Volume 49 Number 1 March 2025 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics

The Oath of Researchers and Developers

Special Issue:

SoICT 2023

Guest Editors:

Huynh Binh,
Ichiro Ide,
Minh Triet Tran

1977

Editorial Boards

Informatica is a journal primarily covering intelligent systems in the European computer science, informatics and cognitive community; scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor from the Editorial Board can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the list of referees. Each paper bears the name of the editor who appointed the referees. Each editor can propose new members for the Editorial Board or referees. Editors and referees inactive for a longer period can be automatically replaced. Changes in the Editorial Board are confirmed by the Executive Editors.

The coordination necessary is made through the Executive Editors who examine the reviews, sort the accepted articles and maintain appropriate international distribution. The Executive Board is appointed by the Society Informatika. Informatica is partially supported by the Slovenian Ministry of Higher Education, Science and Technology.

Each author is guaranteed to receive the reviews of his article. When accepted, publication in Informatica is guaranteed in less than one year after the Executive Editors receive the corrected version of the article.

The Oath of Researchers and Developers

Matjaž Gams
Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
E-mail: matjaz.gams@ijs.si

Editorial

Misinformation, biased research, and unethical tech use are on the rise, threatening trust in science. The Oath of Researchers and Developers sets clear, practical guidelines to counteract fake news, misleading studies, and irresponsible innovation. Unlike vague ethical codes, it provides concrete examples of ethical dilemmas and real-world consequences. Inspired by the Hippocratic Oath, it promotes transparency, accountability, and integrity, ensuring science works for society, not against it.

1 Introduction and motivation

In an age where facts and fiction blur more than ever—where multimedia platforms broadcast unverified information to a global audience in an instant—questions about trust, authority, and accountability have never been more urgent. Encyclopaedias and scientific journals were once considered unassailable sources of knowledge, yet they, along with news outlets, now grapple with significant ideological and commercial pressures. Likewise, ethical standards, once considered cornerstones of academic and professional integrity, risk erosion in an environment that often prioritizes speed and sensationalism over thoughtful analysis. The rapid advent of artificial intelligence (AI) further complicates this landscape, providing transformative opportunities even as it raises critical questions about bias, responsibility, and transparency.

Beyond ideologies, politics, and fake news, unscientific trends continue to gain momentum through the pervasive influence of social networks. A moderately well-known influencer can attract thousands—or even tens of thousands—of views, which is often one or two orders of magnitude more than the readership of a typical scientific paper. Meanwhile, social-media content is typically of lower quality than everyday discourse, whereas scientific publications must meet rigorous standards to uphold objective truth. As a result, we are witnessing not only the spread of "fake news", but also the rise of fake social networks, deceptive websites, and even partially fraudulent encyclopedias—trends that indicate the emergence of "fake science."

Reversing this trajectory is crucial. To maintain public trust in science, we must protect information integrity and promote evidence-based content. Reaffirming the core values underlying research and development has thus become an essential moral and intellectual imperative.

In this editorial, we address these challenges by returning to one of the earliest pledges of professional ethics: the Hippocratic Oath. Although originally designed for medical practitioners, its guiding principles—"do no harm," uphold beneficence, and maintain integrity—are remarkably relevant across all fields of inquiry. By adapting and modernizing the Hippocratic Oath with the support of contemporary resources and large language models (LLMs), we seek to blend time-honored wisdom with modern-day realities, providing a means to reinforce ethical conduct and restore confidence in scientific endeavors.

We then introduce the Oath of Researchers and Developers, conceived not merely as a modern derivative of existing ethical codes, but rather as an active affirmation of enduring principles reinterpreted for a new generation. Through real-world examples—from addressing data privacy issues to counteracting misinformation—we illustrate tangible steps researchers and developers should take to uphold these commitments. Our vision is to guide early-career professionals toward a practice that respects both the innovative momentum of the digital age and the steadfast virtues of responsible scholarship.

Finally, we compare our adapted Hippocratic Oath and the newly proposed Oath of Researchers and Developers to other notable declarations in the broader field of research ethics. This comparative perspective underscores where our proposals converge with widely accepted norms and where they chart new ground, highlighting how each framework contributes to shaping a more ethically grounded future for research and development.

2 Modern Hippocratic Oath (revised)

1. Prioritize Patient Well-being:
   I will make the health and well-being of my patients my primary concern.

2. Maintain Competence and Continuous Learning:
   I will commit to lifelong learning and maintain the highest standards of medical practice through continuous professional development.

3. Respect Patient Autonomy, Dignity, and Privacy:
   I will respect the autonomy of my patients, ensuring informed consent and protecting their dignity and privacy.

4. Practice Prudent and Safe Medicine:
   I will practice medicine judiciously, avoiding unnecessary interventions and seeking consultation when appropriate to minimize harm.

5. Protect Patient Confidentiality:
   I will safeguard the confidentiality of patient information.

6. Ensure Equitable and Accessible Care:
   I will strive to provide fair and accessible care to all patients, regardless of their background or circumstances.

7. Uphold Professional Ethics and Integrity:
   I will adhere to the highest ethical standards of the medical profession, maintaining honesty and integrity in all my actions.

3 The Oath of researchers and developers

Drawing upon the foundational principles of the medical profession's oath, the Oath of Researchers and Developers underscores the ethical conduct and societal responsibility that underpin advancements in knowledge and technology. The explanations and examples that follow offer practical, concrete scenarios designed to affirm these guiding values.

1. Advance knowledge for human benefit

I will strive to create and disseminate knowledge that benefits humanity, whether by deepening our understanding (declarative knowledge) or through its practical application (operational knowledge, systems, or devices).

Explanation: This principle highlights the importance of contributing positively to society through research and development. Declarative knowledge broadens our theoretical understanding (e.g., discovering a new physical law), while operational knowledge yields tangible solutions (e.g., designing an innovative bridge). Both fundamental research and its practical translation are encouraged, provided they serve the greater good. Crucially, knowledge need not be universally popular or accepted, as long as it holds genuine benefit for humanity.

Examples: Initiatives like social networks or commercial AI programs can be ethically dubious if they risk harming users. In such cases, conscientious researchers should refuse involvement. Conversely, a commercial venture can be justified if it contributes meaningfully—for instance, by developing a new product that tangibly improves day-to-day life.

2. Consider overall and societal impact and mitigation of harm

I will thoroughly evaluate the potential societal impact of my work, striving to maximize positive outcomes and minimize potential harm—even when developing technologies with dual-use potential.

Explanation: This pledge underscores the ethical responsibility to anticipate and mitigate broader consequences (environmental, social, economic, etc.). "Dual-use potential" refers to technologies with both beneficial and harmful applications (e.g., nuclear energy for defence). Here, "harm" refers to harm inflicted on society, rather than personal risk to the researcher or threats to those who may benefit from myths or misconceptions.

Examples: When creating a new AI system, one should have in mind that there is a slight chance of the program going astray, and therefore taking precautions to prevent any such case is a must.

Another example are military applications which are acceptable for researchers as long as it is not clear that the harm outweighs the benefits. An example of a forbidden weapon development would be a fully autonomous weapon that is going to be applied on humans or giving AI a decisive role in using nuclear bombs.

3. Adhere to established scientific principles and knowledge

I will respect both empirically verified science and generally acknowledged truths, refraining from dismissing well-founded facts. In cases of error, I will conduct thorough investigations, acknowledge mistakes, and publish necessary corrections or retractions.

Explanation: Social and cultural arenas—such as films—often explore imaginative or science-fiction concepts, spanning both physical and mental realms. In contrast, our actual physical and technological environment is rigorously studied and validated, requiring scientists to adhere to established facts. Researchers who engage in the deliberate and repeated falsification or fabrication of data or research results must be held accountable for scientific misconduct, which may include removal from the scientific community upon verified proof. However, sound evidence and thorough evaluation—rather than public consensus—determine scientific validity. Galileo Galilei's heliocentric theory, once condemned as heresy, ultimately prevailed because it was grounded in verifiable observations rather than popular opinion.

Examples: There are extreme views, such as the denial of anthropogenic climate change, vaccine conspiracies, or the flat-Earth theory. These beliefs, however, are categorically refuted by overwhelming scientific evidence. For instance, the spherical shape of the Earth is undeniably proven by satellite imagery and countless historical voyages of circumnavigation.

Many countries allow individuals to legally change their gender, illustrating the distinction between social/legal constructs and biological realities. While almost all humans possess XX or XY chromosomes, a fundamental scientific fact, asserting that a biological male is not a man is scientifically inaccurate and should be avoided in scientific discourse. However, in non-scientific settings, individuals may be identified in diverse ways.

4. Pursue, communicate, and share truth transparently and accountably

I will rigorously pursue truth, communicate my findings honestly and transparently, and actively share knowledge within both the scientific community and the broader public. I will remain accountable for my methods, ensuring they are open to scrutiny, while protecting confidentiality only when it serves the greater good.

Explanation: Honest pursuit of truth involves conducting rigorous research, reporting all results—including contradictory ones—accurately, and clarifying the probabilistic or provisional nature of scientific findings. Truth, in this context, stands independent of the source, messenger, or ideology and should be defended even when it is unwelcome or risks personal consequences. Researchers who commit fabrication, falsification, or plagiarism undermine the integrity of science, while those who fail to support known scientific facts in public—or remain silent—also neglect their professional responsibilities.

Transparency not only promotes reproducibility and peer review—pillars of scientific progress—but also bolsters public trust. When researchers share their data, methods, and reasoning openly, they enable others to verify claims and build upon discoveries. However, there may be rare instances (e.g., matters of national security, intellectual property) where certain details must be withheld for the greater good; such decisions demand clear justification.

Finally, it is crucial to distinguish between personal opinion and research-based opinion. Everyone enjoys freedom of personal expression, as typically guaranteed by law, but if a researcher's claim falls within their field of expertise, it must withstand professional scrutiny. Persistently promoting false scientific claims while posing as an expert undermines credibility and may have consequences.

Examples: Researchers have an obligation to disseminate established facts—for instance, that Earth is round—even at some personal cost. Only under truly severe circumstances should they postpone sharing such knowledge. In certain extreme cases, scientists asserting there are two biological sexes have been fired. While losing one's position may be considered a "harsh consequence" that warrants pragmatic caution, it remains crucial to defend factual truths and remove from the research community those who persistently deny them.

Assume a researcher in the field of economics finds that selling a national bank is not beneficial for the country. Should he actively promote this analysis or not? The Oath clearly demands promotion of relevant revelations for the local and global community. Not doing so is close to neglecting the Oath.

Finally, a professor endorsing a policy outside their field of expertise is free to hold and express a personal stance, provided it does not misrepresent scientific consensus. However, if they claim such views are rigorously tested within their discipline—despite contradictory evidence—professional accountability mechanisms should apply.

5. Uphold ethical standards and professional responsibility

I will adhere to the highest ethical standards and act with professional responsibility in all my research and design endeavors.

Explanation: This is a general commitment to ethical and responsible conduct, encompassing adherence to professional codes, regulations, and best practices. Adherence to ethical standards and professional integrity safeguards the welfare of individuals and communities impacted by research outcomes, reinforcing the commitment to ethical conduct. Not all can be defined in the Oath for directing proper research activities. Such an examples are ethical and professional standards that should be followed as much as possible.

Examples: Researchers often cooperate in research groups with professional and social interactions. The joint research principles and goals should prevail over the local ones, e.g., not helping a colleague in trouble with a motive to achieve a better position due to that activity. Collegial Support: In a research group, members should collaborate openly and assist one another, rather than withholding help to gain a competitive edge.

Conflict of interest: Researchers must disclose funding sources, affiliations, or any other potential conflicts to ensure transparency and maintain trust in their findings. Addressing Misconduct: If a colleague's work shows serious flaws or unethical practices, it is every researcher's responsibility to address the issue—protecting the integrity of the field, even if doing so may be personally or professionally challenging.

Fair peer review: Evaluating colleagues' work should involve impartial judgment, free from personal biases or favoritism, to maintain a fair and constructive scientific environment.

In particular, no form of discrimination—whether sexism, ageism, racism, or any other form of prejudice—should be tolerated or perpetuated.

6. The self-healing role of the scientific community

I will support and actively promote only researchers following the Oath and insist on detaching the ones not following the Oath from scientific ranks and in extreme cases from all positions. I shall support fellow scientists when they face unjust attacks, provided they adhere to this Oath.

Explanation: Just as any organism requires self-defence mechanisms to survive, the scientific community must protect itself against unjust assaults. This can be likened to the human immune system. What, then, are the safeguards of true science? **We** researchers and developers and our societies are responsible for defence of true science and scientists. We should defend members of our society from the unfounded ideological attacks, and we should refrain from supporting individuals for scientific or developmental positions if they do not adhere to the principles of this Oath.

Examples: Throughout history and in recent years, numerous scientists have faced attacks for disseminating the truth and debunking myths. Unfortunately, these scientists often lacked support from the scientific community and organizations such as associations and academies. In some cases, ideological attacks on scientists were even endorsed by their peers. It is critical for the scientific community to recognize that allowing ideological viewpoints to prevail severely undermines true science. Empirical evidence allows us to perceive the natural gender of a human, regardless of formal or declarative identification. If science adopts ideological stances, it ceases to be science. The Earth becomes flat.

