr.162.16>
- Ravikumar, C.H., Sridevi, M., Ramchander, M., Ramesh, V., & Kumar, V.P. (2024). Enhancing digital security using signa-deep for online signature verification and identity authentication. *International Journal of Systematic Innovation*, 8(2), pp. 58–69. [https://doi.org/10.6977/IJoSI.202406_8\(2\).0005](https://doi.org/10.6977/IJoSI.202406_8(2).0005)
- Rekha, G., Malik, S., Tyagi, A.K., & Nair, M.M. (2020). Intrusion detection in cyber security: Role of machine learning and data mining in cyber security. *Advances in Science, Technology and Engineering Systems Journal*, 5(3), 72–81. <https://doi.org/10.25046/aj050310>
- Talukder, M.A., Hasan, K.F., Islam, M.M., Uddin, M.A., Akhter, A., Yousuf, M.A., Alharbi, F., & Moni, M.A. (2023). A dependable hybrid machine learning model for network intrusion detection. *Journal of Information Security and Applications*, 72, 103405. <https://doi.org/10.1016/j.jisa.2022.103405>
- Vashishtha, L.K., Singh, A.P., & Chatterjee, K. (2023). HIDM: A hybrid intrusion detection model for cloud-based systems. *Wireless Personal Communications*, 128, 2637–2666. <https://doi.org/10.1007/s11277-022-10063-y>
- Wang, C., Sun, Y., Wang, W., Liu, H., & Wang, B. (2023). Hybrid intrusion detection system based on combination of random forest and autoencoder. *Symmetry*, 15(3), 568.
- Wu, P. (2020). *Deep Learning for Network Intrusion Detection: Attack Recognition with Computational Intelligence* (PhD Thesis). UNSW Sydney.
- Xu, Z., Zhang, W., Li, Y., & Li, W. (2024). Secure and efficient intrusion detection in IoT using deep reinforcement learning. *Journal of Computer Science and Technology*, 39(3), 552–570.

AUTHOR BIOGRAPHIES



Ms. Sadargari Viharika. She received her bachelor's degree from ECEW, JNTUH in 2014. She attained her M.Tech degree from MRIET, JNTUH, and Hyderabad in 2020. She is pursuing her Ph.D. in the Computer Science and Engineering Department at KL University from 2022. Presently, she is working as an Assistant Professor in the Department of Information Technology at MLR Institute of Technology. Her areas of interest include Cloud Computing, Machine Learning, Deep Learning, and Artificial Intelligence. She can be contacted at reddiviharika266@gmail.com



N. Alangudi Balaji. He is affiliated with the Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram. Dr. N. Alangudi Balaji is an academic and researcher affiliated with the Department of Computer Science and Engineering at Koneru Lakshmaiah Education Foundation in Vaddeswaram, Andhra Pradesh, India. With over two decades of experience in engineering education, his primary focus areas include cloud computing, machine learning, and information security. He has published extensively on deep learning models, convolutional neural networks, and Internet of Things (IoT) applications in various reputed journals indexed under Scopus and Web of Science. E-Mail: alangudibalaji@gmail.com

YOLOXpress: A lightweight real-time unmanned aerial vehicle detection algorithm

Nguyen Tien Tai, Bui DucThang, Nguyen Ngoc Hung*

Institute of Control Engineering, Le Quy Don Technical University, Hanoi, Vietnam

*Corresponding author E-mail: hungnn@lqdtu.edu.vn

(Received 14 October 2024; Final version received 24 January 2025; Accepted 24 February 2025)

Abstract

The widespread use of drones has made drone detection a critical factor in various fields, particularly in security and defense. However, this task presents unique challenges due to the high speed, small size, and ability of drones to blend into their surroundings, which can hinder detection effectiveness. This paper introduces enhancements to the You Only Look Once (YOLO)-v8 model to improve real-time drone detection capabilities, especially when deployed on resource-constrained devices. We propose an improved model called YOLOXpress, which optimizes both processing speed and model size while maintaining an acceptable level of accuracy. By replacing the Cross-Stage Feature Fusion modules in the Backbone and Neck with Re-parameterization Convolution and RepC3 modules, we significantly reduced the number of computations, achieving a 12.25% increase in processing speed (frames per second) and a 69.96% reduction in model size. Although there was a 6% decrease in average accuracy compared to the original YOLO-v8 model, YOLOXpress remained effective for real-time drone detection. Experiments conducted on the TIB-Net dataset confirmed that this model is highly suitable for deployment on resource-limited devices, such as compact embedded systems.

Keywords: Drone Detection, Deep Learning, Real-Time Processing, Unmanned Aerial Vehicle, YOLO-v8

1. Introduction

The application of scientific and technological advancements in drone production is becoming increasingly widespread. Along with its exceptional advantages, drone production also comes with unforeseeable consequences. Small drones have become more prevalent in recent years, with various models performing various tasks. This directly threatens the security of many countries, as drones can be used for espionage, surveillance, and suicide missions, often equipped with weapons to target key locations with deliberate intent, thus forming new and unconventional methods of warfare (Al-Iqubaydhi et al., 2024). To counter the threats posed by drones, developing and implementing anti-drone systems has become an urgent priority in modern defense and security.

Detecting drones and providing early warnings have received considerable attention from various research groups and have been extensively studied. In

recent years, many studies have utilized deep-learning models for drone detection. Research employing computer vision and deep learning models, such as You Only Look Once (YOLO)-v3 (Alsanad et al., 2022), YOLO-v5 (Lv et al., 2022), and YOLO-v8 (Kim et al., 2023), has yielded promising results.

However, most present anti-drone devices face limitations, such as fixed installation requirements, large size, easy detectability, and difficulty in deployment in areas with space constraints. These limitations directly affect and reduce the effectiveness of the devices. Therefore, developing new deep-learning models with smaller sizes, real-time processing speeds, and higher accuracy for drone detection is essential. Developing such models would help reduce hardware resource usage when designing new anti-drone devices, thereby addressing the issue of device size limitations.

This paper is organized as follows: Section 1 introduces the research problem addressed in this study. Section 2 summarizes related works on unmanned

aerial vehicle (UAV) detection; Section 3 begins with an overview of the methodology used in this research, followed by a presentation of the YOLOv8 network architecture, including detailed descriptions of its key modules. This section also presents the improved UAV target detection model and its architecture. Section 4 introduces the dataset and experimental setup, followed by ablation studies and comparison experiments using the publicly available TIB-Net dataset. It concludes with experiments conducted on a self-constructed dataset to validate the feasibility of the proposed method thoroughly. Finally, Section 5 summarizes the research findings and outlines potential future research directions.

2. Related Work

The task of object detection involves identifying objects within a specific frame. One commonly used method is leveraging convolutional neural networks (CNN) to extract and detect object features. Since the 2010s, as deep learning has advanced, the quality of object detection algorithms has continuously been upgraded and improved, with notable algorithms such as Region-Based CNN (RCNN) and faster RCNN. Although these networks have superior performance in terms of accuracy compared to classical algorithms, their complex structures hinder the models from achieving speeds equivalent to real-time processing. This is a critical requirement for real-world applications in object detection. Therefore, many studies have focused on creating models that balance speed and accuracy to enable wide practical implementation (Lee et al., 2019; Zhai et al., 2023; Zhu et al., 2021).

At present, the YOLO series of models has effectively addressed this problem. The YOLO series has undergone nine iterations of improvement, with several minor versions showing superior performance in both speed and accuracy (Terven et al., 2023). These models are widely applied across various fields, including medicine, transportation, industry, and UAV detection and warning systems. Research groups have conducted numerous studies on applying YOLO for UAV detection. Some studies have shown promising results, such as PaddlePaddle (PP)-YOLO (Long et al., 2020), an object detection method based on YOLOv3, optimized and improved to balance performance and efficiency. PP-YOLO employs existing techniques to improve object detection accuracy without increasing the number of model parameters and computations. With a mean average precision (mAP) performance of 45.2% and frames per second (FPS) speed of 72.9, PP-YOLO surpasses existing detectors, such as EfficientDet and YOLOv4. For Mob-YOLO (Liu et al., 2022), the authors proposed a lightweight model, an object detection method for UAVs. Based on

the high-performance YOLOv4 model, MobileNetv2 (Sandler et al., 2018), a lightweight CNN, is used to replace the original CSPDarknet53 (Bochkovskiy et al., 2020) architecture of YOLOv4. This modification reduces the model size and simplifies computation, resulting in a significant increase in processing speed. Since 2023, improved models derived from YOLOv8 have been actively developed for UAV detection applications. For example, a study by Yılmaz & Oruç (2024) improved YOLOv8's performance in low-light environments by incorporating data augmentation techniques for brightness and color adjustment. It also enhanced feature extraction layers to optimize detection accuracy and speed, making the model more effective for real-time drone monitoring in challenging lighting conditions. Similarly, a study by Zamri et al. (2024) integrated attention modules and contextual learning to optimize UAV detection in aerial surveillance, improving performance in high-resolution video feeds and complex detection scenarios, such as identifying UAVs at high altitudes or in cluttered environments.

Today's primary challenges in applying deep learning to anti-drone systems are model size and real-time processing speed. A practical system, with a compact size and the ability to be installed in locations with limited space, requires simple and efficient hardware. Therefore, the hardware resources of anti-drone systems must be optimized. Although YOLO-v8 has been significantly improved in terms of performance and object detection capabilities, one notable drawback when deploying it on embedded devices is its slower processing speed compared to previous versions, especially when applied to devices with limited computational resources. Therefore, while YOLO-v8 brings substantial upgrades in object detection performance, it still faces limitations in terms of speed when deployed on embedded devices with constrained hardware configurations. This paper proposes a solution that improves computational speed while reducing model size. The approach discussed in this paper aims to minimize the number of computations, enhancing processing speed without compromising the accuracy necessary for object detection. To achieve this objective, a re-parameterization method (Wang et al., 2023) is employed. According to the study, the re-parameterization method integrates computational components into a single inference step. This method transforms a model with a complex structure during training into a significantly simpler structure when deployed on hardware devices, thereby increasing processing speed. With the observations and evaluations mentioned above, a new YOLOXpress model has been developed to address the limitations of YOLO-v8 in drone detection tasks. This paper presents the re-parameterization method used to create YOLOXpress based on the YOLO-v8 framework.