Another example is that of an editor at the highly influential journal Scientific American, who resigned in protest after the U.S. presidential election produced a result contrary to their preference, and then used social media to ridicule ‘uneducated voters.’ Can such a person truly advocate for objective science and development, free from political and ideological bias?

7. Foster education, mentorship, and public engagement

I will actively contribute to the education and mentorship of aspiring researchers, promote interdisciplinary collaboration, and engage with the broader public to enhance scientific understanding and literacy.

Explanation: This commitment recognizes that research thrives when knowledge is openly shared across generational, disciplinary, and societal boundaries. Mentorship and education ensure the continuation of robust scientific inquiry, while public engagement bridges the gap between specialized expertise and community needs. By investing in teaching, outreach, and collaborative efforts, researchers help cultivate a more informed society and a stronger foundation for future innovations.

Examples: Mentorship: Guiding students or early-career professionals through project supervision, workshops, or individual consultations.

Public Outreach: Delivering public lectures, creating educational content, or participating in community events to spark curiosity and spread scientific awareness.

Interdisciplinary Collaboration: Organizing seminars or research initiatives that connect experts from varied fields to tackle complex global challenges—such as climate change, healthcare, or AI governance.

The Oath of Researchers and Developers can be summarized as:

1. **Serve Humanity:** Ensure your work ultimately benefits people and our civilization, whether advancing theoretical understanding or yielding practical solutions.

2. **Be Aware of Societal Impact:** Thoroughly assess potential risks and benefits, striving to maximize positive outcomes and minimize harm—even for dual-use technologies.

3. **Honor and Protect Established Facts:** Adhere to verified scientific principles, investigate errors honestly, and rectify or retract findings if proven invalid.

4. **Pursue and promote truth proactively:** Actively engage in seeking accurate results, communicate them transparently, and remain accountable for your methods—always distinguishing personal viewpoints from rigorously validated scientific and engineering conclusions. It is your duty to promote, uphold, and defend the truth, even in the face of negative backlash.

5. **Maintain ethical and professional standards:** Respect recognized ethical standards and practice responsible conduct in all research and development activities.

6. **Protect the scientific and developers’ community and individuals:** Preserve the integrity and security of our field by actively supporting colleagues who uphold these standards against ideological or public attacks, and ensure that those who fail to follow the Oath are not placed in influential positions.

7. **Educate and engage:** Mentor future researchers, foster interdisciplinary cooperation, and engage the public to enhance scientific understanding and literacy.

4 Similarities between the oaths

4.1 With the Hippocratic Oath

While the Hippocratic Oath centers on the welfare of patients and the Oath of Researchers and Developers focuses on advancing knowledge and technology for the benefit of humanity, both share core ethical themes:

- **Beneficence (Acting for the Good)**

  Physicians commit to promoting patient well-being; similarly, researchers and developers pledge to harness knowledge and technology for the greater social good.

- **Non-Maleficence (Avoiding Harm)**

  Doctors endeavor to minimize harm by avoiding unnecessary interventions. Researchers and developers, in turn, strive to foresee and mitigate any societal, environmental, or economic harm their work might cause.

- **Truthfulness and transparency**

  Both Oaths value honesty and open communication. Physicians must accurately diagnose and share relevant information with patients, while researchers and developers must disclose findings and methodologies clearly, enabling peer review and public trust.

- **Confidentiality and protection of information**

  Medical professionals safeguard patient confidentiality. Researchers and developers likewise maintain discretion, protecting sensitive information unless the public interest unequivocally demands disclosure.

- **Upholding ethical standards**

  Each Oath underscores the highest ethical principles, whether in the practice of medicine or in research and development. By maintaining professional integrity and respecting established norms, both communities protect the welfare of those they serve.

4.2 Relations with other ethical codes and principles

While the Oath of Researchers and Developers draws heavily from the Hippocratic Oath, it also aligns with established ethical frameworks across various domains:

- **Researcher conduct**

Ethical Guidelines for Research (e.g., National Institutes of Health (NIH) Guidelines for Research, European Code of Conduct for Research Integrity): Institutions and funding agencies often issue guidelines emphasizing integrity, honesty, objectivity, and responsible research conduct—covering areas like data integrity, plagiarism, authorship, and conflicts of interest. These align with the Oath’s commitment to truthfulness, transparency, and high ethical standards.

Codes of Conduct for Scientific Societies (e.g., American Chemical Society Code of Conduct, World Medical Association Declaration of Helsinki): Many professional organizations stipulate standards for research, publication, and collegial interactions. These typically promote scientific integrity, open communication, and the avoidance of misconduct, echoing the Oath’s emphasis on honesty, transparency, and accountability.

- **Engineering ethics**

Codes of Ethics for Engineers (e.g., NSPE Code of Ethics, IEEE Code of Ethics): Promulgated by bodies such as the National Society of Professional Engineers (NSPE), these codes stress public safety, environmental protection, and professional responsibility—mirroring the Oath’s principles of assessing societal impact, minimizing harm, and upholding ethical standards.

ACM Code of Ethics and Professional Conduct (Association for Computing Machinery): This comprehensive code provides guidance for computing professionals, emphasizing ethical considerations such as honesty, fairness, respect for users' rights, and the importance of public interest. It aligns with the Oath's principles by advocating for responsible use of technology, minimizing harm, and ensuring that computing serves humanity.

- **AI ethics guidelines**

Principles for AI (e.g., OECD AI Principles, UNESCO Recommendation on the Ethics of Artificial Intelligence): Various international organizations and initiatives have proposed guidelines stressing fairness, accountability, transparency, and human oversight in artificial intelligence. These requirements closely parallel the Oath’s calls to avoid bias, safeguard societal welfare, and maintain open, responsible research practices.

- **Universal declaration of human rights**

(United Nations, 1948): Although not specifically crafted for researchers, this declaration champions essential human freedoms and the right to benefit from scientific progress. It underpins the Oath’s overarching vision of advancing knowledge for humanity’s collective well-being while carefully considering potential societal impact.

4.3 Similarities between the Oath and the EU AI Act

- **Focus on human well-being**

Oath: Prioritizes the advancement of knowledge for human benefit and the minimization of potential harm.

EU AI Act (Regulation (EU) 2024/1689): Seeks to ensure AI systems uphold human rights, safety, and key societal values, banning applications that pose a clear threat to well-being and fundamental rights.

- **Transparency and accountability**

Oath: Demands openness in research methods, findings, and accountability for actions.

EU AI Act: Requires high-risk AI systems to incorporate human oversight and mandates transparency, ensuring that users can understand and challenge AI-driven decisions.

- **Ethical considerations**

Oath: Stipulates adherence to rigorous ethical standards and professional responsibility.

EU AI Act: Establishes a trustworthy AI framework, emphasizing fairness, non-discrimination, and respect for personal data throughout the AI lifecycle.

- **Risk assessment and mitigation**

Oath: Encourages researchers to proactively assess potential risks and strive to minimize harm.

EU AI Act: Categorizes AI systems by risk level and imposes requirements for conformity assessments, ongoing monitoring, and risk management for high-risk applications.

- **Avoiding harm**

Oath: Explicitly discourages actions likely to harm society.

EU AI Act: Prohibits AI systems deemed dangerous to public safety, livelihoods, and rights, also barring practices that manipulate or exploit human vulnerabilities.

By comparing these core principles, it appears that both the Oath of Researchers and Developers and the EU AI Act converge on the goal of promoting responsible, ethical innovation. The Oath provides a framework for individual researchers, while the EU AI Act offers legal and regulatory guidelines for AI deployment across the European Union.

5 Conclusions

Acknowledging these connections, the Oath of Researchers and Developers draws upon a broad legacy of ethical principles and contributes to a diverse, evolving body of guidelines that govern research and development in numerous fields. By redefining the values and motivations of researchers and designers, this text aims to serve both present and future generations.

It is essential to acknowledge that our civilization faces escalating pressures. Even the most fundamental scientific truths face scrutiny, and scientists who resist non-scientific ideologies risk attacks or even the loss of their positions. It is on us to reintroduce basic scientific and ethical standards of an advanced civilisation.

Always remember that science and development rank among humanity’s greatest assets, propelling technological progress and enhancing quality of life. Science and development are the most noble of professions with highest standards and also responsibilities, dedicated to preserving truth and objectivity in the pursuit of the common good.

Acknowledgement

We thank Tine Kolenik, Tamara Lah and Peter Glavič for their suggestions.

Disclaimer

This information is for general knowledge and informational purposes only and does not constitute legal or professional advice.

Memetic Algorithm for Maximizing K-coverage and K-Connectivity in Wireless Sensor Network

Hanh Nguyen Thi[1], Cuong Van Duc[2], Chinh Tran Duc[2], Hieu Ha Minh[2], Son Nguyen Van[3] and Quan La Van[3]
[1]Faculty of Interdisciplinary Digital Technology, PHENIKAA University, Yen Nghia, Ha Dong, Hanoi, 12116, Vietnam
[2]Hanoi University of Science and Technology, Hanoi, Vietnam
[3]Faculty of Computer Science, PHENIKAA University, Yen Nghia, Ha Dong, Hanoi, 12116, Vietnam
E-mail: hanh.nguyenthi@phenikaa-uni.edu.vn, cuong.vd220021@sis.hust.edu.vn, chinh.td224936@sis.hust.edu.vn, hieu.hm220026@sis.hust.edu.vn, son.nguyenvan@phenikaa-uni.edu.vn, quan.lavan@phenikaa-uni.edu.vn

Keywords: Target coverage, connectivity, node deployment, heuristic algorithm, memetic algorithm, local search, prim algorithm

Received: July 21, 2024

The rapid growth of IoT has enabled diverse applications using Wireless Sensor Networks across various fields. A significant challenge in Wireless Sensor Networks is the efficient deployment of sensors to ensure coverage and connectivity. Effective coverage allows continuous target tracking and data collection, while connectivity ensures data transmission to the base station. In this paper, we address the challenge of maximizing the number of targets satisfying K-coverage and K-connectivity, where each target is tracked by K sensors and has K transmission paths to the base station. We propose a two-phase methodology to tackle this challenge. The first phase enhances the Greedy algorithm to solve the K-coverage problem. The second phase addresses the K-connectivity problem using Memetic algorithms augmented by an efficient local search mechanism called PMA. We evaluate the algorithm on various datasets and compare it with baseline methods, including Greedy and Prim-based with the withdrawal strategy (PWS). Our results show that the proposed PMA with a robust local search outperforms alternative algorithms, with improvements exceeding 10% *to* 15% *compared to the baseline methods. Additionally, we validate the performance of the proposed method using a real-world dataset and outline plans for further enhancements in the near future.*

Povzetek: Avtorji so razvili dvofazni pristop za maksimiranje K-pokritosti in K-povezljivosti v brezžičnih senzorskih omrežjih, ki združuje izboljšan pohlepni algoritem in memetični algoritem z lokalnim iskanjem, imenovan PMA.

1 Introduction

Amidst rising environmental concerns, escalating global political tensions, and the widespread proliferation of Internet of Things (IoT) technology and products, particularly emphasizing privacy and security, Wireless Sensor Networks (WSNs) have attracted significant attention[15, 1]. WSNs are composed of sensors equipped with data collection capabilities. Devices must be outfitted with sensor chips capable of detecting environmental phenomena and converting them into accessible data on the Internet for users to analyze and process[19]. These sensors collect data from specific areas and transmit monitoring information to a central base station.

WSNs find crucial applications across various domains, such as military operations, healthcare services, environmental monitoring, biodiversity studies, industrial processes, and urban infrastructure management [20, 7, 12, 5]. For example, wearable, embedded, or ingestible sensors enable continuous monitoring of health parameters and vital signs, such as blood pressure or heart rate, offering vital insights wherever patients or caregivers are situated. The proliferation of WSNs has stimulated significant scientific research and publications aimed at tackling key challenges in sensor networks, including issues related to lifetime, coverage, connectivity, fault tolerance, load balancing, and security. In addition, sensors, characterized by their compact dimensions, face limitations in storage capacity, operational lifespan, and susceptibility to environmental conditions. In areas where battery replacement or recharging is impractical—such as hazardous or obstructed environments like deep oceans or dense forests—ensuring continuous multi-coverage and connectivity between targets and base stations becomes crucial for maintaining network integrity, particularly in scenarios involving potential node failures. Therefore, in this paper, we focus on solving the problem of K-coverage and K-connectivity.

Coverage concern is divided into various subproblems, such as target coverage, area coverage, and barrier coverage. In this study, we focus on resolving the target coverage problem[16, 22]. The target coverage problem involves guaranteeing thorough monitoring of specified targets within a designated surveillance area through strategically placed sensors. The aim is to ensure that every target falls within the sensing range of at least one sensor, enabling comprehensive monitoring and detection capabilities. This challenge is critical across various applications such as environmental monitoring, surveillance, and intrusion detection, where adequate coverage of specific targets is vital for operational efficacy and informed decision-making. The target coverage problem includes 1-coverage, K-coverage, Q- coverage. Within this context, 1-coverage guarantees that all targets are monitored by at least one sensor, K-coverage ensures at least K sensors track each target, and Q-coverage ensures that targets are tracked by Q sensors, the specific value of Q can be adjusted based on priority requirements. In this paper, our primary objective is to resolve the K-coverage problem[8].

Connectivity in WSNs denotes the capacity of sensors to establish and sustain communication links within the network[18]. This ensures reliable data transmission among sensors, facilitating seamless information flow throughout the network. Connectivity is pivotal for fostering collaboration among sensors, streamlining data aggregation, and bolstering network functions like routing and data forwarding. A well-connected network enhances efficiency, resilience, and reliability, enabling effective monitoring and communication across various applications. Connectivity issues encompass 1-connectivity, K-connectivity, and Q-connectivity. In 1-connectivity, a minimum of 1 communication path exists from the target to the base station. K-connectivity guarantees the presence of at least K disjoint paths from the target to the base station[23]. Finally, Q-connectivity ensures the existence of at least Q disjoint paths from the target to the base station, with the value of Q being adjustable based on the target's priority level. In this paper , our primary objective is to resolve the K-connectivity problem.

Recent research endeavors to address multi-coverage and multi-connectivity [10, 3] have encountered limitations, particularly when prioritizing the minimization of sensors required to meet problem constraints, assuming an unlimited number of sensors. Nevertheless, deploying sensors presents substantial hurdles in environments where battery replacement or recharging is unfeasible, such as hazardous or obstructed locations like deep oceans or dense forests. Consequently, the practicality of sensor deployment is constrained, leading to a limited number of sensors in reality. Hence, our team is dedicated to tackling novel problems that, to our knowledge, have yet to be explored by other research groups. Specifically, we focus on determining the maximum number of targets simultaneously fulfilling K-coverage and K-connectivity requirements, given a fixed number of sensors.

In response to the identified challenge, we propose a two-phase strategy. Initially, we aim to resolve the K-coverage issue by refining the Greedy algorithm. Subsequently, the second phase addressed the K-connectivity problem, employing Heuristic and Memetic algorithms augmented with an efficient local search mechanism. The simulation results indicate that the proposed Memetic algorithm combined with Prim and a robust local search function (PMA) outperforms alternative methods, demonstrating superior performance. Therefore, investigating this problem holds scientific and practical significance. In the subsequent section, we present relevant studies concerning this matter.

Our main contributions are listed as follows:

- Formulating a novel problem of K-coverage and K-connectivity suitable for practical application in the 2D domain.

- Presenting a Greedy based method for node deployment that provides K-coverage to all of targets.

- Proposing two baseline methods: PWS and Greedy combined with withdrawal strategy to address connectivity issues.

- Proposing a new approach called PMA (Prim-based Memetic Algorithms): A special Memetic Algorithm Strategy Enhanced with Robust Local Search for Effective Problem Solving.

- Evaluating the proposed method across 40 experimental and real-world datasets.

The rest of the paper is structured as follows: Section 2 provides a comprehensive review of related works. In Section 3, we present the system model and the problem formulation. The proposed algorithms are detailed in Section 4. Section 5 contains the experimental settings, obtained results on various test sets, and a performance comparison with other algorithms to demonstrate the proposal's efficacy. Section 6 discusses conclusions and future.

2 Related work

Coverage and connectivity are two paramount challenges in WSNs. Specifically, coverage in WSNs pertains to the comprehensive monitoring and surveillance of every point within the designated area of interest. [11] The coverage challenge is categorized into three distinct classes based on the intended application: area coverage, target coverage, and barrier coverage [21], [20]. In this paper, our primary focus is on the target coverage predicament, which has been identified as an NP-hard problem. [16] elucidates the various iterations of target coverage.

The emphasis on the NP-completeness of the coverage problem is attributed to the research conducted by [13]. Consequently, most studies advocate solutions employing integer linear programming, heuristic and metaheuristic algorithms to address this challenge. Integer linear programming, which involves constructing a mathematical model, is one of the methodologies employed to resolve the target coverage quandary [3], [23].

However, its effectiveness is evident primarily when dealing with smaller problem sizes, while it demands increased computing time for larger problem sizes [4].

Henceforth, researchers are increasingly delving into the exploration and utilization of heuristic and meta-heuristic algorithms to address the coverage problem. Chien-Chih Liao et al. [14] propose a novel memetic algorithm (MA) that integrates an integer-coded genetic algorithm with local search techniques to solve the K-coverage problem. This approach adapts crossover and mutation operators to integer representation. It introduces a novel fitness function that considers both the number of covers and the individual contribution of sensors to these covers.

When sensors are deployed, a critical consideration arises: determining whether any node in the network can communicate with any other node. Connectivity thus broadens the scope of the coverage problem, aiming to guarantee the existence of pathways between nodes to facilitate the transmission of collected data to external destinations. Moreover, securing network connectivity is paramount for effective WSNs operations. One prevalent approach is to maintain the K-connectivity property, which ensures that the removal of up to K—1 sensor nodes does not lead to network partitioning, thereby preventing the isolation of one or more sensor nodes from the network.

A common tactic to preserve K-connectivity entails adding new nodes as needed. The principal design aim is to reduce the number of additional nodes required while retaining K-connectivity. As with coverage, connectivity poses an NP-hard problem [2] that can be tackled using linear programming and approximation algorithms. One method to address the target connectivity dilemma is integer linear programming, which entails constructing a mathematical model[22]. Nonetheless, its efficacy is more pronounced in managing smaller problem sizes, necessitating escalated computational resources for larger-scale problems. Consequently, researchers are increasingly venturing into exploring and implementing heuristic and meta-heuristic algorithms to tackle the coverage issue. Szczytowski et al. [18] introduced an innovative method for runtime repair and preservation of global WSN K-connectivity, relying solely on localized information. This approach significantly reduces resource demands compared to previous studies.

In recent years, researchers have focused on addressing the challenges of weak security, connection losses during operation, and damaged relay nodes, aiming to ensure dependable monitoring and information transmission. To mitigate these risks, they have specifically targeted solutions for multiple coverage and multiple connections. In [6], Gupta et al. explored a genetic algorithm (GA)-based approach to identify the minimum number of selected potential positions suitable for deploying sensor nodes in target-based wireless sensor networks, ensuring both K-coverage and M-connectivity of the sensor nodes. The study assumes predefined potential positions for sensor node deployment to monitor targets. Similarly, [17] introduces a method based on the Imperialist Competitive Algorithm (ICA), aiming to identify the minimum number of suitable locations for sensor node deployment while meeting coverage and connectivity requirements.

3 System model and problem formulation

3.1 System model

We assume a Wireless Sensor Network and all sensor nodes in it have the same transmission range. Each target collects information from the environment in the range which it is deployed, this region is assumed to be a circular disk whose radius is equal to the sensing range of a sensor node. Then target transmits that information through the sensor nodes on predetermined paths. Transmitting in different paths avoid losing information, if a sensor has problem lead to a path disconnect, there's still other path to transmit information. Two sensor nodes can connect with each other if the Euclidean distance between them is less than or equal the sensing range. Finally, the information is transferred to the Base Station.

3.2 Problem formulation

Let us define surveillance region A as a rectangular with area $W \times H$ and a set T includes m targets $T = \{T_i(x_i, y_i) | 0 \leq x_i \leq W, 0 \leq y_i \leq H, \forall i \in [1, m]\}$. B is the Base Station in A with coordinates (x_B, y_B). We assume set $S = \{S_1, S_2, ..., S_n\}$ is set of n sensors. Our goal is to place n sensors in region A such that maximize the number of targets that satisfied both K-coverage and K-connectivity.

A sensor node can connect with a target if their Euclidean distance is not greater than the sensing range, denoted r_s . Similar, two sensor notes can connect if their Euclidean distance is not greater than the communication range, denoted r_c. Let $c(S_i, T_j)$ denote the connectivity probability between sensor S_i and target T_j, which is calculated via:

$$c(S_i, T_j) = \begin{cases} 1, & \text{if } d(S_i, T_j) \leq r_s, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

and the number of sensors in each target's sensing range is calculated by the following forrmula:

$$C_{T_j} = \sum_{i=1}^{n} c(S_i, T_j). \quad (2)$$

A target T_j is K-coverage if and only if $C_{T_j} \geq K$.

We assume each target, for example T_j, has a set include K path $P = \{P_i | P_i = (T_j, S_{i1}, S_{i2}, ..., S_{il_i}, B), i \in [1, K], l_i$ is number of sensors nodes in $P_i\}$. Then, these K paths will be disjoint if

$$P_a \cap P_b = \{T_j, B\} \; \forall a, b \in [1, K], a \neq b, \tag{3}$$
$$d(T_j, S_{i1}) \leq r_s \; \forall i \in [1, K], \tag{4}$$
$$d(S_{iu}, S_{i(u+1)}) \leq 2r_c \; \forall i \in [1, K], u \in [1, l_i - 1], \tag{5}$$
$$d(S_{il_i}, B) \leq r_c \; \forall i \in [1, K]. \tag{6}$$

Equation (3) ensures that the K paths have no common sensor. Equation (4) make sure that the target and sensors can connect (Their distance are satisfy the sensing range). Equation (5) make sure that the sensors can connect. Equation (6) make sure that the sensors and the Base Station can connect.

A target is K-connectivity if and only if its K paths are disjoint.

Let E_j is the connectivity and coverage status of target T_j. Then

$$E_j = \begin{cases} 1, & \text{if } T_j \text{is both } K\text{-connectivity} \\ & \quad \text{and } K\text{-coverage}, \\ 0, & \text{otherwise}. \end{cases} \tag{7}$$

From equation (2),(3),(4),(5),(6) and (7) we have problem model:

Maximize

$$\sum_{j=1}^{m} E_j. \tag{8}$$

Subject to

$$C_{T_j} \geq K \; \forall j \in [1, m], \tag{9}$$
$$P_a \cap P_b = \{T_j, B\} \; \forall a, b \in [1, K], a \neq b, \tag{10}$$
$$d(T_j, S_{i1}) \leq r_s \; \forall i \in [1, K], \tag{11}$$
$$d(S_{iu}, S_{i(u+1)}) \leq 2r_c \; \forall i \in [1, K], u \in [1, l_i - 1], \tag{12}$$
$$d(S_{il_i}, B) \leq r_c \; \forall i \in [1, K]. \tag{13}$$

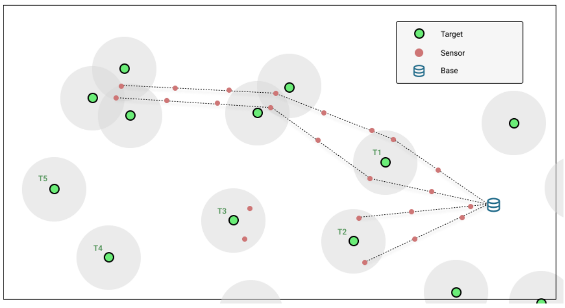


Figure 1: Problem formulation

In **Figure 1**, the targets T_1 and T_2 are characterized by meeting K-coverage and K-connectivity criteria, where K is set to 2. On the other hand, target T_3 achieves K-coverage but fails to meet the K-connectivity requirement. Additionally, targets T_4 and T_5 do not fulfill the K-coverage and K-connectivity criteria.

4 Proposed method

To address the identified challenge, we propose a two-phase methodology. The first phase focuses on resolving the K-coverage issue by enhancing the Greedy algorithm. Subsequently, the second phase is dedicated to tackling the K-connectivity problem, employing Memetic Algorithms augmented by an efficient local search mechanism, PMA. Furthermore, we conduct comprehensive evaluations of the proposed algorithm using various datasets and compare its performance with baseline methods, including Greedy and PWS.

4.1 Coverage phase

In this phase, we aim to determine an optimal approach for placing sensors within region A that can provide K-coverage for each target with the minimum number of sensors. In order to minimize the number of sensors, we apply a Greedy based algorithm.

We consider the set of disks D is the set that conclude the targets which was not satisfied K-coverage. From this set, we construct a set O conclude overlapping regions, which we will use to placed sensors in. An overlapping region is defined by the intersection of the disks in D. And we have a set I that concludes the disks that have no intersection with other disks. For example, in figure below, we have set $O = \{1 \cap 2 \cap 3, 1 \cap 4\}$ and $I = \{5\}$.

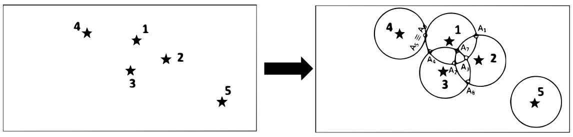


Figure 2: Coverage phase

After having two sets O and I, we will only place sensors in these two sets. At first, we choose a region O_i in O that can cover the most targets. We place K sensors at a random point in it. After placing sensors, if a target satisfies K-coverage, we will remove it from D and update the set O and I. We repeat that procedure until $O = \emptyset$. After placing sensors in O, we start with set I. With every region in I, we repeat placing k sensors at a random point. The entire algorithm is described in **Algorithm 1**.