Several layers in the YOLO-v8 architecture are replaced to create YOLOXpress with features better suited to the task's requirements. Through experimentation, the model has achieved notable results and demonstrated advantages over YOLO-v8, such as being smaller, deployable on more cost-effective and smaller devices, and ensuring greater usability, flexibility, and the ability to be deployed in various locations, terrains, and conditions. Furthermore, the model strikes a balance between speed and accuracy, processing faster than YOLO-v8 while maintaining an acceptable level of accuracy.

3. Method

3.1. YOLO-v8 Architecture

YOLO-v8 is an upgraded version of the YOLO model series, designed to enhance speed and accuracy in real-time object detection tasks. The architecture of YOLO-v8 comprises three main components: Backbone, Neck, and Head. The Backbone extracts key features from images through a CNN network. The Neck utilizes techniques such as the feature pyramid network (FPN) and path aggregation network (PAN) to combine multiple-level features, improving object detection capabilities, particularly for small objects. The Head predicts bounding boxes and labels, with Non-Maximum Suppression reducing prediction overlap.

- **Backbone:** This component is responsible for extracting features from the input image. The Backbone typically uses deep CNN to learn low-level and high-level features from the image.
- **Neck:** This section enhances the features extracted by the Backbone. The Neck usually includes layers such as FPN or PAN to combine and further enrich the features.
- **Head:** The final part of the network is responsible for generating the final predictions. The Head predicts the bounding boxes and the classes of objects in the image.

YOLO-v8's loss function consists of object existence prediction, object classification, and accurate bounding box localization. Significant improvements

such as Mosaic augmentation and other optimization techniques enhance YOLO-v8's learning and generalization abilities. However, for specific tasks, such as real-time UAV detection, YOLO-v8 reveals certain limitations, particularly in detecting small objects, failing to fully meet real-time requirements on embedded devices. This limitation arises from the computational inefficiencies of the Cross-Stage Feature Fusion (C2f) block, which remains cumbersome and suboptimal in terms of time efficiency. The architecture of C2f is shown in Fig. 1.

A layer with four characteristic parameters (w , h , C_m , C_{out} , K) was considered, where:

h denotes the height of the feature map,

w denotes the width of the feature map,

C_m represents the depth of the input feature map (with $C_{out} \geq C_m$),

C_{out} represents the depth of the output feature map,

and K represents the kernel size of the convolutional layer.

The computational cost of C2f was calculated by Eq. (1) (Wei et al., 2021):

$$\begin{aligned} Cost_{C2f} &= Cost_{Conv1} + n * Cost_{BottleNeck} + Cost_{Conv2} \\ &= h * w * C_{in} * C_{out} * K_1^2 + n * \left(h * w * \frac{C_{out} * C_{out}}{2 * 4} + \right. \\ &\quad \left. h * w * \frac{C_{out} * C_{out}}{2 * 4} \right) * \\ &\quad K_2^2 + h * w * (2 + n) * \frac{C_{out}}{2} * C_{out} * K_2^2 \end{aligned} \quad (1)$$

where n is a parameter of C2f, representing the number of times the BottleNeck block is repeated, with $n \geq 1$, K_1 and K_2 are the sizes of the two standard CBS layers, where $K_1 = 1$ and $K_2 = 3$, respectively).

Assuming $C_m = 3$, $C_{out} = 32$, $h = 640$, $w = 640$ (standard input image size), the computational cost is given by $Cost_{C2f} = 196758400 + n * 4718592000 \geq 6645350400$ (parameters).

Thus, even with the input image size and the number of iterations $n = 1$, the amount of computation required by C2f is still relatively large.

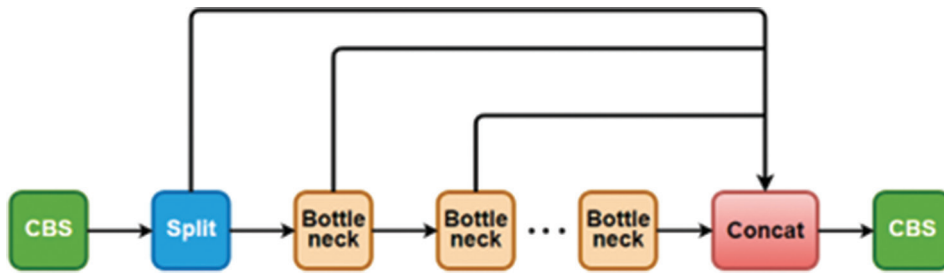


Fig. 1. Architecture of cross-stage feature fusion in YOLO-v8. Redrawn based on the original design by Zhai et al. (2023), with some adjustments to suit our study

Abbreviation: CBS: Convolution, batch normalization, and SiLU activation functions

3.2. YOLO-v8 Improvement for Real-time UAV Detection

For the real-time UAV detection task, the model must meet several requirements; it should be lightweight, suitable for embedded devices, and capable of accurately identifying small objects in real-time.

Based on the proposals and results from previous studies (Howard et al., 2017; Sandler et al., 2018; Terven et al., 2023), to reduce the model size, it is necessary to decrease the number of computations in the convolutional layers. Therefore, the C2f module was replaced with a simpler one.

Several challenges must be addressed regarding the small object detection problem: no object should be missed, and the object must be distinguishable from the background. Studies have proved that a 3×3 convolutional layer is effective in gathering local information while requiring lower computational costs compared to larger convolutional layers (Dosovitskiy, 2020; Szegedy et al., 2015; Szegedy et al., 2016). At present, the main approaches for small object detection rely on the Vision Transformer (ViT) architecture (Wu & Dong, 2023; Zhai et al., 2023; Zhu et al., 2021), (Dosovitskiy, 2020). This study showed that although ViT offers significant advantages in capturing global information, its cost is excessively high due to the use of the Self-Attention mechanism.

Therefore, this paper proposed an architecture block primarily based on 3×3 and 1×1 convolutional layers to retain the same level of information as the C2f block in YOLO-v8, but with reduced model size. The C2f layer in the Backbone was replaced with Re-parameterization Convolution (RepConv) to create a lighter Backbone while ensuring sufficient information is provided to the Neck. Subsequently, the C2f layer in the Neck was replaced with RepC3 to reduce computational complexity during feature extraction and aggregation while retaining essential object-related information.

3.2.1. Re-parameterization convolution

As presented above, we proposed an architectural block called RepConv in this section. The architecture of RepConv is shown in Fig. 2.

To compare RepConv with the C2f block, a layer with four characteristic parameters was considered (w, h, C_m, C_{out}, K).

The computational cost with RepConv was calculated by Eq. (2) (Wei et al., 2021):

$$Cost_{RepConv} = h * w * C_{in} * C_{out} * K_1^2 + h * w * C_{in} * C_{out} * K_2^2 \quad (2)$$

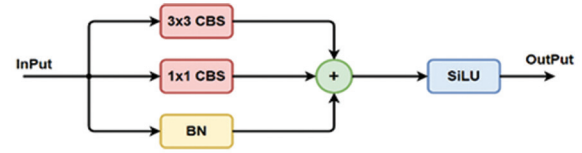


Fig. 2. Architecture Re-parameterization Convolution. Redrawn based on the original design by Zhai et al. (2023), with some adjustments to suit our study
Abbreviation: BN: Batch normalization; CBS: Convolution, batch normalization, and SiLU activation functions; SiLU: Sigmoid linear unit

$$Cost_{RepConv} = 10 * h * w * C_{in} * C_{out} \quad (3)$$

On the other hand, Eq. (IV) is as follows:

$$\begin{aligned} Cost_{C2f} &= h * w * C_{in} * C_{out} + 9 * n * h * w * C_{out} * \frac{C_{out}}{2} + \\ &9 * h * w * (2 + n) * \frac{C_{out}^2}{2} \geq h * w * C_{in} * C_{out} + 9 * n * h * \\ &w * C_{out} * \frac{C_{out}}{2} + 27 * h * w * \frac{C_{out}^2}{2} > 10 * h * w * C_{out}^2 \\ &> 10 * h * w * C_{in} * C_{out} \end{aligned} \quad (4)$$

Hence, $Cost_{RepConv} < Cost_{C2f}$

As presented in Section 3.2, the 3×3 convolutional layer captures local information in a smaller spatial area compared to larger convolutional layers, such as 5×5 or 7×7 . In the case of the drone detection task, the object typically occupies a very small portion of the entire image space. Therefore, using a convolutional layer that observes a large spatial area made it challenging to detect the object's features.

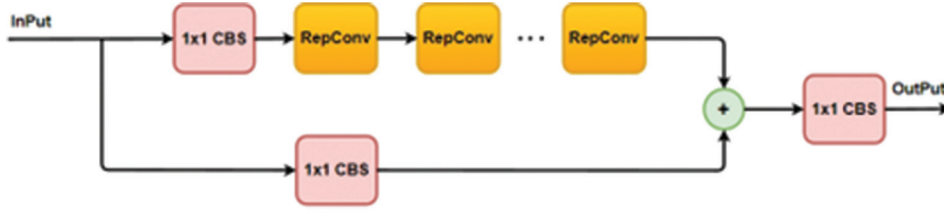
In addition, including a 1×1 branch to retain the original information of the object, which is then aggregated, helps highlight the key features extracted by the 3×3 layer.

This demonstrates that RepConv performs well as expected, with the target object standing out relative to its surrounding area. RepConv is capable of replacing C2f in the Backbone to meet real-time requirements.

3.2.2. RepC3

As demonstrated in Section 3.2 regarding the effectiveness of RepConv and its comparison with C2f, the BottleNeck blocks in C2f were proposed to be replaced with RepConv blocks to improve the speed of the model's Neck. The architecture of RepC3 is shown in Fig. 3.

Computational cost with RepC3w as calculated using the following equation:

**Fig. 3.** Architecture of RepC3

Abbreviation: CBS: Convolution, batch normalization, and SiLU activation functions;
RepConv: Re-parameterization Convolution

$$\begin{aligned}
 Cost_{RepC3} &= Cost_{Conv1} + n * Cost_{RepConv} + Cost_{Conv2} = \\
 h * w * C_{in} * C_{out} &+ n * 2.5 * h * w * \frac{C_{out}^2}{2} = h * w * \\
 C_{in} * C_{out} &+ (n * 2.5 + 0.5) * h * w * C_{out}^2 \quad (5)
 \end{aligned}$$

From Eqs. (4) and (5), we can see: $Cost_{RepC3} < Cost_{C2f}$

Two modules, RepConv and RepC3, were developed based on the aforementioned theoretical frameworks and reasoning. Both architectures achieved the dual objectives of minimizing computational overhead while preserving the informational fidelity of target objects. These two architectural blocks were then integrated to replace the Backbone of the original YOLO-v8 model, completing the YOLOXpress model.