4.2 Connectivity phase

While targets may meet K-coverage without achieving K-connectivity, and vice versa, these scenarios are not applicable in real-world settings. Therefore, after completing Coverage phase in sensor placement for K-coverage, specialized strategies will optimize sensor positioning. This approach aims to maximize targets by simultaneously meeting both K-coverage and K-connectivity criteria. Based on our understanding, no research studies have addressed the issue we raised. Therefore, we propose the main method, the

```
Algorithm 1: Greedy based algorithm for coverage
Input  : $T$: Set of targets.
         $r_s$: Sensing range.
         $n$: The number of available sensors
Output : Set $R$ concludes candidate regions.
         Location of minimum number of sensor
         nodes needed to satisfy $K$-coverage $S_2$
Build disk set $D = \{D_j(T_j, r_c) | T_j \in T\}$ with each
  disk has center at one target and radius $r_s$
$S_2 = \emptyset, R = \emptyset$;
$O \leftarrow$ set of overlapping regions;
$I \leftarrow$ set of targets without overlapping;
while $O \neq \emptyset$ do
    $O_i \leftarrow$ Region in $O$ can coverage the most
      targets;
    $\{s_1, \ldots, s_K\} \leftarrow$Place $K$ sensors at a random
      point at $O_i$;
    $R = R \cup O_i, S_2 = S_2 \cup \{s_1, \ldots, s_K\}$;
    if a target $T_j$ satisfies $K$-coverage then
        Remove $D_j$ from $D$;
        Update $O, I$;
    end
end
for region $\in I$ do
    $\{s_1, \ldots, s_K\} \leftarrow$Place $K$ sensors at a random
      point in region ;
    $R = R \cup O_i, S_2 = S_2 \cup \{s_1, \ldots, s_K\}$;
end
return $S_2, R$;
```

Prim-based Memetic Algorithms (**PMA**) in Section 4.2.3, alongside two baseline methods, **Greedy** and Prim-based with withdrawal strategy **PWS** in Section 4.2.1, 4.2.2, for comparative analysis.

4.2.1 Greedy based algorithm for connectivity (Greedy)

At first, we introduce a Greedy based algorithm used for maximize the number of targets satisfy both K-coverage and K-connectivity by a limited number of sensors. After coverage phase, we assume number of available sensors is $n_a = n - |S|$. However, in this phase, when we exhaust all of the sensors, we will remove sensors from special regions within coverage phase to optimize the result.

We assume set U concludes the base station and the sensors which is available to connect to the base station and set L concludes the regions which unsatisfactory K-connectivity. Beginning, we initialize $U = \{B\}$ and $L = R$. Each region in L, U is represented by the location of the sensors in it. We define the region in L which has the shortest distance to a point in U. Then, we place sensors in order to connect that region with U. To connect two regions, from a point in first region, we create a path to the nearest point which not have path in the second region and repeat until we have K separate paths between two regions. The sensors amount need to connect two points I and J (I, J can be Base Station or sensor) is calculates by

$$sensor\_amount = [\frac{d(I, J) - 2r_c}{2r_c}] + 1. \tag{14}$$

where [.] denotes the integer part of a number.

If in this process, we exhaust all of the sensors, we will choose the region that have furthest distance to a point in U and remove all sensors in it to use for connect until the sensors amount is enough to connect or there are no more sensors to remove. If the available sensor can not increase the number of target satisfy K-coverage and K-connectivity, we will place them in some region that has density of sensors highest aim to enhance connectivity.

The entire algorithm is described in **Algorithm 2**.

```
Algorithm 2: Greedy based algorithm for connectivity
Input  : $R$: Set of candidate regions.
         $S_2$: Set of sensors use for coverage.
         $n_a$: The number of available sensors
Output : The number of targets satisfy both
         $K$-coverage and $K$-connectivity
$U = \{B\}$ Connected set;
$L = R \leftarrow$ Unconnected set;
$res \leftarrow$ Number of target satisfy $K$-coverage and
  $K$-connectivity;
while $n_a > 0$ or $|L| > 0$ do
    $min_{dist} = +\infty$;
    for $i \in L$ do
        for $j \in U$ do
            if $d(i, j) < min_{dist}$ then
                $min_{dist} = d(i, j)$;
                $l_{choose} = i, u_{choose} = j$;
            end
        end
    end
    $sensor_{need} \leftarrow$ Sensors amount need to connect
      $l_{choose}, u_{choose}$
    while $n_a < sensor_{need}$ and $|L| > 1$ do
        Choose region in $V$ furthest to $U$ and
          remove from $L$;
        $n_a = n_a + K$;
    end
    if $n_a \geq sensor_{need}$ then
        Place sensors;
        Update $L, U, res, n_a$;
    end
    else
        Remove sensors in $L$; $n_a = n_a + K$;
        Place $n_a$ sensors in high sensor density
          region;
    end
end
return $res$;
```

4.2.2 Prim-based with withdrawal strategy (PWS)

First, each region is represented by the location of the sensors in it. We consider a graph $G = (V, E)$ where V conclude Base Station and candidate regions $V = B \cup R$ and $E = \{(u, v, dist(u, v)|u, v \in V, u \neq v\}$ with $dist(u, v)$ is the number of sensors required to connect u to v by K node-disjoint paths calculate by the formula has been presented before. We will apply Prim's algorithm to find the Minimum Spanning Tree starts from B. We define set $S_l = \{v|v\text{is leaf node}\}$. n_{left} is the number of sensors left after placing sensors to the tree (n_{left} can be negative if sensor amount is not enough) calculate by $n_{left} = n_a - n_{MST}$ with n_{MST} is the number of sensor need to placing in Minimum Spanning Tree. Until $n_{left} \geq 0$, we choose node v in S_l that have $dist(v, parent(v))$ is the biggest, remove v from s_l and update n_{left}, s_l. The number of satisfied target is the number of remaining node in the tree (except B).

The entire algorithm is described in **Algorithm 3**.

```
Algorithm 3: Prim-based with withdrawal strategy(PWS)
Input  : $R$: Set of candidate regions.
         $S$: Set of sensors use for coverage.
         $n_a$: The number of available sensors
Output : The number of targets satisfy both
         $K$-coverage and $K$-connectivity
Graph $G = (V, E)$ with $V = B \cup R$ , $E = \{(u, v, dist(u, v)|u, v \in V, u \neq v\}$;
Apply Prim's algorithm to find the Minimum Spanning Tree;
$S_l \leftarrow$ Set of leaf node;
$n_{left} \leftarrow$ the number of sensors left after placing sensors to the tree;
while $n_{left} < 0$ do
    $v \leftarrow$ Leaf node has biggest $dist(v, parent(v))$;
    Remove $v$ from $S_l$;
    update $S_l, n_{left}$
end
$res \leftarrow$ The number of remaining node (except $B$);
return $res$
```

4.2.3 Prim-based memetic algorithms (PMA)

Our proposed method presents an innovative and efficient approach to solving the problem. Each solution element is encoded as a binary vector, representing whether a specific area is utilized for connectivity. This encoding serves as the basis for generating the initial population. We introduce two advanced strategies for crossover and mutation operators to enhance the evolutionary process. These strategies are designed to direct new individuals toward promising regions in the solution space while preserving population diversity, thereby expediting convergence to optimal solutions. Furthermore, a local search mechanism is integrated to refine the best-performing individuals, increasing the potential to escape local optima. A distinctive evaluation mechanism is employed in which unsatisfying individuals are not immediately discarded. Instead, they are retained and evaluated using a specialized strategy, ensuring consistent population diversity throughout the optimization process.

Detailed explanations of each component in the proposed method are provided in the following sections.

4.2.3.1 Individual representation A individual is a vector of integers of size $n + 1$, where n denotes the number of coverage areas. We incremented the count by 1 to incorporate the Base station. Our research paper defines a individual as a significant binary sequence comprising 0s and 1s. Here, a 1 denotes the location of the associated coverage area for establishing the connecting line. Conversely, a 0 indicates the corresponding coverage area where the response is negated, thus disregarding any potential connection path. For example if $n = 5$, a individual can can be $c_1 = [110100]$, this represent that areas $1, 3$ are considered to find the connection to Base station.

4.2.3.2 Genetic operators In this paper, we employ a novel crossover and mutation heuristic strategy along with a potent local search function to seek optimal results.

Crossover : With two random chromosomes from the population, we denoted them as P_1, P_2 . Then we introduce the following heuristic crossover method for generating new chromosomes C from P_1 and P_2.

$$C[i] = \begin{cases} P_1[i], & \text{if } P_1[i] = P_2[i], \\ P_1[i], & \text{if } P_1[i] \neq P_2[i] \\ & \quad \text{and } p < \frac{fitness(P_1)}{fitness(P_1)+fitness(P_2)}, \\ P_2[i], & \text{otherwise} \end{cases} \tag{15}$$

where p is a random number in range $[0, 1]$.

Mutation : We propose a Heuristic Mutation. A chromosome satisfies when it has enough sensors for connectivity. For these chromosomes, we will iterate through points with a value of 0 and change them to 1 with a given probability α (based on experimentation). Each time there's a change, decrease α by an amount of $\frac{1}{2n}$ to ensure there are not too many changes (as excessive changes can lead to violations). Similarly for chromosomes that do not satisfy, we apply the same strategy but instead of changing 0 to 1, we change 1 to 0.

Local search : With the chromosome that has the best fitness P_{best}, we will iterate through all points in P_{best} and replace each value of 0 with 1. If a new chromosome that satisfies the conditions is generated, this will be the new best chromosome.

4.2.3.3 Evaluation The fitness value of a chromosome A is determined according to a special strategy as follows:

$$fitness(A) = \begin{cases} |T|, & \text{if enough sensors for connectivity} \\ \frac{1}{|U|}, & \text{otherwise} \end{cases} \tag{16}$$

with $|T|$ is the number of targets that satisfy K-coverage and K-connectivity and $|U|$ is the number of missing sensors.

In our study, we retain unsatisfactory chromosomes within the population to preserve potentially beneficial genetic material for subsequent generations. Notably, individuals with fewer sensor deficiencies are assigned higher fitness values than those with greater deficiencies. To consistently meet the constraint, individuals of higher quality are assigned elevated fitness values (If chromosome X outperforms chromosome Y, then $fitness(X) > fitness(Y)$).

4.2.3.4 Selection and replacement Starting from a population denoted as P comprising N elements, we will generate a new population, labeled P_{new} also consisting of N chromosomes, employing a specialized heuristic strategy :

Initially, the individuals within P will be arranged in descending order based on their fitness value.

We defined two probabilities p_1, p_2. Where p_1 decides whether we will use crossover or mutation and p_2 to decide whether we will do with the whole population or with some top chromosomes. New chromosomes are created and added to P_{new} until $|P_{new}| = N$.

Next, we merged populations P and P_{new} to form P_{mix}, then sorted P_{mix} in descending order of fitness values. Subsequently, to construct the potential population P_p, we select individuals as follows: Initially, the top c_{top}% elements of P_{mix}, representing the best individuals, are chosen and removed. Next, the remaining aspects of P_{mix} are shuffled, and $c_{routlette}$% elements are selected using the roulette wheel selection method. And then $P = P_p$. After that, choosing randomly c_{loc}% elements in P to undergo local search. This process iterated max_gen times.

Function $bestfitness(P)$ return the individual with the highest fitness value in P.

The entire algorithm is described in **Algorithm 4**.

5 Numerical results

5.1 Parameter setting

Our algorithms are implemented in Python and executed on Visual Studio Code with Intel(R) i5-12500H 3.1GHz CPU, RAM 16GB DDR4 1600MHz.

The parameter is configured for presentation inTable 1.

Algorithm 4: Prim-based Memetic Algorithms (PMA)

```
Input  : $R$ : Set of candidate regions.
         $n_a$ : The number of available sensors.
         $N$ : The number of individuals in a population.
         $max\_gen$ : The number of generation.
         $p_1, p_2$ : Mutation coefficient
Output : The number of targets satisfy both
         $K$-coverage and $K$-connectivity
$P \leftarrow$ Randomly generate N individuals;
$count = 0$
while $count < max\_gen$ do
    $P_{new} = \emptyset$
    while $|P_{new}| < N$ do
        if $p_1 < m$ then
            $x = crossover(P_1, P_2)$ if $p_2 < m_1$
            $y = mutation(P_1)$ if $p_2 < m_2$
            where $P_1, P_2$ is randomly in $P$
        end
        else
            $x = crossover(P_1, P_2)$ if $p_2 < m_1$
            $y = mutation(P_1)$ if $p_2 < m_2$
            where $P_1, P_2$ are randomly selected
              from the top $N_{best}$ elements in $P$.
        end
        $P_{new} \cup x \cup y$
        $p_1, p_2$ is random numbers in range $[0, 1]$
    end
    $P_{mix} = P + P_{new}$
    Calculate fitness value of every individual in
      $P_{mix}$.
    $X$ = Select $c_{top}$% of elements in $P_{mix}$ with the
      highest fitness values.
    $P_{mix} \setminus X$
    $X_1$ = Choose $c_{routlette}$% elements using the
      roulette wheel selection method in $P_{mix}$
    $P = X + X_1$ , Apply local search for randomly
      $c_{loc}$% elements in $P$
    $count = count + 1$
end
$x$ = local search($x$) , $x$in $P$
Return $bestfitness(P)$ ;
```

Table 1: Parameter value for PMA

Parameter	**Value**
Population size (N)	200
N_{best}	50
Number of generations (max_gen)	300
Crossover rate($p_1 = p_2$)	20%
c_{top}	25
$c_{routlette}$	75
c_{loc}	10
α	0.3

5.2 Problem instances

Due to the lack of public research related to this problem, we conducted an experiment on a new dataset consisting of four scenarios for both phases: K-coverage and K-connectivity fromTable 1 . The data set is limited to the 1000×1000 domain. We randomly generate the locations of targets and Base stations in surveillance region A of size $1000 \times 1000(m^2)$ with uniform distribution.

We have 4 scenarios:

scenario 1 The scenario includes 10 instances as given inTable 2; each instance undergoes execution across 10 distinct test sets, followed by averaging, to assess the impact of the number of sensors on solution quality.

scenario 2 The scenario includes 10 instances as given inTable 3; each instance undergoes execution across 10 distinct test sets, followed by averaging, to assess the impact of the number of targets on solution quality.

scenario 3 The scenario includes 5 instances as given inTable 4; each instance undergoes execution across 10 distinct test sets, followed by averaging, to assess the impact of the number of K on solution quality.

scenario 4 The scenario includes 10 instances as given inTable 5; each instance undergoes execution across 10 distinct test sets, followed by averaging, to assess the impact of the number of r on solution quality.

Table 2: Parameter values for test instances in scenario 1

Dataset	n	m	r	K	$A(W \times H)(m^2)$
s1-1	400	150	20	3	1000×1000
s1-2	440				
s1-3	480				
s1-4	520				
s1-5	560				
s1-6	600				
s1-7	640				
s1-8	680				
s1-9	720				
s1-10	760				

Table 3: Parameter values for test instances in scenario 2

Dataset	n	m	r	K	$A(W \times H)(m^2)$
s1-1	400	60	20	3	1000×1000
s2-2		70			
s2-3		80			
s2-4		90			
s2-5		100			
s2-6		110			
s2-7		120			
s2-8		130			
s2-9		140			
s2-10		150			

Table 4: Parameter values for test instances in scenario 3.

Dataset	n	m	r	K	$A(W \times H)(m^2)$
s3-1	400	150	20	1	1000×1000
s3-2				2	
s3-3				3	
s3-4				4	
s3-5				5	

Table 5: Parameter values for test instances in scenario 4

Dataset	n	m	r	K	$A(W \times H)(m^2)$
s4-1	400	150	12	3	1000×1000
s4-2			14		
s4-3			16		
s4-4			18		
s4-5			20		
s4-6			22		
s4-7			24		
s4-8			26		
s4-9			28		
s4-10			30		

5.3 Experiment results

We run 4 scenarios on the dataset and evaluate the obtained results. For this evaluation, we used the variable score $= \frac{E}{n}$, where E represents the number of targets satisfying K coverage and K connectivity, and n denotes the total number of targets. The detailed results are presented below.

In this experiment, domain A is a large region of size 1000 × 1000. Result of this experiment is given in Fig 3, Fig 4, Fig 5 and Fig 6. With the dataset we have, it is clear that there is a significant distance between nodes, which means there are not many overlapping regions. In comparing two base methods, their results exhibit a notable similarity, whereas the proposed method PMA consistently demonstrates superior performance over both base methods.

scenario 1

In this experimental setup, the value of n was incrementally raised from 400 to 760 to investigate the influence of the sensor count on the outcomes generated by three distinct algorithms. The result is shown in Figure 3. The analysis reveals that PMA surpasses the performance of the two base methods. To be precise, PMA exhibits a superiority of 110% over PWS and Greedy algorithms. In expansive spatial contexts, PMA demonstrate enhanced efficacy relative to baseline methods, owing to the integration of heuristic crossover, mutation mechanisms, and extensive local search. As the number of sensors increases, the opportunity for targets to meet both K-coverage and K-connectivity requirements rises significantly

scenario 2

In our experimental setup, we incrementally varied the value of m from 60 to 150 to study how the number of

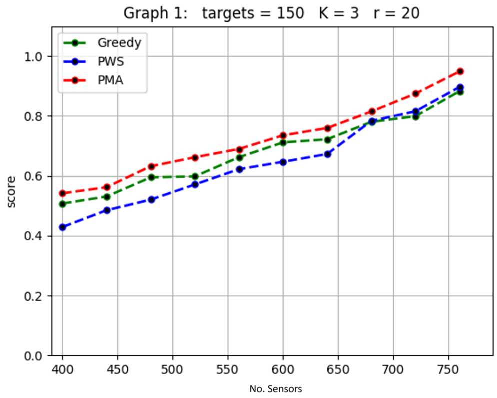


Figure 3: Impact of the number of sensor on PMA, PWS and Greedy

targets influences the outcomes produced by three different algorithms. Figure 4 illustrates the results, showing that PMA outperforms both baseline methods. Specifically, PMA demonstrates a superiority of 113% over the PWS and Greedy algorithms in terms of performance. As the number of targets increases, more sensors are needed to cover and connect all targets in the region. Consequently, the number of targets satisfying the constraints decreases.

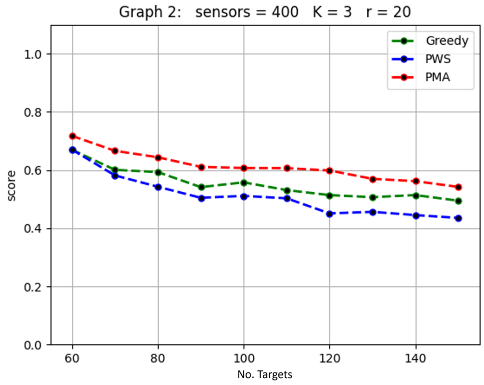


Figure 4: Impact of the number of sensor on PMA, PWS and Greedy

scenario 3

In this experiment, we varied K from 1 to 5 to assess its influence on the outcomes produced by three algorithms. The results depicted in Figure 5 demonstrate that PMA outperforms both baseline methods, showing a performance advantage of 115% over the PWS and Greedy algorithms. To explain why PMA outperforms, as the value of K increases, devising an effective sensor deployment strategy for optimal solutions becomes more challenging. However, PMA maintains its strength through diverse exploration of feasible solution spaces, consistently approaching nearly optimal outcomes.

scenario 4

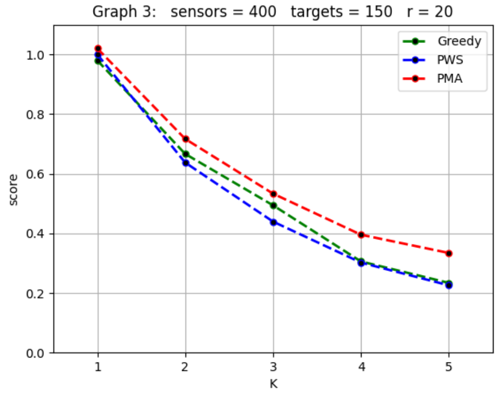


Figure 5: Impact of the number of sensor on PMA, PWS and Greedy

In this experiment, r ranged from 12 to 30 to examine its impact on the outcomes of three algorithms. The findings in Figure 6 indicate that PMA surpasses the performance of the two baseline methods, exhibiting a superiority of 124% over Prim and Greedy algorithms. As r increases, sensor deployment creates more overlapping areas, which reduces the required number of sensors and thereby increases the number of targets meeting both K-coverage and K-connectivity criteria. Furthermore, more significant overlap introduces various deployment strategies, and through its diverse exploration of solution space, PMA demonstrates superior performance compared to baseline methods.

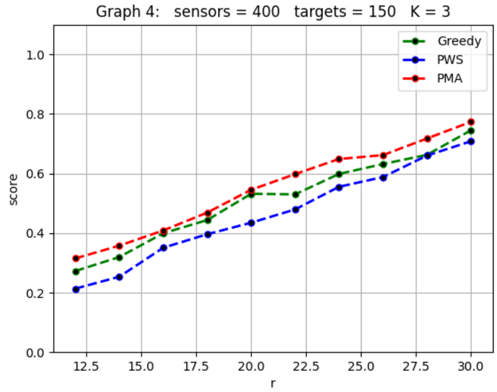


Figure 6: Impact of the number of sensor on PMA, PWS and Greedy

5.4 Real-world dataset

In this section, we utilize a real-world dataset to assess the performance of the PMA method. The original coordinate data were sourced from [9]. This dataset comprises the coordinates of 43 targets, representing railway stations and bus stops near the center of Hanoi, the capital of Vietnam. The base station is positioned at the Vietnam Academy of Agriculture. To facilitate analysis, we normalized the coordinates to a range of $[0, 1000]$.

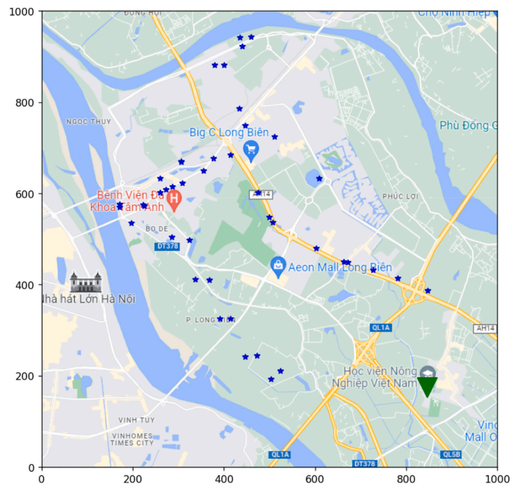


Figure 7: Target locations on the map

The illustration for the target coordinates is depicted in Figure 7. Suppose we aim to monitor these targets from the academy utilizing a WSN. We consider the following two scenarios:

- scenario 1: Let $K = 2$, which means the targets require low resources.
- scenario 2: Let $K = 5$, which means the targets require high resources.

All other parameters remain constant: $r_c = r_s = 60m$.

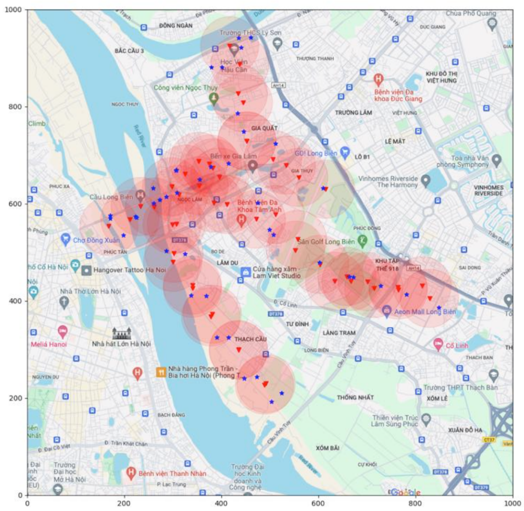


Figure 8: Result of scenario 1

The outcomes of the two scenarios are depicted in Figures 8, and 9. In these figures, blue stars denote targets, while red triangles indicate sensor nodes. In scenario 1, the PMA algorithm requires a total of 50 sensors and operates in 0.033 seconds. Conversely, scenario 2 requires 125 sensors and runs in 0.054 seconds. It is evident that selecting optimal regions in Phase 1 allows PMA to deploy sensors effectively within the intersection areas of the target disks. Furthermore, optimizing connections in Phase 2 helps reduce the consumption of sensor nodes.

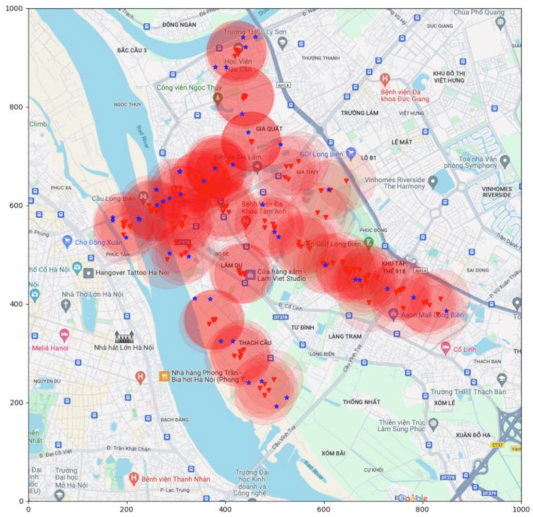


Figure 9: Result of scenario 2

6 Conclusion

This paper presents a model that maximizes the number of targets satisfying K-coverage and K-connectivity with a fixed number of sensors. The problem is addressed in two phases: the first phase optimally places sensors to achieve K-coverage, while the second phase establishes optimal connections to ensure K-connectivity. The Greedy algorithm is proposed to solve the first phase, while a novel method called PMA is employed for the second phase and compared with Prim and Greedy algorithms. Extensive testing across four scenarios reveals that optimizing K-coverage and K-connectivity significantly impact network deployment. The proposed PMA outperforms existing Prim and Greedy methods.

These findings promise future advancements in Wireless Sensor Networks. In the future, we plan to further study this problem and consider more factors such as obstacles, energy efficiency and network lifetime, clustering and routing, deployment in 3D environment

Acknowledgement

This research is funded by Ministry of Education and Training under project number B2024.NHF.01.

References

[1] Nguyen Thi My Binh et al. "An efficient exact method with polynomial time-complexity to achieve k-strong barrier coverage in heterogeneous wireless multimedia sensor networks". In: *Journal of Network and Computer Applications* 231 (2024), p. 103985. URL: https://doi.org/10.1016/j.jnca.2024.103985.

[2] Jonathan L. Bredin et al. "Deploying sensor networks with guaranteed capacity and fault tolerance". In: (2005). DOI: https://doi.org/10.1145/1062689.1062729.

[3] Dina S. Deif and Yasser Gadallah. "Classification of wireless sensor networks deployment tech niques". In: (2013). DOI: https://doi.org/10.1109/SURV.2013.091213.00018.