3.3. Architecture YOLOXpress

Based on the theoretical foundation presented in Sections 3.1 and 3.2, the C2f blocks were replaced with RepConv and RepC3, respectively, as shown in Fig. 4.

The architecture in Fig. 4 primarily illustrates that the C2f blocks are the main components being replaced. We also modified the model's upscaling method by employing a convolutional layer instead of interpolation, which was used in the original architecture. Techniques such as anchor-free design, Feature Pyramid architecture, CIOU loss, DFL, and BCE remained unchanged.

4. Experiment and Results

This paper employed the UAV dataset, originally used to train the TIB-net model (Sun et al., 2020), for training the YOLOXpress model. The results obtained from training were then utilized to evaluate the model's performance, conduct ablation experiments, and compare them with other models.

4.1. Dataset

The TIB-Net UAV dataset contains 2,850 images, capturing various types of UAVs, including multi-rotor UAVs and fixed-wing UAVs. These images were

collected using a ground-based camera 500 m from the airborne UAVs, with a 1920×1080 pixels resolution. The scenery in the images includes low-altitude views, such as the sky, trees, and buildings, recorded at different times of the day and under various weather conditions. Analysis reveals that the UAVs occupy <1% of the area in each image.

4.2. Setup and Training Network

In this experiment, data from TIB-net were utilized to train the YOLO-v8 and YOLOXpress models. The experiment was performed on two pieces of hardware. The model training phase employed the Google Colab platform with a Tesla V100 GPU. After completing the training, the model was deployed on Jetson Orin Nano hardware. The NVIDIA Jetson Orin Nano 8GB Developer Kit was used for artificial intelligence processing applications, featuring a 6-core Arm® Cortex®-A78AE v8.2 64-bit CPU with 1.5MB L2 + 4MB L3 cache and an NVIDIA Ampere architecture GPU with 1024 CUDA cores and 32 Tensor cores. This configuration provides AI processing power up to 80 times greater than its predecessor, the Jetson Nano.

4.2.1. Loss function setting

The loss function of the YOLOXpress model is presented in Eq. (vi). It retains the same structure as the YOLOv8 loss functions, consisting of three components: rectangular box loss ($Loss_{box}$), distribution focal loss ($Loss_{dfl}$), and classification loss ($Loss_{cls}$).

$$Loss = a * Loss_{box} + b * Loss_{dfl} + c * Loss_{cls} \quad (6)$$

In this case, a , b , and c each represent the weighted proportion of the corresponding loss function in the overall loss function. In this experiment, the three weights were set as $a = 7.5$, $b = 1.5$, and $c = 0.5$.

4.2.2. Network training

Before training the network, the data directory was prepared in the YOLO-v8 format, which includes two folders: "images" and "labels." A batch size of 16

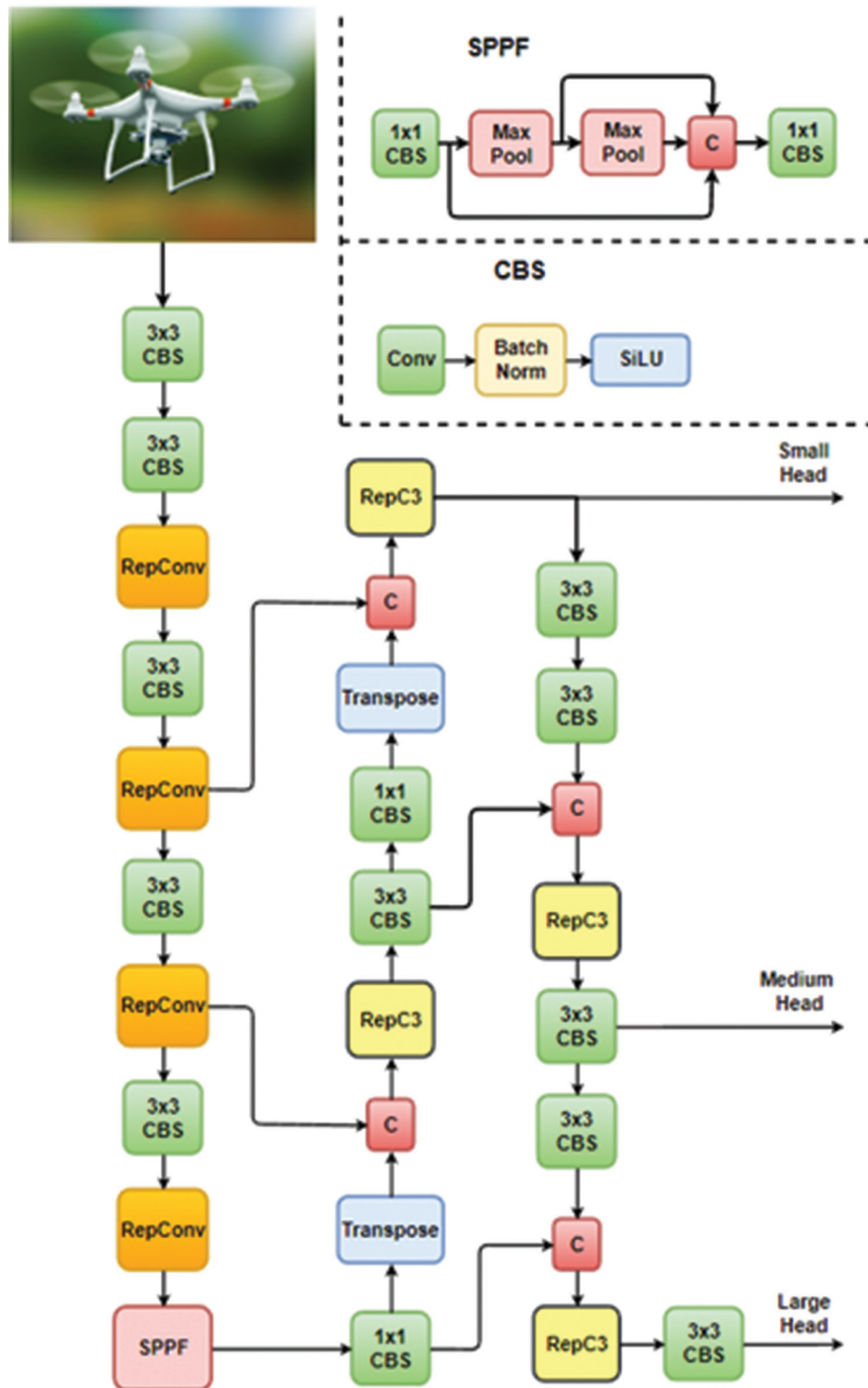


Fig. 4. Architecture of the YOLOXpress model

Abbreviation: CBS: Convolution, batch normalization, and SiLU activation functions;
 RepConv: Re-parameterization Convolution; SiLU: Sigmoid linear unit; SPPF: Spatial Pyramid
 Pooling Fusion

was chosen, and the model was trained for 250 epochs using an initial learning rate of 0.01. Table 1 describes

the configuration parameters used during network training.

Table 1. Network training configuration

Parameter	Values
Epochs	250
Warm up epochs	10
Batch size	16
Image size	640×640
Initial learning rate	0.01
Final learning rate	0.01

4.3. Evaluation Metrics

To evaluate the model's quality, parameters, such as Precision (P), Recall (R), Average Precision (AP), mAP, the number of parameters, model size, and FPS were used.

The precision rate and recall were calculated using the following Eqs. (7) and (8), respectively:

$$P = \frac{TP}{(TP + FP)} \times 100\% \quad (7)$$

$$R = \frac{TP}{(TP + FN)} \times 100\% \quad (8)$$

True Positives (TP) represent the number of accurately detected objects, False Positives (FP) represent the number of non-target objects incorrectly detected as targets and False Negatives (FN) represent the number of targets not detected.

The AP and mAP were calculated using the following Eqs. (9) and (10), respectively:

$$AP = \int_0^1 p(r) d(r) \quad (9)$$

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (10)$$

where $p(r)$ is precision at recall r , and N denotes the total number of classes. In this paper, $N = 1$ corresponds to the task of UAV detection.

4.4. Ablation Experiments

In this section, the impact of each module replaced in the structure of the Model is clarified. Based on the TIB-net UAV dataset, replacement experiments were conducted on the original YOLO-v8 model, where modules in the Backbone and Neck were sequentially replaced according to a predetermined order. Four models were proposed for examination to analyze each module's impact. Model (a) is the standard YOLO-v8, Model (b) is the enhanced version with the RepC3 module replaced in the Neck, Model

(c) is the enhanced version with the RepConv module replaced in the Backbone, and Model (d) is the enhanced version with both the RepC3 and RepConv modules replaced. The changes in these models were evaluated through the quantitative assessment of the parameters used to measure model performance, as presented in Table 2.

From the results in Table 2, the following observations can be made:

The model with the C2f module in the Backbone replaced by the RepConv module was proven effective in reducing the model size while maintaining performance. This is reflected in the data shown in Table 2, where the model sizes of (c) and (d) decreased by 11.17 MB and 12.44 MB, respectively, compared to model (a). Meanwhile, the P, R, and mAP parameters changed only slightly compared to model (a), with mAP decreasing by 0.244% for model (c) and 6.136% for model (d), and the changes in P and R being insignificant.

The reduction in the number of parameters, model size, and the number of computations in model (b) compared to model (a) demonstrates the effectiveness of the RepC3 module when replacing the C2f module in the Neck of model (a). The number of parameters decreased by 44.3%, the model size decreased by 80.8%, and GFLOPs decreased by 43.9%, while the P, R, and mAP parameters changed slightly compared to model (a). These findings prove that the RepC3 module effectively reduces computational load and model size while maintaining model performance.