[4] S.S. Dhillon, K. Chakrabarty, and S.S. Iyengar. "Sensor placement for grid coverage under imprecise detections". In: (2002). DOI: https://doi.org/10.1109/ICIF.2002.1021005.

[5] Liang Du. "Design and Application of Intelligent Building Environment Monitoring System Based on Wireless Sensor Network". In: *Informatica* 48.15 (2024). URL: https://doi.org/10.31449/inf.v48i15.6514.

[6] Suneet Kumar Gupta, Pratyay Kuila, and Prasanta K. Jana. "Genetic algorithm approach for k-coverage and m-connected node placement in target based wireless sensor networks". In: (2016). DOI: https://doi.org/10.1016/j.compeleceng.2015.11.009.

[7] N. T. Hanh et al. "Minimal Node Placement for Ensuring Target Coverage With Network Connectivity and Fault Tolerance Constraints in Wireless Sensor Networks". In: (2019). DOI: http://dx.doi.org/10.1109/CEC.2019.8789961.

[8] Nguyen Thi Hanh et al. "Minimal Relay Node Placement for Ensuring Network Connectivity in Mobile Wireless Sensor Networks". In: (2021). DOI: https://doi.org/10.1109/NCA51143.2020.9306727.

[9] Nguyen Thi Hanh et al. "Node placement optimization under Q-Coverage and Q-Connectivity constraints in wireless sensor networks". In: (2023). DOI: https://doi.org/10.1016/j.jnca.2022.103578.

[10] Nguyen Thi Hanh et al. "Optimizing wireless sensor network lifetime through K-coverage maximization and memetic search". In: (2023). DOI: https://doi.org/10.1016/j.suscom.2023.100905.

[11] Banu Kabakulak. "Sensor and sink placement, scheduling and routing algorithms for connected coverage of wireless sensor networks. Ad Hoc Networks". In: (2019). DOI: https://doi.org/10.1016/j.adhoc.2018.11.005.

[12] Dionisis Kandris et al. "Applications of wireless sensor networks: An up-to-date survey". In: (2020). DOI: https://doi.org/10.3390/asi3010014.

[13] Wei-Chieh Ke, Bing-Hong Liu, and Ming-Jer Tsai. "The critical-square-grid coverage problem in wireless sensor networks is NP-complete". In: (2011). DOI: https://doi.org/10.1016/j.comnet.2011.03.004.

[14] Chien-Chih Liao and Chuan-Kang Ting. "A Novel Integer-Coded Memetic Algorithm for the Set k-Cover Problem in Wireless Sensor Networks". In: (2017). DOI: https://doi.org/10.1109/TCYB.2017.2731598.

[15] S. Pundir adn M. Wazid et al. "Intrusion detection protocols in wireless sensor networks integrated to internet of things deployment: Survey and future challenges". In: (2020). DOI: http://dx.doi.org/10.1109/ACCESS.2019.2962829.

[16] S. Mini, Siba K. Udgata, and Samrat L. Sabat. "Sensor deployment and scheduling for target cover age problem in wireless sensor networks". In: (2013). DOI: https://doi.org/10.1109/JSEN.2013.2286332.

[17] Hemmat Sheikhi and Wafa Barkhoda. "Solving the k-Coverage and m-Connected Problem in Wireless Sensor Networks through the Imperialist Competitive Algorithm". In: (2020). DOI: https://doi.org/10.1142/S0219265920500024.

[18] Piotr Szczytowski, Abdelmajid Khelil, and Neeraj Suri. "DKM: Distributed k-Connectivity Maintenance in Wireless Sensor Networks". In: (2012). DOI: https://doi.org/10.1109/WONS.2012.6152244.

[19] A. Tripathi et al. "Coverage and Connectivity in WSNs: A Survey, Research Issues and Challenges". In: (2018). DOI: https://doi.org/10.1109/ACCESS.2018.2833632.

[20] Bang Wang. "Coverage control in sensor networks". In: (2010). DOI: http://dx.doi.org/10.1007/978-1-84996-059-5.

[21] Bang Wang. "Coverage problems in sensor networks: A survey". In: (2011). DOI: https://doi.org/10.1145/1978802.1978811.

[22] Huping Xu, Jiajun Zhu, and Bang Wang. "On the deployment of a connected sensor network for confident information coverage". In: (2015). DOI: https://doi.org/10.3390/s150511277.

[23] Xiaochun Xu and Sartaj Sahni. “Approximation algorithms for sensor deployment”. In: (2008). DOI: https://doi.org/10.1109/TC.2007.1063.

Dynamic Routing for Large-Scale Mobility On-Demand Transportation Systems

Chijia Liu[1], Alain Quilliot[1], Hélène Toussaint[1], Dominique Feillet[2]
[1]Université Clermont Auvergne, CNRS, UMR 6158 LIMOS, 63178 Aubière, France
[2]Mines Saint-Etienne, Univ Clermont Auvergne, INP Clermont Auvergne, CNRS, UMR 6158 LIMOS, F - 42023 Saint-Etienne, France
E-mail: chijia.liu@uca.fr, alain.quilliot@uca.fr, helene.toussaint@uca.fr,feillet@emse.fr

Keywords: Large-scale dial-a-ride problem, ride-sharing, shared autonomous vehicles

Received: July 23, 2024

We study a prospective large-scale Ride-Sharing Mobility-on-Demand (RSMoD) transportation system. In this system, a fleet of Shared Autonomous Vehicles (SAVs) provides services to a very large number of passengers (up to 300,000). Passenger requests are submitted throughout the service horizon and require an immediate response from the system. We formulate the resulting decision problem as a dynamic dial-a-ride problem (DARP), characterized by its very large size, necessitating well-fitted filtering devices.
We first derive a static version of this dynamic DARP from a statistical knowledge of the requests. Solving this static DARP version allows us to identify reference requests and related routes, which we incorporate into a Guided Insertion Mechanism (GIM). This mechanism aims to expedite the insertion of real dynamic requests. To handle requests that do not fit the GIM learning basis, we complement this mechanism with a Filtering System (FS), creating an algorithmic GIM-FS framework that dynamically routes the SAVs and assigns them to requests. Numerical tests focus on the behavior of both this prospective system RSMoD and the GIM-FS algorithmic scheme. They show that a large-scale RSMoD system involving SAVs is likely to significantly reduce the number of on-route vehicles as well as the energy consumption and that the two-phase GIM-FS approach is more efficient than a baseline method that does not learn from a pre-processed static version of the DARP

Povzetek: Raziskava obravnava dinamično usmerjanje v velikih sistemih prevoza na zahtevo z avtonomnimi vozili. Predlagan je dvofazni algoritem, ki vključuje mehanizem vodenega vstavljanja in filtrirni sistem za učinkovito dodeljevanje vozil zahtevam. Numerični testi kažejo na zmanjšanje števila vozil na cesti in manjšo porabe energije.

1 Introduction

The concept of *Mobility-on-Demand* (MoD) has gained a reputation for being accessible and available to users whenever needed, thus offering more flexible transportation services [4, 40, 29]. On-demand ride services are usually provided by *Transportation Network Companies* (TNCs) such as Uber and Lyft, which communicate with users through digital platforms and mobile applications. Some of these on-demand ride services implement *ride-sharing* or *ride-pooling* [2, 42], allowing passengers with similar itineraries to share vehicles, thus alleviating both carbon emissions and traffic pressure. We refer to these services as *Ride Sharing Mobility-on-Demand* (RSMoD) systems.

Nowadays, most MoD or RSMoD systems aim to provide convenient transportation for residents living in areas with sparse public transportation, such as suburban or rural areas, and for elderly and disabled passengers [36, 35]. These systems are small due to the costs of drivers. However, the current development of autonomous car technology suggests that integrating autonomous vehicles into RSMoD systems could overcome these limitations [46, 16, 37].

Connected autonomous vehicles would drastically decrease the costs of RSMoD systems and make them accessible to all kinds of users. In urban areas, their flexibility might allow them to occupy a significant market share between standard public transportation systems (bus, tram, and underground) and individual cars, whose excessive number and energy consumption are currently major concerns for public authorities. In most European cities, public authorities are promoting both the emergence of RSMoD systems and advances in autonomous vehicle technology [28, 30], with the prospect that a significant portion of individual vehicles might soon be replaced by *Shared Autonomous Vehicles* (SAVs) involved in large-scale RSMoD systems.

In this study, we address some of the decision problems likely to be related to the management of a prospective RSMoD system, relying on SAVs and thus capable of capturing a large part of individual mobility in a medium-sized urban/peri-urban area, such as the one surrounding the town of Clermont-Ferrand in France (population around 300,000). We consider a fleet of mid-sized SAVs controlled by a service-providing company to meet a very large number of daily passenger requests (around 300,000, i.e., around 40% of the individual mobility currently handled by private cars). Installing and managing such a fleet requires handling several decision levels: strategic ones, such as pricing the services, determining the size of the fleet, and establishing the infrastructure of the support transit network (monitoring devices, sensors, protected areas, etc.); operational ones, such as vehicle routing and scheduling, assigning vehicles to requests, maintenance scheduling, etc. In our case, we focus on two tightly connected questions:

- The size of the fleet, which may be considered a decision at the strategic level, which also determines at the operational level the number of requests likely to be rejected.
- The operational dynamic (online) assignment of the large number of requests to the vehicles, and the dynamic routing and scheduling of the vehicles.

While addressing these two questions, we follow a two-fold purpose: Designing efficient algorithms for the real-time management of large sets of requests and vehicles, and the minimization of both the running costs and the number of unsatisfied requests; observing the behavior of the RSMoD system itself, including its size, costs, and its potential positive impacts on urban mobility.

To achieve this purpose, we presuppose the availability of a suitable statistical representation of the mobility demand. This initial statistical knowledge allows us to solve a virtual static version of our dynamic *dial-a-ride problem* (DARP) and use the resulting virtual collection of routes as a learning basis for fast decision-making. Our main contributions are as follows:

- We design a two-phase algorithm that dynamically inserts the requests into the routes followed by the vehicles while minimizing the running costs. This algorithm relies on:
  - A *Guided Insertion Mechanism* (GIM) that takes advantage of the large-scale feature of the system and works as a learning device. It derives high-quality reference routes from the resolution of a static virtual version of the DARP, and uses them to insert real requests r into real vehicles v;
  - A *Filtering System* (FS), that applies to a given request r in case the GIM process fails to efficiently assign a vehicle to r. FS reduces the search space by filtering the candidate insertion parameters, namely the vehicles and their insertion positions within the vehicle routes.
- We conduct numerical experiments that assert the ability of this algorithm to significantly reduce computational times compared to the baseline approach (which does not involve the GIM mechanism), while maintaining the quality of the resulting solution.
- Furthermore, according to experimental results, an RSMoD system relying on SAVs might reduce the number of on-route vehicles by around 99% compared to a private system where all 300,000 requests would be serviced by private vehicles. These experiments also reveal that such an RSMoD system would manage significantly more dynamic requests in a shorter total drive time than a taxi-like MoD system involving the same number of vehicles and lacking a ride-sharing feature

To our knowledge, this is the first study that explicitly links the resolution of dynamic DARP to the use of statistical vehicle travel patterns.

The paper is organized as follows: Section 2 is devoted to the state of the art. Section 3 describes how our prospective RSMoD system works. We present in Section 4 a static version of the resulting large-scale DARP and describe how it can be handled through a best-fit insertion heuristic. The core of our contribution is in Section 5, which describes the two-phase GIM-FS algorithm for the handling of the dynamic version of the large-scale DARP. Section 6 presents numerical experiments. We briefly conclude with a discussion about open questions.

2 Related works

2.1 MoD systems

Together with the significant increase in smartphone ownership and the growing demand for flexible mobility, *Mobility-on-Demand* (MoD) transportation systems have emerged in recent years, greatly facilitating travel for urban and rural residents. MoD systems manage requests through digital platforms to customize efficient travel routes. Since the use of *Shared Autonomous Vehicles* (SAVs) decreases the running costs of MoD systems and opens the way to large-scale MoD systems, academics are increasingly interested in studying these systems [13, 1]. Research in this area generally falls into the following two categories.

The first category contains studies focusing on the systems themselves, including their different configurations, deployment viability, and potential advantages. For example, in [15], the authors consider a dynamic RSMoD system, where a fleet of SAVs is employed to cater for $54,324$ passenger requests during a service horizon of 24 hours in the city of Austin. They demonstrate the ability of the ride-sharing feature to reduce SAVs' total travel time. In [26], the author focuses on the congestion issue. He sets up a Linear Programming model and runs experiments that reveal that, due to the empty repositioning trips performed by the SAVs, congestion during peak hours may increase. In [28], the authors introduce a simulation framework to analyze the behavior of an MoD system with Electric SAVs

(SAEVs). They show that optimizing the number of charging stations and reducing charging times are crucial for the efficiency of the SAEV fleet.

The second category contains studies on how requests and vehicles might be efficiently managed. It puts algorithm design at the core of the research: passenger requests are submitted online, hoping for almost immediate feedback. Deciding on the requests means efficiently assigning them to vehicles and accordingly routing and scheduling the vehicles. This defines the dynamic version of the *Dial-a-Ride* Problem (DARP) (see [38]). The case when the number of requests is high (large-scale MoD systems) raises the issue of filtering, concerning the introduction of filtering devices that will guide the decision-makers, for any given request, towards the best-fitted vehicles. For example, authors in [43] introduce a *dual-sided taxi searching* approach to efficiently filter candidate vehicles inside a taxi system that dynamically handles over 330,000 requests. In [3], the authors implement a similar idea to match requests and vehicles in a ride-sharing system with thousands of requests.

2.2 Static DARP

Addressing the dynamic version of the problem requires the ability to efficiently solve its static version, which is characterized by the fact that all requests are known in advance and there are no restrictions regarding computational times. Reviewing the literature reveals that most studies on the DARP have been conducted concerning its static version [12].

The static DARP involves a set of passenger requests, each typically containing a pick-up location (origin) and a drop-off location (destination). Additionally, certain constraints regarding service quality are imposed, such as time windows for pick-up or drop-off services, or a maximum ride time. The goal is to organize the routes of a fleet of vehicles so that certain objectives are optimized, such as minimizing total travel time or maximizing service quality. Meanwhile, routes should satisfy constraints such as vehicle capacity, maximum route duration, and those imposed by the requests [18].

For very small-sized DARPs, authors propose exact algorithms. Studies [38, 39] introduce dynamic programming algorithms to solve specific instances of DARPs. Exact algorithms based on Branch-and-Bound, Branch-and-Cut, and Branch-and-Price [10, 7] are the most common methods used to solve the DARP. For example, in [32], the authors propose a Brand-and-Price method to solve a DARP with *Split Requests and Profit* (DARPSRP) with an objective of maximizing the total profit. Requests involving several passengers can be split to be served separately if it is of benefit to do so. Different instances with up to 4 vehicles and 40 requests, or 5 vehicles and 20 requests can be solved to optimality.

In real-life situations, the number of requests to be processed usually forbids exact methods and leads to the design of heuristics. In [11], authors design a tabu search heuristic to solve the DARP with time windows and deal with instances with up to 295 requests with 20 vehicles. Solving the largest instances necessitates over 200 minutes of CPU time. Their work has inspired other tabu search-based methods [6, 31]. Other types of metaheuristics are also widely studied, such as Variable Neighborhood Search (VNS) [33], Large Neighborhood Search (LNS) [21, 34] and Adaptive Large Neighborhood Search (ALNS) [47, 5].

Insertion heuristics [22], which gradually constructs feasible solutions, may be adapted in a natural way to dynamic contexts and mid-to-large-sized problems [8, 41, 20]. Among them, the *best-fit insertion* heuristics [44] appear as the most efficient ones. In order to deal with very large instances, [27] introduces a filtering system based on a spatial-temporal decomposition, which, given a target request, not only selects candidate vehicles but also provides the candidate insertion positions within the route of these vehicles where the request can be inserted. Part of this filtering system will be used in this paper.

2.3 Learning travel patterns

In large-scale systems, most requests correspond to travelers who make similar trips from one day to another, and thus adhere to some statistical patterns that one may try to capture through statistical analysis or machine learning [17, 49]. For example, in [48], the authors propose a deep learning-based approach, *ST-ResNet*, which models the temporal closeness and properties of crowd traffic and predicts the inflow and outflow of crowds in every region. In [45], authors introduce a learning model based on a convolution network to predict hourly Origin-Destination Matrices in the Beijing urban area.

On the other hand, due to the similarity in passenger requests, the trajectories of vehicles supporting these requests are also likely to exhibit some repetitive or similar travel patterns [25, 19]. In [24], the authors construct a sequence of dynamic graphs from the aggregation of trajectory data and apply graph mining algorithms to analyze the spatial and temporal travel patterns in the transit network. In [23], a trajectory clustering method based on identifying the *Longest Common Sequence* of each trajectory is presented, which facilitates traffic flow pattern analysis.

3 A target prospective RSMoD system

The transit network The key underlying infrastructure of our system is a transit network, represented here as an oriented graph $\mathcal{G} = (\mathcal{N}^\dagger, \mathcal{A})$, where $\mathcal{A}$ is the set of road links (elementary arcs) and $\mathcal{N}^\dagger$ is the set of road junctions (nodes). We denote by $\mathcal{N} \subset \mathcal{N}^\dagger$ the nodes where a vehicle involved in the system may pick up or drop some passengers. We suppose, for the sake of simplicity, that the travel time on any arc $a \in \mathcal{A}$ for any vehicle v is constant and that the travel times from any node $n_1 \in \mathcal{N}^\dagger$ to any node

$n_2 \in \mathcal{N}^\dagger$ are pre-calculated. We denote by $t(n_1, n_2)$ this quantity. In addition, we only consider one vehicle depot $n_0 \in \mathcal{N}^\dagger$ in the network. Throughout this paper, we refer to this network as representative of the transit network of a mid-sized urban area with about 500,000 inhabitants, involving between 5,000 and 15,000 nodes.

The main players Three agents should be identified as the main players of our RSMoD system: the central operator, the SAV fleet, and the passengers. Passengers submit their travel requests online to the central operator via a digital terminal. The central operator then assigns requests to SAVs and plans the routes for these vehicles. Finally, each SAV follows its route to provide services to passengers. More precisely:

- **The Passenger Requests**: A passenger request r, submitted at time $tSub^r$ is defined by: a load (number of passengers) q^r; a pick-up service O^r, containing a pick-up node $o^r \in \mathcal{N}$ and a pick-up time window $[e_{o^r}, l_{o^r}]$, where e_{o^r} and l_{o^r} represent respectively the earliest and latest pick-up time for r; a drop-off service D^r, containing a drop-off location $d^r \in \mathcal{N}$, and a maximum ride-time T^r. We easily deduce a drop-off time window $[e_{d^r}, l_{d^r}]$, where

  $$e_{d^r} = e_{o^r} + t(o^r, d^r);$$

  and

  $$l_{d^r} = l_{o^r} + T^r.$$

  We assume that the service times at pick-up and drop-off locations can be neglected.

  The Large Scale Feature: We assume that this prospective RSMoD system is likely to replace around 40% of the current mobility currently reliant on individual cars. So we deal here with a number of requests between 2.10^5 and 4.10^5.

  Statistical Behavior of the Requests: Requests are submitted online throughout the time horizon and they may differ from one day to another. However, we suppose that they follow some probabilistic distribution that can be reproduced through simulation.

- **The SAV Vehicles**: The SAV fleet involves V identical autonomous electric vehicles, all with capacity Q. The running times of the vehicles along the arcs of the transit network come with the network. Each request can be served by at most one vehicle, and no transfers between vehicles are allowed. Once again, we do not address the energy issue and assume that the vehicles are able to run without recharging throughout the time horizon.

- **The Central Operator**: The central operator is the *decider* of the system. It collects the requests and assigns them to the vehicles (or possibly rejects them), while simultaneously routing, scheduling, and controlling the vehicles. It processes the requests by "packages". More precisely, $[0, T]$ being the planning horizon, it divides it into E intervals $[t_e, t_e + I_E]$ of duration I_E, resulting in a set of *decision epochs* $\mathcal{E} = \{0, 1, \ldots, e, \ldots, E-1\}$. At each decision epoch $e \in \mathcal{E}$, it collects a set $\mathcal{R}^{e+1}$ of requests that will be processed during epoch $e+1$ according to the following decision scheme ($\mathcal{R}^0$ means the request that have been submitted in advance) :

  Step 1 At the beginning of epoch e, the central operator proceeds the requests of $\mathcal{R}^e$ during $[t_e, t_e+\eta]$, where $\eta < I_E$ is a parameter that bounds the computational time allowed to this part of the process. We note that η must be small enough to leave enough time for time-consuming steps 2 and 3.

  Step 2 Then it informs all passengers about the way their requests have been handled and asks them to confirm.

  Step 3 According to this it communicates with the vehicles and updates their routes and tentative schedules.

  The **primary objective** of the central operator is to minimize the number of *rejected* requests, that cannot be serviced by the vehicle fleet. The **secondary objective** is the minimization of an operational cost, reduced here to the Vehicles' total Travel Time, denoted by VTT.

Remark 1: In this work, we care neither about stochastic events (passenger no-shows, traffic accidents, etc.), nor about the communications processes that take place between the passengers, the central operator and the vehicles, nor, as previously mentioned, about the energetic issue. Yet it will be indispensable to address issues while designing the processes aiming at the control of any real-life *RSMoD* system involving SAVs.

Remark 2: It is worth noting that treating each decision epoch as having equal duration I_E is a simplification. In real-life dynamic systems, the frequency of initiating routing and replanning procedures usually varies according to how quickly passengers should receive a response and the quality of the routing. However, this aspect is not the main concern of this study.

Our Target Problem According to the above description, our challenge becomes the design of algorithmic tools for the handling, at any epoch e, of the requests r of $\mathcal{R}^e$. In order to deal with this dynamic large-scale DARP, we first solve a *static* (where all requests are known in advance) *virtual* version of our DARP. Next, we use the virtual solution obtained this way as a reference assignment and routing strategy, and handle the real requests while trying to mimic this strategy. In case we fail to insert a request r

in the current collection Θ of vehicle routes, we rely on a complementary filtering device to compute the well-fitted insertion parameters of the insertion of r into Θ.

4 Dealing With a virtual static large-scale dial-a-ride problem

Formally, the main components of the *static* DARP are adapted from what has just been previously described:

- The Transit Network.
- The Vehicle Fleet: Vehicles are the same as described above, but their number is not known in advance. Consequently, this Static DARP also aims to help decide the size of the vehicle fleet. In addition, it could potentially be utilized to identify the part of the transit network that should support some kind of monitoring infrastructure. Nevertheless, this specific issue is not addressed in this study.
- The requests: We consider a large-scale set $\mathcal{R}$ of known-in-advance *virtual* requests. They are generated according to the probabilistic distribution of real requests. Being static, these requests are not characterized by submission times.

The goal is to build a collection $\Theta = \{\theta^v, v = 1, \ldots, V\}$ of routes, assign every request to some route, and schedule the vehicles along those routes following the objective below:

- (Primary Objective) Minimize V.
- (Secondary Objective) Minimize VTT.

This performance criterion is based on a lexicographic ordering of the pairs (V, VTT): (V_1, VTT_1) is better than (V_2, VTT_2), if and only if:

- $V_1 < V_2$, or
- $V_1 = V_2$ and $VTT_1 < VTT_2$.

Notice that our two criteria are in some way antagonistic. We denote the resulting problem by *S_DARP*.

4.1 Encoding a route: notion of key point

The large scale of our problem leads us to rely on a non-standard representation of the routes: For any vehicle v, its route θ^v is a list $\theta^v = \{P_0, \ldots, P_i, \ldots, P_{M(v)-1}\}$ of *key points*, where a *key point* is defined by all the drop-off and pick-up services to be simultaneously performed by v at a specific node $n_P \in \mathcal{N}$. Proceeding this way will drastically diminish the length of the routes with respect to the standard encoding and make it easier to perform both the search of well-fitted insertion parameters for the requests into the vehicles and the feasibility tests related to such an insertion. More specifically, each key point $P \in \theta^v$ specifies: (a) a service location $n_P \in \mathcal{N}$; (b) an arrival time window $[e^a_P, l^a_P]$, where e^a_P and l^a_P indicate the earliest and latest possible arrival times at node n_P, respectively, together with the set R^-_P of requests scheduled to get out of the vehicle at P (drop-off services). (c) a departure time window $[e^d_P, l^d_P]$, where e^d_P and l^d_P represent the earliest and latest departure times from n_P respectively, together with the set R^+_P of requests scheduled to get in (pick up services) the vehicle at P; (d) the load of vehicle q_P before leaving n_P.

The first and last points in θ^v, i.e., P_0 and P_{M-1}, represent the fact that every vehicle should leave the depot to start providing service, and return to the depot after having finished all services by the end of time horizon T. We also denote them by $P_0 = D_1$ and $P_{M-1} = D_2$. Every route is initialized as $\{D_1, D_2\}$, with: $n_{D_1} = n_{D_2} = n_0$, $e^a_{D_1} = e^d_{D_1} = e^a_{D_2} = e^d_{D_2} = 0$, $l^a_{D_1} = l^d_{D_1} = l^a_{D_2} = l^d_{D_2} = T$, $R^+_{D_1} = R^-_{D_1} = R^+_{D_2} = R^-_{D_2} = \varnothing$, and $q_{D_1} = q_{D_2} = 0$. For the sake of simplicity, given two key points $P_i \in \theta^v$ and $P_j \in \theta^v$, we use $t(P_i, P_j)$ to denote the travel time from node n_{P_i} to node n_{P_j}. For any request r, we denote by $P(O^r)$ and $P(D^r)$ the key points supporting O^r and D^r, respectively.