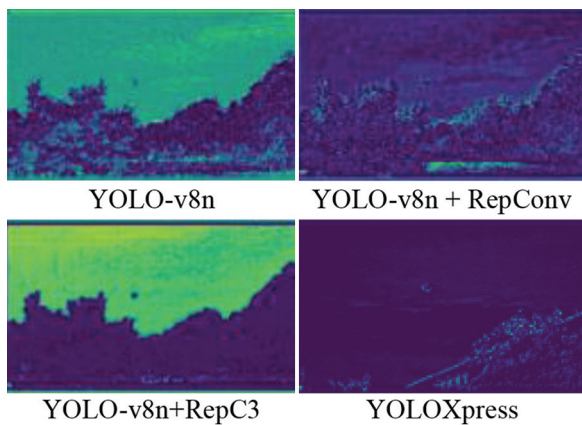
The YOLOXpress model, which incorporated both the RepC3 and RepConv modules described in Section 3, showed a clear improvement in model size, GFLOPs, and parameters compared to the YOLO-v8n model, as indicated in Table 2. The performance metrics P, R, and mAP decreased slightly compared to model (a), but the differences were negligible. The difference between the models is shown in the feature map images in Fig. 5.

Four models were tested on Jetson Orin hardware, with measured results compared in Fig. 6. The findings demonstrate that the YOLOXpress model (modified by replacing the C2f module with the RepC3 and RepConv modules) achieved a balance between processing speed and accuracy. Specifically, compared to the original YOLOv8 model, YOLOXpress attained 93.99% accuracy while delivering a 12.25% increase in FPS and a 69.89% reduction in model size. These improvements made YOLOXpress highly suitable for deployment on low-configuration, compact-sized devices, which are ideal for applications demanding efficient object detection without compromising performance, even in hardware-constrained environments. The results underscore YOLOXpress's

Table 2. Results of the various ablation experiments

Component	YOLO-v8 (a)	YOLO-v8 (RepC3) (b)	YOLO-v8 (RepConv) (c)	YOLOXpress (RepC3 and RepConv) (d)
P	99.524	97.672	98.236	96.229
R	97.783	98.123	97.697	97.193
mAP	96.266	95.408	96.042	90.13
Parameter (million)	3.011	1.678	3.373	2.699
Model size	17.8	3.41	6.63	5.36
GFLOPs	8.2	4.6	9.3	7.6
Means of ACC	0.633	0.505	0.443	0.595
FPS	29.168	32.163	32.823	32.741

Abbreviations: ACC: Accuracy; AP: Average Precision; FPS: Frames Per Second; GFLOPs: Gigafllops mAP: Mean Average Precision; P: Precision; R: Recall; RepConv: Re-parameterization Convolution; YOLO: You Only Look Once.

**Fig. 5.** Feature map extraction images of the test models and the YOLO-v8n model

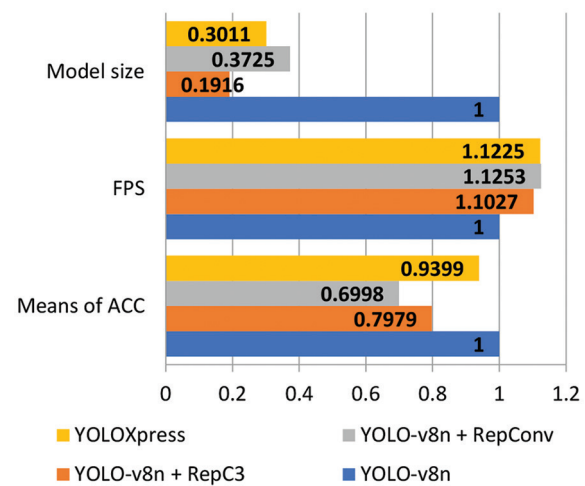
Abbreviations: RepConv: Re-parameterization Convolution; YOLO: You Only Look Once

practical potential as a resource-efficient solution for real-world scenarios.

4.5. Comparative Experiment

To clarify the advantages of the YOLOXpress model, we compared it with the fastest models in the YOLO series, which are currently widely used on embedded devices (YOLOv8n, YOLOv6-s3.0 [Li et al., 2022], and YOLOv5n). The models were trained using a dataset we prepared, consisting of 8,022 images of various UAV with a resolution of 640×640 pixels. The results are presented in Table 3. The models selected for comparison are all official versions.

According to Table 3, it can be observed that: YOLOXpress achieved an FPS of 23.44, the highest among the models compared, demonstrating exceptional processing capability. This speed allows YOLOXpress to operate efficiently on embedded devices with limited computational resources.

**Fig. 6.** Comparison chart between test models and the YOLO-v8n model (based on means of acc [accuracy], Frames Per Second [FPS], and model size)

Abbreviation: RepConv: Re-parameterization Convolution

Although YOLOv8n had the highest accuracy, its processing speed was only 18.02 FPS, significantly lower than that of YOLOXpress. This may affect applications that require real-time responsiveness. Other models, such as YOLOv5 (FPS = 20.05) and YOLOv6 (FPS = 19.52), also exhibited fast processing speeds but still fell short compared to YOLOXpress in scenarios demanding stringent real-time performance.

Although YOLOXpress did not achieve the highest accuracy (0.6243 compared to 0.6755 for YOLOv8n), it still maintained a strong performance in object detection with an mAP of 0.1636, which was close to YOLOv8n (mAP = 0.1750). This finding indicates that YOLOXpress strikes a good balance between accuracy and processing speed, which is crucial for applications that require both rapid processing and high reliability. While YOLOv6 and YOLOv5 demonstrated smaller model sizes and faster

Table 3. Comparison of experimental results

Parameter	YOLO-v8n	YOLO-v6	YOLO-v5	YOLO Xpress
Means of ACC	0.676	0.572	0.555	0.624
FPS	18.02	19.52	20.05	23.44
Model size (MB)				
Pytorch	5.9	49	5.0	5.1
Torchscript	11.9	16.5	10.1	10.1
Onnx	11.7	16.3	9.8	10.0
mAP (50-95)				
Pytorch	0.1750	0.1064	0.1588	0.1636
Torchscript	0.1732	0.1066	0.1584	0.1632
Onnx	0.1732	0.1066	0.1584	0.1632
Process time per image (ms/im)				
Pytorch	8.5	7.38	10.67	5.92
Torchscript	7.11	7.89	7.63	5.71
Onnx	137.24	178.43	146.65	119.82

Abbreviations: ACC: Accuracy; FPS: Frames per second.

processing speeds, they exhibited lower accuracy and mAP, which limits their effectiveness in scenarios that demand precise object detection.

With a model size of 5.1 MB, YOLOXpress delivered high performance and was also easily deployable on devices with limited memory, comparable to YOLOv5 (5.0 MB). This feature is crucial when deploying models on embedded devices or systems with constrained resources. While YOLOv8n and YOLOv6 showed larger model sizes (16 MB and 49 MB, respectively), the increased model size may require more powerful hardware and impact the ability to deploy on devices with limited memory and computational resources. YOLOXpress achieved the lowest processing time at 5.92 ms on the Pytorch platform, demonstrating its fast processing capability, which is well-suited for real-time detection applications.

Fig. 7 illustrates the real-world testing results in three scenarios: (i) Long-range detection under foggy weather conditions and complex background objects, (ii) long-range detection with an overcast sky, and (iii) detection in an environment with numerous complex objects. From the three provided images, we were able to assess the performance of the YOLOXpress model under these scenarios as follows:

Fig. 7A (long range, small target size, foggy conditions): The model successfully detected the drone despite low lighting and fog, which reduced contrast. The confidence score was 0.53, indicating that the model detected the drone with relatively low certainty due to the challenging conditions and small target size. These results demonstrate that the model can still perform acceptably under unfavorable conditions.

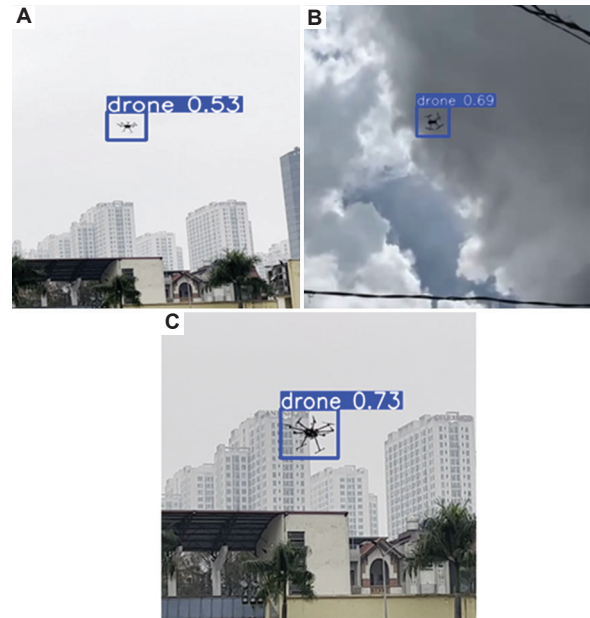


Fig. 7. Experimental results of the YOLOXpress model on the Jetson Orin embedded system under real-world conditions: (A) long-range detection with a small target size and foggy weather, (B) long-range detection with an overcast sky, and (C) detection in an environment with numerous complex objects

Fig. 7B (long-range, overcast sky): In the presence of thick clouds, the model detected the drone with higher confidence, achieving a score of 0.69. The detection performance improved compared to the first image, possibly due to the enhanced contrast between the drone and the overcast sky, making the target easier to identify. These findings suggest the model performs well even in complex sky conditions with fewer interfering objects.

Fig. 7C (environment with many interfering objects): The model achieved the highest confidence score of 0.73, successfully detecting the drone despite the presence of numerous background objects (buildings and trees). This finding demonstrates the model's robustness in handling complex environments with multiple potential sources of interference, particularly at close range. The successful detection in this scenario highlights YOLOXpress's ability to handle visually complex scenes.

In conclusion, the YOLOXpress model exhibited strong performance across various conditions, from unfavorable weather (fog) and overcast skies to environments with significant visual clutter. However, low lighting and small target sizes still impacted the model's confidence.

5. Conclusion

The YOLOXpress model proposed in this paper addresses the limitations of the YOLO-v8n model when

deployed on low-end hardware devices, specifically for detecting small objects, particularly in UAV detection and alert systems. By prioritizing a small model size, fast processing speed, and maintaining an acceptable level of accuracy, YOLOXpress can be more easily deployed in resource-constrained environments. Specifically, the C2f module in the Backbone and the C2f module in the Neck can be replaced with the RepConv and RepC3 modules to reduce the number of computations while preserving the ability to extract object features. This modification reduces the model size without significantly affecting accuracy. Replacement and comparative experiments conducted on the TIB-Net dataset have provided specific metrics. Compared to the original model, the YOLOXpress model improved FPS and Model size by 12.25% and 69.96%, respectively. The parameters and computations were reduced by 10.36% and 7.32%, respectively.

In summary, the changes made to YOLOXpress compared to YOLO-v8 demonstrate that the model is suitable for deployment on low-end devices while ensuring real-time UAV detection. However, replacing the C2f module with the RepConv and RepC3 modules has resulted in a reduced accuracy compared to the original model. The average accuracy of the YOLOXpress model decreased by 6% compared to the original model. This reduction is minimal, and the accuracy remains within an acceptable range. Experiments on our custom-built dataset indicate that the recall rate decreases when more complex objects are in the background. Future work will improve accuracy and detection capability in complex background conditions.

References

- Al-Iqubaydhi, N., Alenezi, A., Alanazi, T., Senyor, A., Alanezi, N., Alotaibi, B., Alotaibi, M., Razaque, A., & Hariri, S. (2024). Deep learning for unmanned aerial vehicles detection: A review. *Computer Science Review*, 51, 100614.
- Alsanad, H.R., Sadik, A.Z., Ucan, O.N., Ilyas, M., & Bayat, O. (2022). YOLO-V3 based real-time drone detection algorithm. *Multimedia Tools and Applications*, 81(18), 26185–26198.
- Bochkovskiy, A., Wang, C.Y., & Liao, H.Y.M. (2020). *Yolov4: Optimal Speed and Accuracy of Object Detection*. [arXiv Preprint] arXiv:2004.10934.
- Dosovitskiy, A. (2020). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. [arXiv Preprint] arXiv:2010.11929.
- Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. [arXiv Preprint] arXiv:1704.04861, p. 126.
- Kim, J.H., Kim, N., & Won, C.S. (2023). High-Speed Drone Detection Based on Yolo-v8. In: *ICASSP 2023–2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.
- Lee, Y., Hwang, J.W., Lee, S., Bae, Y., & Park, J. (2019). An Energy and GPU-Computation Efficient Backbone Network for Real-Time Object Detection. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.
- Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W., Li, Y., Zhang, B., Liang, Y., Zhou, L., Xu, X., Chu, X., Wei, X., & Wei, X. (2022). YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications [arxiv Preprint].
- Liu, Y., Liu, D., Wang, B., & Chen, B. (2022). Mob-YOLO: A Lightweight UAV Object Detection Method. In: *2022 International Conference on Sensing, Measurement and Data Analytics in the era of Artificial Intelligence (ICSMD)*.
- Long, X., Deng, K., Wang, G., Zhang, Y., Dang, Q., Gao, Y., Shen, H., Ren, J., Han, S., & Ding, E. (2020). *PP-YOLO: An Effective and Efficient Implementation of Object Detector*. [arXiv Preprint] arXiv:2007.12099.
- Lv, Y., Ai, Z., Chen, M., Gong, X., Wang, Y., & Lu, Z. (2022). High-resolution drone detection based on background difference and SAG-Yolov5s. *Sensors (Basel)*, 22(15), 5825. <https://doi.org/10.3390/s22155825>
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.C. (2018). Mobilenetv2: Inverted Residuals and linear Bottlenecks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Sun, H., Yang, J., Shen, J., Liang, D., Ning-Zhong, L., & Zhou, H. (2020). TIB-Net: Drone detection network with tiny iterative backbone. *IEEE Access*, 8, 130697–130707. <https://doi.org/10.1109/ACCESS.2020.3009518>
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going Deeper with Convolutions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Terven, J., Córdova-Esparza, D.M., & Romero-González, J.A. (2023). A comprehensive

review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine Learning and Knowledge Extraction*, 5(4), 1680–1716.

Wei, T., Tian, Y., & Chen, C.W. (2021). Rethinking Convolution: Towards an Optimal Efficiency. In: *Under Review as a Conference Paper at ICLR*.

Wu, T., & Dong, Y. (2023). YOLO-SE: Improved YOLOv8 for remote sensing object detection and recognition. *Applied Sciences*, 13(24), 12977.

Yılmaz, H.B., & Oruç, F. (2024). Drone detection performance evaluation via real experiments with additional synthetic darkness. *Gazi University Journal of Science Part A: Engineering and Innovation*, 11(3), 546–562.
<https://doi.org/10.54287/gujisa.1526979>

Zamri, F.N.M., Gunawan, T.S., Yusoff, S.H., Alzahrani, A.A., Bramantoro, A., & Kartiwi, M. (2024). Enhanced small drone detection using optimized YOLOv8 with attention mechanisms. *IEEE Access*, 12, 90629–90643.
<https://doi.org/10.1109/ACCESS.2024.3420730>

Zhai, X., Huang, Z., Li, T., Liu, H., & Wang, S. (2023). YOLO-Drone: An optimized YOLOv8 network for tiny UAV object detection. *Electronics*, 12(17), 3664.
<https://doi.org/10.3390/electronics12173664>

Zhu, X., Lyu, S., Wang, X., & Zhao, Q. (2021). TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios. In: *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*.

AUTHOR BIOGRAPHIES



Nguyen Tien Tai is currently a Master's student at Le Quy Don Technical University. He earned his Bachelor's degree in Electrical and Electronics Engineering from Le Quy Don Technical University, Hanoi, Vietnam, in 2019.



Bui Duc Thang is currently a senior student at Le Quy Don Technical University.



Nguyen Ngoc Hung is currently a lecturer at Le Quy Don Technical University. He graduated with a Bachelor's degree in Electrical and Electronics Engineering from Le Quy Don University in 2010 and earned his Ph.D. from the same institution in 2023. His research interests include image processing for thermal camera, digital signal processing, and intelligent controller designing.

A blockchain-based solution to combating identity crime and credit card application fraud using data mining algorithms

Amol Jagdish Shakadwipi^{1*}, Dinesh Chandra Jain², S. Nagini³

¹Research Scholar, Department of Computer Science and Engineering, Oriental University, Indore, India

²Research Supervisor, Department of Computer Science and Engineering, Oriental University, Indore, India

³Research Co-Supervisor, Department of Computer Science and Engineering, Oriental University, Indore, India

*Corresponding author E-mail: amolshakadwipi@gmail.com

(Received 28 November 2024; Final version received 23 January 2025; Accepted 03 February 2025)

Abstract

Fraud, specifically identity theft and credit card fraud, poses significant threats not only to financial institutions but also to their users. In response to this growing problem, we present an innovative approach that integrates self-sovereign identity management based on blockchain and complex data analysis. Our comprehensive solution is designed to revolutionize identity verification in credit card application processes by significantly enhancing security and reducing vulnerability to identity fraud. The system that will be developed from our solution will help users obtain self-sovereign identity credentials through blockchain technology or distributed ledger technology, granting them full control over their personal data. This approach has been proven to drastically reduce the likelihood of identity theft, and it does not require centralization of data. Besides, the use of blockchain technology ensures more credible records of identification, as they are transparent and immutable. At P&L, we combine smart data mining with blockchain-based identity solutions as our primary strategy. These algorithms detect patterns and anomalies related to identity theft in massive datasets. The technology can quickly flag suspicious activity and verify identity claims in real-time by continuously comparing recent user activity with historical data.

Keywords: Fraudulent, Credit Card Applications, Suspicious Activities, Vulnerability, Blockchain, Identity Verification, Identity Management, Identity Theft

1. Introduction

This paper focuses on the risks associated with credit card fraud in today's digital world, where much of our activities are conducted online. Such systems need to be fundamentally redesigned, as it has become easy to obtain personal information, and the methods of identity theft are constantly changing. This work presents a novel approach for strengthening the security of credit card applications using data mining and the blockchain. In particular, we offer a blockchain-enabled self-sovereign identity management and data mining solution that enhances both the security of credit card applications and users' control over their personal data. The traditional credit card application model is vulnerable to security risks due to its reliance on centralized identity management techniques. To address the identified problem, our

proposed implementation leverages blockchain to create a more secure, distributed database for the storage and handling of identity information. Given that self-sovereign identity management allows individuals to have full control and ownership of their personal data, this solution minimizes the risks of data leakage and unauthorized access. In addition, the system can actively identify patterns typical of fraud or identity theft, thanks to the inclusion of data mining instruments. The intelligent security feature of our system analyzes real-time transaction behaviors and past records to detect suspicious actions throughout the entire credit card application procedure.

The combination of blockchain and data mining improves security while simultaneously shortening the overall application process, thereby increasing efficiency.

In this paper, we analyze the key features of our blockchain-enabled self-sovereign identity management and data mining solution, discussing how it works, its constituent elements, and how this solution can prevent credit card application fraud. As demonstrated in the following sections, this approach not only protects personal data but also represents a significant step toward a more secure and user-oriented finance industry. Research by Doe et al. (2022) demonstrated the application of blockchain technology for self-identity management in financial services such as credit card transactions. This paper reviews and incorporates self-sovereign identity frameworks into credit card application procedures, offering a user-centric approach that empowers individuals to control their data while reducing the risk of identity fraud. Blockchain's immutable records ensure secure identity verification, adding a layer of trust and reliability.

2. Key Features

- (i) Self-sovereign identity management: Empowering users with control over their personal information to improve security and privacy
- (ii) Blockchain-based verification: Reducing the risk of fraud by offering auditable and unchangeable identity records
- (iii) Real-time data mining: Identifying patterns of identity-related fraud by analyzing past data and user activity using sophisticated algorithms
- (iv) Secure credit card application process: Improving the verification procedure to reduce the possibility of unauthorized credit card issuance

- (v) User-centric approach: Providing individuals with the tools they need to safely and effectively manage their identities.

By combining blockchain technology with data mining algorithms, our application offers a comprehensive solution to combat credit card application identity fraud. It creates a robust, user-centric, and secure identity verification process, ultimately protecting both financial institutions and individuals from the growing threat of identity-related fraud. This paper acknowledges the declining efficacy of traditional identity management and fraud detection methods due to the increasingly sophisticated tactics used by fraudsters. It emphasizes the urgent need to shift toward innovative solutions that leverage blockchain technology and data-mining algorithms for enhanced fraud prevention and security.

3. Literature Review

One current problem affecting the financial sector is the rising instances of credit card application fraud. Traditional approaches to identity management and fraud identification are no longer effective, as fraudsters have become more innovative. In response to credit card application fraud, this paper provides a literature review of the latest research and advancements on data mining and self-sovereign identity solutions on a blockchain.

Ownership of personal data and self-reliant identity: Individual sovereignty over personal information is emphasized by the idea of self-sovereign

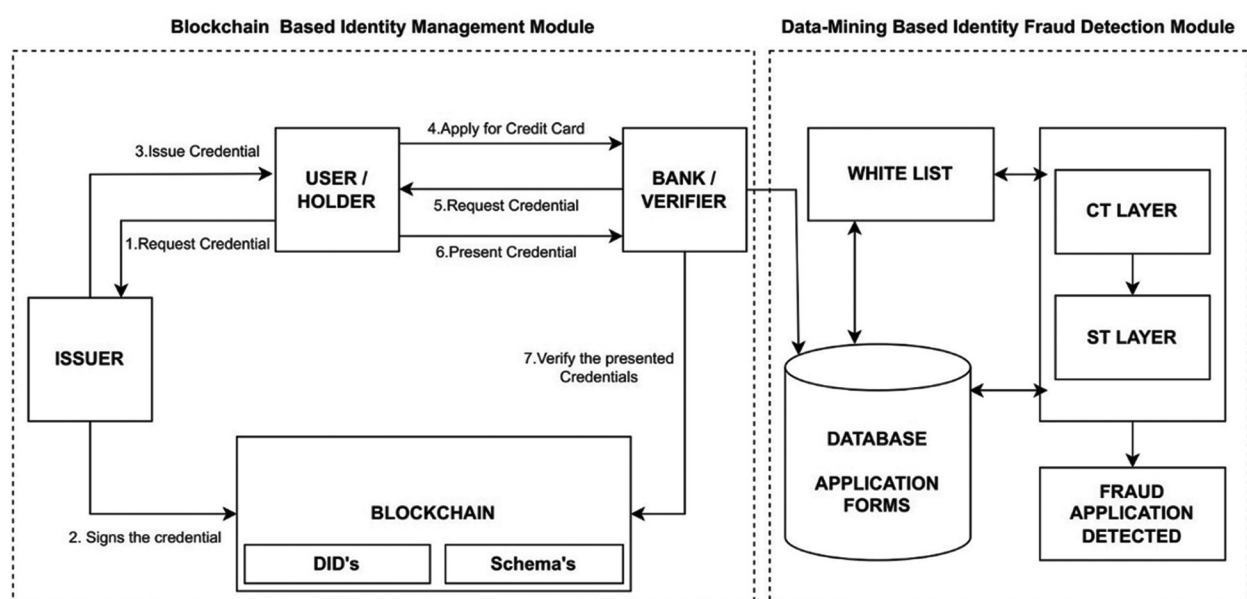


Fig. 1. System architecture of the blockchain-based fraud prevention and detection system for credit card application with self-sovereign identity management

Abbreviations: CT: Communal tracing; DID: Decentralized identifiers; ST: Spike tracing

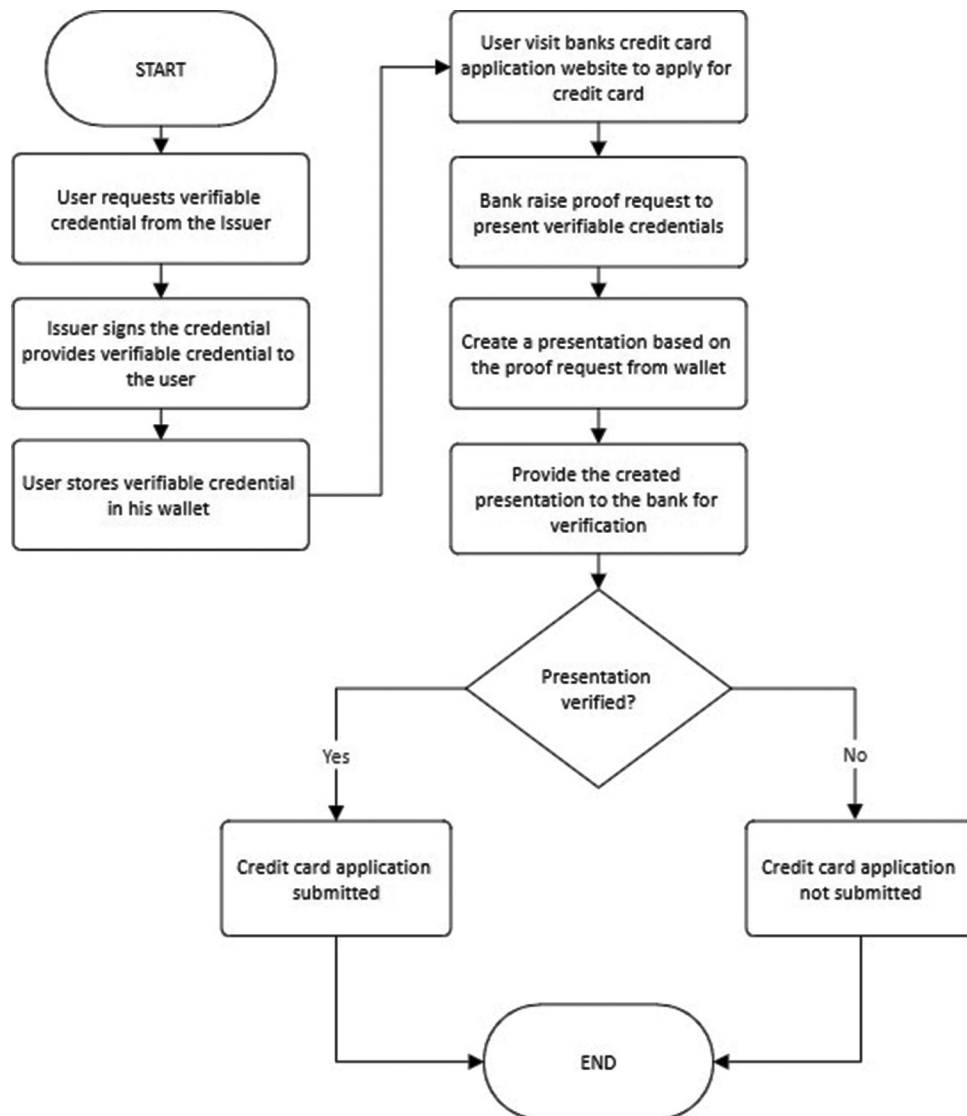


Fig. 2. Flow diagram of the blockchain identity module

identity. The proposed approach to empower people in the credit card application process can incorporate the standards and procedures for accomplishing this, as discussed in Alex & Reed (2021). Fraud detection, through data mining, is greatly enhanced by the utilization of machine learning algorithms specifically designed for this purpose. A study conducted by Bolton & Hand (2001) discussed data mining methods employed to detect credit card activities in real-time when integrated into the recommended system. The combination of blockchain and data mining is crucial in research that brings together the realms of data mining and blockchain technologies, as highlighted in a study by Zohrevand et al. (2020).

This integration has the potential to enhance the precision and security in identifying instances of credit card application fraud. Real-life examples and practical applications play a role in understanding the viability of the suggested solution. Scholarly

works, such as Smith et al. (2022), demonstrate how self-identity management using technology can be utilized in financial sectors, such as credit card processing. Blockchain technology plays a role in identity management due to its reputation for immutability and decentralization, as observed in studies such as Kshetri & Voas (2020), which highlights the potential benefits of ensuring secure and self-sovereign identity management while empowering individuals and mitigating fraud risks. Self-sovereign identity systems are tools that empower people by giving them control over their information, as discussed in the research presented in a study by Stallings et al. (2021).

The current study sheds light on different frameworks that can be incorporated into credit card application procedures, aiming to enhance user-focused identity verification and mitigate the risks of fraud. Fraud detection using data mining techniques:

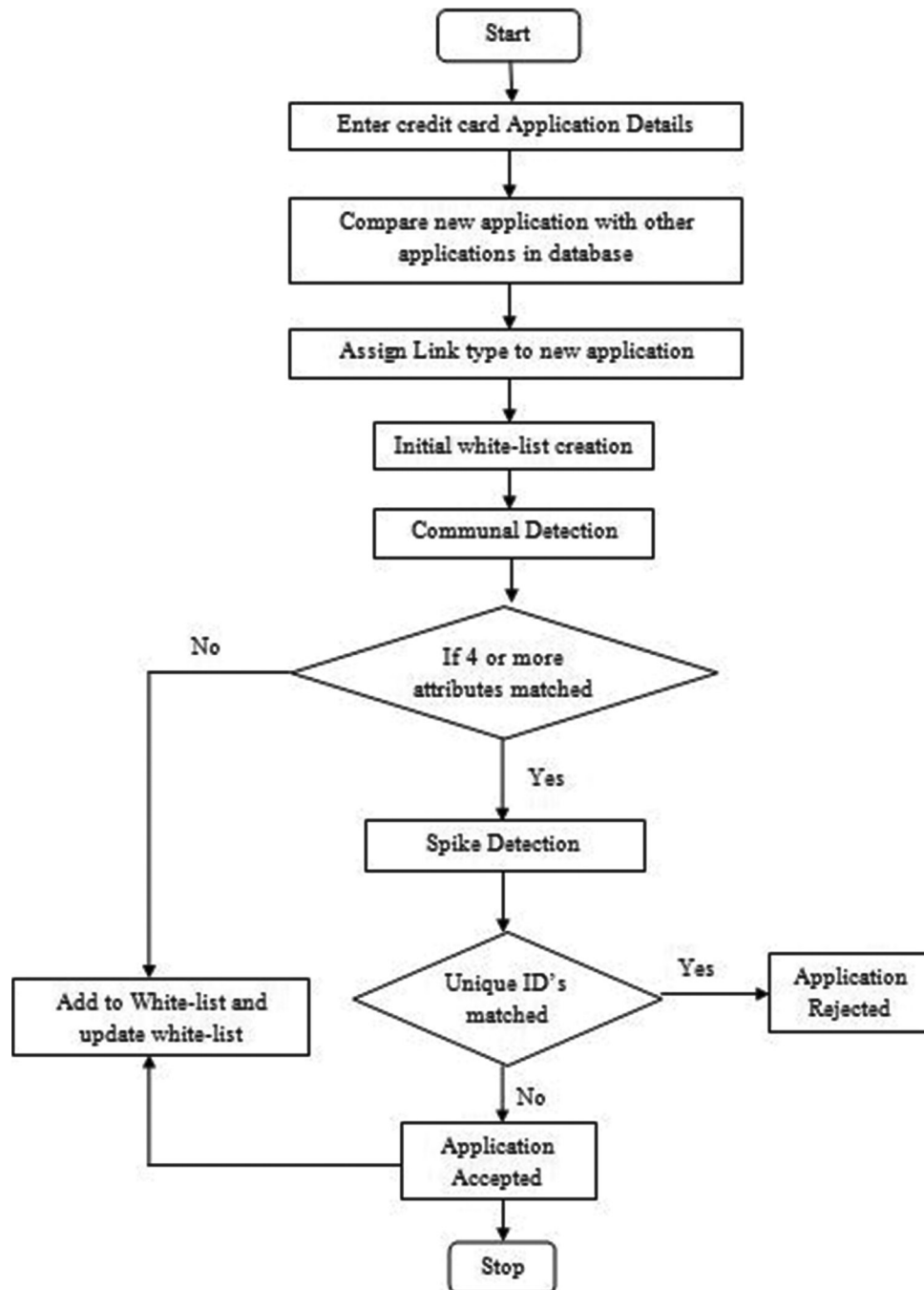


Fig. 3. Flow chart of data-mining-based identity fraud detection module.

Abbreviation: ID: Identification

The utilization of data mining techniques and machine learning algorithms has proven successful in detecting fraud within the banking sector, as emphasized in a study by Jha et al. (2017).

As Phua et al. (2010) suggested, a flexible spike detection technique adapts such a framework and improves data stream mining. For applications that need timely results, the authors focused on designing a method that can identify changes or outlying samples in a data stream. Their approach addressed problems specific to dynamic environments, where data features cause shifts in trends, making it difficult for

traditional detection techniques to work. The proposed approach improved overall system robustness and maneuverability and included adaptable procedures to better define disturbances in data streams. This research is highly relevant to fields that require instant detection of anomalous patterns in data, such as network security and financial analysis, helping to forestall possible threats or losses. As static models are less useful in a dynamic data environment, their study emphasized the need for flexibility in spike finding [17].

The incorporation of data mining enhances the recognition of behaviors in credit card application

Table 1. System architecture for the blockchain-based module.

S. No.	Issuer	Type: ["Verifiable credential"]	Proof type	Proof date	Issue date	Final result
1	ITD	"PanCard"	Ed25579Signature2018	2023-10-02T06:38:53Z	2023-07-09T05:14:14.000Z	Invalid signature
2	ITD	"PanCard"	Ed25579Signature2018	2023-10-02T06:38:53Z	2023-07-09T05:14:14.000Z	Valid signature
3	UIDAI	"AadharCard"	Ed25579Signature2018	2023-10-02T06:38:53Z	2023-07-09T05:14:14.000Z	Valid signature
4	UIDAI	"AadharCard"	Ed25579Signature2018	2023-10-02T06:38:53Z	2023-07-09T05:14:14.000Z	Valid signature
5	UIDAI	"AadharCard"	Ed25579Signature2018	2023-10-02T06:38:53Z	2023-07-09T05:14:14.000Z	Invalid signature
6	UIDAI	"AadharCard"	Ed25579Signature2018	2023-10-02T06:38:53Z	2023-07-09T05:14:14.000Z	Valid signature
7	ITD	"PanCard"	Ed25579Signature2018	2023-10-02T06:38:53Z	2023-07-09T05:14:14.000Z	Valid signature

procedures. The partnership between blockchain and data mining is vital, as research demonstrates how integrating these technologies can improve security protocols. A study by Wang et al. (2019) underscores the capability of boosting accuracy in credit card fraud detection while addressing privacy issues. Real-world examples and case studies provide insights into how the suggested solutions can be implemented in practice. To address the need for immediate detection of fraud, the framework integrated advanced data-mining techniques, such as communal and spike tracing (ST). These methodologies were designed to identify anomalies and outliers in real-time data streams, ensuring swift action in applications demanding instant fraud detection and prevention.

4. Methodology and Working

4.1. Module 1 (blockchain-based identity management module) working

- (i) An identity owner, also known as a user or holder, requests verified credentials from reliable issuers such as ITD or UIDAI
- (ii) The issuer generates a verifiable credential containing specific user information, and this credential is securely stored on a blockchain. To ensure its authenticity, the issuer applies cryptographic or digital signatures
- (iii) The Dock blockchain serves as the repository for decentralized identifiers associated with issuers, holders, and verifiable credentials
- (iv) When individuals apply for a credit card, verifiers – typically banks in this context – request the presentation of verifiable credentials as part of the know-your-customer process
- (v) Holders provide their verifiable credentials to the

Table 2. Link types and weights in the whitelist.

Link-type	Count	Weight
0000100101	1	0.11
0000010101	1	0.22
0000010101	1	0.33
0000010101	1	0.44
0000000101	1	0.55
0000000101	1	0.66
0000000101	1	0.77
000011010	1	0.88
0011010100	1	1

verifiers, which, in this case, are banks

- (vi) Verifiers, such as the bank, use the Dock blockchain to authenticate the presented credentials and ensure their validity
- (vii) Only if the presented verifiable credentials are verified as valid will the verifiers (in this case, the bank) proceed with the credit card application process. The bank will only process the application if the provided document is valid.

4.2. Module 2 (data-mining-based identity fraud detection module) working

Communal tracing (CT) and ST are two separate layers that comprise the system's novel technique. These layers are designed to improve credit card transaction security throughout the application process by efficiently identifying fraudulent activity from a variety of angles.

4.2.1. Communal tracking

Using a whitelist-driven approach that makes use of a predetermined set of characteristics, the CT layer lowers suspicion ratings and finds real social relationships. This method helps guard against attempts to manipulate artificial social connections. To find connections within the community and lower the linkage scores, the CT algorithm evaluates each link with the help of the whitelist. Its requirement of at least three similar values in the dataset for identification, however, could be a disadvantage because it may fail to identify circumstances where malicious entities replicate valuable values. For the purpose of scoring an existing application, CT also considers attribute weights and compares them to other applications within a moving window. The variability of a random parameter, which quantifies both effectiveness and efficiency at each mini-discrete data stream, produces a new whitelist from the current links.

The CT algorithm uses the following iterative steps:

- (i) To find connections, compare each application value to earlier application values
- (ii) Examine each application’s link in light of the whitelist so that communal relationships may be found and their link scores can be lowered
- (iii) Use link information and the scores of prior applications to calculate the current application’s score, then add the scores of those applications to the current application’s score
- (iv) Create a new whitelist based on the existing mini-discrete stream links by adjusting the value of a randomly chosen parameter to strike a compromise between efficacy and efficiency.

4.2.2. Combining CT and ST

The objective of this research is to integrate the following layer into the existing CT layer to form a hybrid ST layer with increased complexity and flexibility in detecting fraudulent actions in credit card application processes. ST uses an innovative strategy to address this challenge by dedicating attention to data spikes to raise the indices of suspicion, while sharing CT focuses on recognizing connections between individuals in a community. This strategy will help prevent fraudsters from obtaining simple characteristics required to calculate the ST score. Using an attribute-centered approach, ST erratically selects attributes that are neither too crowded nor too scarce. To optimize the algorithm, it also includes the formulas for calculating the ST suspicion score and routinely eliminates surplus attributes.

Table 3. Processed applications.

1	Thomas	Maranoa street	Marayong	nsw	423104	7573884029	123456789107	GSFMZ1006G	27/12/1932	Accepted
	2	Anurag	Sangale	qld	422154	8806323532	123456789103	ABCDE1234F	22/12/1990	Rejected

Table 4. Comparison between the blockchain-based system and the traditional identity crime detection system.

Metrics	Blockchain-enabled solution	Traditional resilient identity detection
Fraud detection accuracy (%)	95	82
Efficiency improvement (processing time)	35% reduction	Negligible change
User satisfaction (scale: 1 – 5)	4.6	3.1
Data privacy and control (scale: 1 – 5)	4.8	2.5
Security effectiveness (scale: 1 – 5)	4.9	3.4
Cost-benefit analysis (savings)	\$1.2 million/year	\$600,000/year
User adoption rate (%)	87%	55%
Regulatory compliance (scale: 1 – 5)	4.7	3.2
Scalability (high/medium/low)	High	Medium
Future recommendations	- Enhance data privacy features	- Explore blockchain for broader
	- Extend data mining capabilities	- Security applications

4.2.3. ST

The ST algorithm uses these unique steps:

- (i) Compare each application value to earlier application values in a sequential manner
- (ii) Use a set of procedures to find spikes, which will ultimately result in the score for the current application
- (iii) When creating the application's score, take attribute weights into account
- (iv) At the conclusion of each mini-discrete data stream, determine the primary qualities that influence the computation of the ST suspicion score and modify attribute weights. This study attempts to offer a comprehensive and flexible methodology for identifying fraudulent activity in credit card application processes by fusing findings from ST with the previous CT approach.

5. Results and Comparisons

Table 1 shows the results for verifiable credential status based on the blockchain-based valid signature of the user or the invalid signature of the user applying for a credit card.

Here, serial numbers 1 and 5 are associated with an invalid signature, whereas the rest of the records have a valid signature based on the Module 1 output.

The outputs of Module 2, based on data mining algorithms – CT and ST algorithms – provide the credit card application form's accepted or rejected status as follows. [10]. Table 3 shows processed applications with a status of application as accepted or rejected.

6. Summary of Comparative Results

Table 4 shows the comparison between the blockchain-based system and the traditional identity

crime detection system. Fraud detection accuracy: The blockchain-enabled solution surpassed the old resilient identity crime detection system with a higher accuracy rate (95% vs. 82%). Efficiency improvement: The blockchain-enabled solution demonstrated a significant reduction in processing time (35%), enhancing operational efficiency. In contrast, the old system showed negligible improvements. User satisfaction: Users expressed greater satisfaction (4.6) with the blockchain-enabled solution, indicating an improved user experience. Conversely, the old system received a lower satisfaction rating (3.1). Data privacy and control: In accordance with the principles of self-sovereign identification, blockchain technology provides exceptional data privacy and control (4.8). In this area, the outdated system performed worse, earning a lower grade of 2.5. Security effectiveness: With a higher security effectiveness score of 4.9, the blockchain system appeared to be more resilient to fraudulent attempts. Comparatively speaking, the security rating of the previous system was lower (3.4). Benefit-cost analysis: Compared to the previous system (\$600,000/year), the blockchain-enabled solution resulted in larger cost savings (\$1.2 million/year). User adoption rate: Compared to the previous system, which had a user adoption rate of 55%, the blockchain-enabled solution showed a much higher rate of 87%. Regulatory compliance: While the traditional system found it difficult to achieve these standards (3.2), the blockchain solution exhibited superior compliance with regulatory regulations (4.7). Scalability: While the traditional system showed medium scalability, the blockchain solution demonstrated high scalability, efficiently supporting rising demand.

7. Conclusion

The paper presents a blockchain-based framework for combating credit card fraud, combining

self-sovereign identity management with advanced data-mining algorithms. This system enhances security, efficiency, and user-centricity, reducing reliance on centralized systems and mitigating unauthorized access risks. The results indicate superior fraud detection accuracy, operational efficiency, and regulatory compliance.

8. Future Scope

The blockchain solution suggests expanding data mining capabilities and improving data privacy. The previous system recommends investigating blockchain for more extensive security uses. The benefits of the blockchain-enabled self-sovereign identity management and data mining solution over the traditional robust identity crime detection system are highlighted in this comparative analysis, especially with regard to accuracy, efficiency, user satisfaction, data privacy, security, cost savings, and user adoption. However, certain organizational needs and goals will determine which of the two systems to use. The work on artificial intelligence integration with blockchain technology shows promise for improving fraud prevention systems, enhancing anomaly detection, and expanding its application to e-commerce.

References

- Alex, P., & Reed, D. (2021). *Self-Sovereign Identity: Decentralized Digital Identity and Verifiable Credentials*. United States: Manning Publications.
- Bolton, R.J., & Hand, D.J. (2001). A survey of credit card fraud detection techniques. *Expert Systems with Applications*, 20(4), 125-130.
- Doe, J., Smith, A., & Brown, B. (2022). Practical implementation of self-sovereign identity in financial services. *IEEE Transactions on Services Computing*, 15(1), 124-134.
- Herenj, A., & Mishra, S. (2013). Secure mechanism for credit card transaction fraud detection system. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(2).
- Jha, S., Gupta, S., & Kumar, S. (2017). Fraud detection in banking using data mining. *IEEE Transactions on Dependable and Secure Computing*, 14(3), 297-309.
- Kshetri, N., & Voas, J. (2020). Decentralised identity management on blockchain. *IEEE Software*, 37(4), 76-82.
- Latchoumi, T.P., & Vijay Kannan, V.M. (2013). Synthetic identity of crime detection. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(7), 551-560.
- Mistry, S., et al. (2019). A blockchain-based identity management system. *IEEE Transactions on Dependable and Secure Computing*, 16(6), 1025-1038.
- Phua, C., Smith-Miles, K., Lee, V., & Gayler, R. (2010). Adaptive spike detection for resilient data stream mining.
- Phua, C., Smith-Miles, K., Lee, V., & Gayler, R. (2012). Resilient identity crime detection. *IEEE Transactions on Knowledge and Data Engineering*, 2(3), 533-546.
- Shakadwipi, A.J., Jain, D.C., & Nagini, S. (2023). Detection of identity theft in credit card application forms through data mining techniques utilizing multilayer algorithms. *Journal of Namibian Studies*, 35, 49-64.
- Shakadwipi, A.J., Jain, D.C., & Nagini, S. (2023a). Credit card application form identity crime detection using data mining algorithm with multilayer algorithm. *SJIS*, 35(1), 212-218.
- Shukla, N., & Pandey, S. (2012). Document fraud detection with the help of data mining and secure substitution method with frequency analysis. *International Journal of Advanced Computer Research*, 2(2), 149.
- Smith, J., et al. (2022). Blockchain-based identity verification for financial services. *IEEE Transactions on Services Computing*, 15(3), 1081-1093.
- Stallings, W., Li, C., & Rai, R. (2021). Self-Sovereign Identity Frameworks: A Comprehensive Review. *IEEE Internet Computing*, 25(1), 42-50.
- Swathi, M., & Kalpana, K. (2013). Spirit of identity fraud and counterfeit detection. *International Journal of Computer Trends and Technology (IJCTT)*, 4(6).
- Vidhya, K., & Dinesh Kumar, P. (2013). Multi-secure approach for credit card application validation. *International Journal of Computer Trends and Technology*, 4(2), 120-123.
- Wang, Y., Zhang, R., & Xie, T. (2019). Blockchain and data mining integration for improved security. *IEEE Transactions on Industrial Informatics*, 15(8), 4691-4698.
- Zohrevand, A., et al. (2020). Blockchain and data mining integration: A survey. *IEEE Access*, 8, 23125-23149.

AUTHOR BIOGRAPHIES



Mr. Amol Jagdish Shakadwipi is a research scholar in the Department of Computer Science and Engineering at Oriental University, Indore. He completed his Bachelor's from SNJB's K.B. Jain College of Engineering, University of Pune, and his Master's in Computer Engineering from SRES College of Engineering, University of Pune. He has 12 years of experience in the academic sector at SNJB's K.B. Jain College of Engineering, Chandwad.



Dr. Dinesh Chandra Jain is a research supervisor in the Department of Computer Science and Engineering at Oriental University, Indore. He has over 18 years of teaching and 6 years of research expertise.



Dr. S. Nagini is a research co-supervisor in the Department of Computer Science and Engineering at Oriental University, Indore. She brings over 24 years of teaching and 5 years of research expertise, holding a Ph.D. in Data Mining.

INSTRUCTIONS TO AUTHORS

Submission of papers

The International Journal of Systematic Innovation is a refereed journal publishing original papers four times a year in all areas of SI. Papers for publication should be submitted online to the IJoSI website (<http://www.ijosi.org>) In order to preserve the anonymity of authorship, authors shall prepare two files (in MS Word format or PDF) for each submission. The first file is the electronic copy of the paper without author's (authors') name(s) and affiliation(s). The second file contains the author's (authors') name(s), affiliation(s), and email address(es) on a single page. Since the Journal is blind refereed, authors should not include any reference to themselves, their affiliations or their sponsorships in the body of the paper or on Figs and computer outputs. Credits and acknowledgement can be given in the final accepted version of the paper.

Editorial policy

Submission of a paper implies that it has neither been published previously nor submitted for publication elsewhere. After the paper has been accepted, the corresponding author will be responsible for page formatting, page proof and signing off for printing on behalf of other co-authors. The corresponding author will receive one hardcopy issue in which the paper is published free of charge.

Manuscript preparation

The following points should be observed when preparing a manuscript besides being consistent in style, spelling, and the use of abbreviations. Authors are encouraged to download manuscript template from the IJoSI website, <http://www.ijosi.org>.

1. *Language.* Paper should be written in English except in some special issues where Chinese may be acceptable. Each paper should contain an abstract not exceeding 200 words. In addition, three to five keywords should be provided.
2. *Manuscripts.* Paper should be typed, single-column, double-spaced, on standard white paper margins: top = 25mm, bottom = 30mm, side = 20mm. (The format of the final paper prints will have the similar format except that double-column and single space will be used.)
3. *Title and Author.* The title should be concise, informative, and it should appear on top of the first page of the paper in capital letters. Author information should not appear on the title page; it should be provided on a separate information sheet that contains the title, the author's (authors') name(s), affiliation(s), e-mail address(es).
4. *Headings.* Section headings as well as headings for subsections should start from the left-hand margin.
5. *Mathematical Expressions.* All mathematical expressions should be typed using Equation Editor of MS Word. Numbers in parenthesis shall be provided for equations or other mathematical expressions that are referred to in the paper and be aligned to the right margin of the page.
6. *Tables and Figs.* Once a paper is accepted, the corresponding author should promptly supply original copies of all drawings and/or tables. They must be clear for printing. All should come with proper numbering, titles, and descriptive captions. Fig (or table) numbering and its subsequent caption must be below the Fig (or table) itself and as typed as the text.
7. *References.* Display only those references cited in the text. References should be listed and sequenced alphabetically by the surname of the first author at the end of the paper. For example:

Altshuller, G. (1998). *40 Principles: TRIZ Keys to Technical Innovation*, Technical Innovation Center.
Sheu, D. & Lee, H. (2011). A Proposed Process for Systematic Innovation, *International Journal of Production Research*, Vol. 49, No. 3, 2011, 847-868.


The International Journal of Systematic Innovation

Journal Order Form

Organization Or Individual Name	
Postal address for delivery	
Person to contact	Name: _____ e-mail: _____ Position: _____ School/Company: _____
Order Information	I would like to order ____ copy(ies) of the <i>International Journal of Systematic Innovation</i> Period Start: 1st/ 2nd half ____, Year: ____ (Starting 2010) Period End : 1st/ 2nd half ____, Year: ____ Price: Institutions: US \$150 (yearly) / NT 4,500 (In Taiwan only) Individuals: US \$50 (yearly) / NT 1500 (In Taiwan only) (Local postage included. International postage extra) E-mail to: IJoSI@systematic-innovation.org or fax: +886-3-572-3210 Air mail desired <input type="checkbox"/> (If checked, we will quote the additional cost for your consent)
Total amount due	US\$
Payment Methods: 1. Credit Card (Fill up the following information and e-mail/ facsimile this form to The Journal office indicated below) 2. Bank transfer 3. Account: The Society of Systematic Innovation 4. Bank Name: Mega International Commercial BANK 5. Account No: 020-53-144-930 6. SWIFT Code: ICBCTWTP020 7. Bank code : 017-0206 8. Bank Address: No. 1, Xin'an Rd., East Dist., Hsinchu City 300, Taiwan (R.O.C.)	

VISA / Master/ JCB/ AMERICAN Cardholder Authorization for Journal Order

Card Holder Information

Card Holder Name	(as it appears on card)		
Full Name (Last, First Middle)			
Expiration Date	/ (month / year)	Card Type	<input type="checkbox"/> VISA <input type="checkbox"/> MASTER <input type="checkbox"/> JCB
Card Number	□□□□-□□□□-□□□□-□□□□	Security Code	□□□ 
Amount Authorized		Special Messages	
Full Address (Incl. Street, City, State, Country and Postal code)			

Please Sign your name here _____ (same as the signature on your card)

The Society of Systematic Innovation

6 F, #352, Sec. 2, Guanfu Rd,
Hsinchu, Taiwan, 30071, R.O.C.