Travel Time and Feasibility of a Route

According to this specific encoding, the travel time of a route $\theta^v = \{P_0, \ldots, P_i, \ldots, P_{M-1}\}$ is equal to $\sum_{i \leq M-2} t(P_i, P_{i+1})$. Besides, θ^v is *feasible* if it satisfies the following constraints:

- For any request r inserted at θ^v, the pick up service O^r should take place before the drop-off service D^r: if $P(O^r) = P_i$ and $P(D^r) = P_j$, then $i < j$;
- For any i, the load inside the vehicle must not exceed its capacity:
$$q_{P_i} \leqslant Q; \tag{1}$$
- For any i, v should not leave P_i before arriving at it:
$$e^a_{P_i} \leqslant e^d_{P_i}, \tag{2}$$
$$l^a_{P_i} \leqslant l^d_{P_i}; \tag{3}$$
- For any $i < M - 1$, the arrival time at P_{i+1} should be consistent with the departure time from P_i:
$$e^d_{P_i} + t(P_i, P_{i+1}) \leqslant e^a_{P_{i+1}}, \tag{4}$$
$$l^d_{P_i} + t(P_i, P_{i+1}) \leqslant l^a_{P_{i+1}}; \tag{5}$$
- For any request r inserted at θ^v, its maximum ride-time should be bounded:
$$e^a_{P(D^r)} - \min(e^d_{P(O^r)}, l_{o^r}) \leqslant T^r, \tag{6}$$
$$l^a_{P(D^r)} - \min(l^d_{P(O^r)}, l_{o^r}) \leqslant T^r, \tag{7}$$

where the term $\min(e^d_{P(O^r)}, l_{o^r})$ represents the actual earliest in-vehicle time of r. Generally, passengers are expected to board the vehicle no earlier than the earliest departure time from O^r, denoted $e^d_{P(O^r)}$. However, if $e^d_{P(O^r)}$ is later than r's latest permissible in-vehicle time, l_{o^r}, then r may board at l_{o^r} and wait until the vehicle is ready. This situation could arise, for example, if the vehicle must wait for other requests to get onboard at the same key point before it can depart. The same reasoning applies to how we represent the actual latest pick-up time of r, $\min(l^d_{P(O^r)}, l_{o^r})$.

- For any r getting in v at P_i, v must be able to arrive at n_{P_i} before the latest pick-up time of r:

$$\forall r \in \mathcal{R}^+_{P_i},\ e^a_{P_i} \leqslant l_{o^r}; \tag{8}$$

- For any i, no related time windows are empty:

$$e^a_{P_i} \leqslant l^a_{P_i}, \tag{9}$$

$$e^d_{P_i} \leqslant l^d_{P_i}. \tag{10}$$

4.2 A best-fit insertion heuristic

In the static context, we are given *a-priori* unlimited time to solve *S_DARP*. Yet both the problem's very large size and its complexity forbid exact methods. So we handle *S_DARP* through a *greedy Best-Fit insertion heuristic* (BF). This greedy heuristic might be augmented with local search devices and meta-heuristic schemes. However, it is not the focus of this study, and the performance of the *best-fit* insertion heuristic is enough concerning our purpose [20].

4.2.1 The insertion procedures

Since we rely here on a specific encoding of the routes, we need to explain the way we adapt the standard insertion procedures to this encoding: Inserting a request r in a route θ^v according to insertion parameters (v, P_o, P_d) usually means inserting O^r between P_o and its successor in θ^v and D^r between P_d and its successor in θ^v. Here, the fact that a key point refers to several pick-up and drop-off services leads us to consider several cases (for the sake of simplicity, we only consider O^r and P_o):

1. If $o^r \neq n_{P_o}$, then inserting O^r means creating a new key point. There are two ways to insert this new key point into θ^v. The first one (*standard* one, with a parameter $\epsilon_o = 0$) means inserting it between P_o and its successor in θ^v; The second one (*split* one, with a parameter $\epsilon_o = 1$) means splitting the current key point P_o into 2 key points P_o^{arr} and P_o^{dep}, that respectively represents the drop-off and pick-up services performed by the vehicle at n_{P_o}, and inserting the new key point between P_o^{arr} and P_o^{dep}. In practice, this *split* insertion means that v arrives at n_{P_o}, performs its drop-off service, moves to O^r to pick up r and finally comes back to n_{P_o} to perform the pick-up part of its service.
2. If $o^r = n_{P_o}$, then no additional key point is created, and the pick-up service O^r is aggregated to the pick-up service related to P_o, whose time window is impacted.

Consequently, the insertion parameters of the insertion of r into current route collection Θ define a 5-tuple $(v, P_o, \epsilon_o, P_d, \epsilon_d)$.

4.2.2 A best-fit insertion heuristic

According to this, we adapt the classic greedy insertion heuristic framework presented in [22]:

Step 1: Sort $\mathcal{R}$ according to the *ST-Cluster strategy*: Divide the transit network into small zones and the time horizon $[0, T]$ into small periods and cluster the requests according to their pick-up and drop-off zones and their earliest pick-up time periods (see [27] for details). Then handle the clusters according to increasing periods, while randomly selecting the requests belonging to the same cluster.

Step 2: For each request $r \in \mathcal{R}$, sorted according to Step 1:

Step 2.1: For any *insertion parameter* 5-tuple $(v, P_o, \epsilon_o, P_d, \epsilon_d)$, where $v \in \mathcal{V}$ is an activated vehicle, P_o located before P_d in θ^v, check the feasibility of the insertion of O^r according to P_o and ϵ_o, and D^r according to P_d and ϵ_d.

Step 2.2: If at least one insertion is feasible, keep the *best-fitted* one that minimizes detour made by v, and insert r; otherwise, activate a new vehicle v and insert r into the related trivial route.

4.3 Performing step 2.1: checking the feasibility of an insertion

We briefly provide here some insight into Step 2.1. For the sake of simplicity, we restrict ourselves to the case $(\epsilon_o = 0, \epsilon_d = 0)$, which means to the case of *standard* insertions.

4.3.1 A cascade strategy

We proceed according to a *cascade* strategy: We first perform a set of *fast* tests. In case of success, we continue to perform a more time-consuming constraint propagation process that checks the mathematical feasibility of the target insertion. This cascade strategy comes as follows:

- First, we test the insertion feasibility of O^r:
  - If $o^r = n_{P_o}$, then O^r is directly aggregated at P_o, and $P(O^r) = P_o$. We increase the load q_{P_o} by q^r and add r into the set $\mathcal{R}^+_{P_o}$. In addition, we update the earliest departure time $e^d_{P_o}$:

$$e^d_{P_o} \leftarrow \max(e^d_{P_o}, e_{o^r}). \tag{11}$$

The other timestamps of the passage time windows are not directly impacted.

- If $o^r \neq n_{P_o}$, then $P(O^r)$ is a new key point to be inserted between P_o and P_{o+1}, with $n_{P(O^r)} = o^r$, $\mathcal{R}^+_{P(O^r)} = \{r\}$ and $q_{P(O^r)} = q_{P_o} + q^r$. The earliest arrival time and the latest departure time of $P(O^r)$ are computed:

$$e^a_{P(O^r)} = e^d_{P_o} + t(P_o, P(O^r)) \tag{12}$$

$$l^d_{P(O^r)} = l^a_{P_{o+1}} - t(P(O^r), P_{o+1}) \tag{13}$$

- Then, we test the insertion feasibility of D^r:
  - Proceed with D^r and P_d as with O^r and P_o;
  - Deal with the maximum ride time T^r as follows: Two key points $P(O^r)$ and $P(D^r)$ are now associated with O^r and D^r. Depending on the case, they have just been created or they were previously existing. In any case, we may denote by $[e^d_{P(O^r)}, l^d_{P(O^r)}]$ the departure time window of $P(O^r)$ and by $[e^a_{P(D^r)}, l^a_{P(D^r)}]$ the arrival time window of $P(D^r)$. Then we update these time windows:

$$e^d_{P(O^r)} \leftarrow \max(e^d_{P(O^r)}, e^a_{P(D^r)} - T^r) \tag{14}$$

$$l^a_{P(D^r)} \leftarrow \min(l^a_{P(D^r)}, l^d_{P(O^r)} + T^r) \tag{15}$$

- Increase by q^r the load of all key points between $P(O^r)$ and $P(D^r)$ and check that the capacity constraint (1) is not violated.
- Apply the *constraint propagation procedure* and accordingly adjust the time windows along θ^v.

4.3.2 The constraint propagation procedure

The *constraint propagation procedure* [14, 27], checks the consistency of current time windows along the route θ^v with the temporal constraints induced by the *S_DARP* and accordingly updates the influenced time windows. This procedure is the most time-consuming component of our *cascade* strategy, with a complexity $O(M^2)$, where M is the number of key points in θ^v.

Remind that a route is feasible if constraints (2) to (10) in Section 4.1 are satisfied. Constraint propagation means that the violation of such a constraint will trigger a *constraint propagation rule*:

- If (2) is violated, then $e^d_{P_i} \leftarrow e^a_{P_i}$;
  if (3) is violated, then $l^a_{P_i} \leftarrow l^d_{P_i}$.
- If (4) is violated, then $e^a_{P_{i+1}} \leftarrow e^a_{P_i} + t(P_i, P_{i+1})$;
  if (5) is violated, then $l^d_{P_{i-1}} \leftarrow l^a_{P_i} - t(P_{i-1}, P_i)$.
- If (6) is violated, then for any $r \in \mathcal{R}^-_{P_i}$, delay the earliest departing time at its destination: $e^d_{P(O^r)} \leftarrow e^a_{P_i} - T^r$;
  if (7) is violated, then for any $r \in \mathcal{R}^+_{P_i}$, decrease the latest arrival time at its destination: $l^a_{P(D^r)} \leftarrow l^d_{P_i} + T^r$.
- If any of constraints (8) to (10) is violated, then return a *Fail* signal.

Starting from the initial triggers induced by the first updates at $P(O^r)$ and $P(D^r)$ ((11) to (15)), the procedure propagates the temporal *S_DARP* constraints all along θ^v while updating the related time windows according to the above rules. The procedure ends when no new trigger requires activation or when a *Fail* signal is emitted. In such a case, the current insertion parameter 5-tuple $(v, P_o, \epsilon_o, P_d, \epsilon_d)$ is discarded as infeasible.

5 Dealing with the original dynamic large-scale dial-a-ride problem

We may now come back to our original dynamic large-scale DARP as defined in Section 3. We denote the problem by D_DARP. We suppose here that the fleet size V is fixed, and that the requests are dynamically submitted to the system along the time horizon $[0, T]$, which is divided into E decision epochs. Remind that $\{\mathcal{R}^e,\ e \in \mathcal{E}\}$ denotes the set of requests submitted during epoch $e-1$. It is not known in advance and differs from one day to another. As in *S_DARP*, our goal is to assign the requests to the vehicles and route these vehicles. But our lexicographic performance criterion becomes:

- (Primary Objective) Minimizing the number of rejected requests.
- (Secondary Objective) Minimizing VTT.

We must take into account the *dynamic* constraint that keeps us from processing a request before it is submitted. As mentioned in Section 3, we apply the following decision framework:

1. Initialize the route collection Θ by processing the set R^0 of requests that have been submitted before the beginning of the process. During all the decision processes, we use the same key point-based route encoding as in Section 4.
2. For $e = 0, \ldots, E-1$ do
   (a) ***D_DARP***($\mathcal{R}^e$) **instruction**: Process the requests $r \in \mathcal{R}^e$ during the time interval $[t_e, t_e + \eta]$;
   (b) **Dispatch** the decision towards the vehicles and the users during the time interval $[t_e + \eta, t_{e+1}]$. Update the current state of the system while making the vehicles achieve their duty at epoch e while following the earliest possible time according to current time windows.

The rest of this section is devoted to the algorithmic part of this decision framework, i.e., the *D_DARP*($\mathcal{R}^e$) instruction.

5.1 Two-phase algorithmic scheme for *D_DARP*($\mathcal{R}^e$)

To address the dynamic version of the DARP, we first employ *preprocessing* that leverages our statistical knowledge of the request distribution and the resolution to the static problem introduced in Section 4. The preprocessing helps to obtain an estimation of the optimal fleet size, a collection of reference (virtual) requests, and a collection of *travel patterns*. These elements form the basis of the *Guided Insertion Mechanism*, which will be discussed later. Then we perform the *D_DARP*($\mathcal{R}^e$) instruction by applying the following two-phase framework, termed *GIM-FS*:

- **Phase 1: Guided Insertion Mechanism** Each request $r \in \mathcal{R}^e$ is processed by a *Guided Insertion Mechanism* (GIM), which tries almost immediate insertions of r into the current route collection Θ under the guidance of insertion patterns deduced from the reference requests and preprocessing. If the insertion of r is feasible according to GIM, then we keep the best-fitted one and insert r;

- **Phase 2: Filtering System** For any request r not inserted by GIM, we invoke a specific *Filtering System* (FS) to fast identify the best candidate insertion parameters. If there are parameters that allow a feasible insertion of r then we choose the best-fitted one; otherwise, we reject r.

Let us now describe the preprocessing process, which serves as the basis for GIM, before delving into the details of both GIM and FS.

5.2 Preprocessing

Urban short-distance trips primarily consist of commuting, shopping, and other personal errands, making up the majority of daily travel purposes [9]. Therefore, large sets of requests should present some similar patterns. Consequently, if all requests are served by a *RSMoD* system, daily vehicle travel patterns should also be characterized by some similar patterns. GIM is based on the same idea that drives statistical learning: if two large request sets $\bar{\mathcal{R}}$ and $\mathcal{R}$ are similar from a statistical point of view, then vehicle routes $\bar{\Theta}$ designed for $\bar{\mathcal{R}}$ and vehicle routes Θ designed for $\mathcal{R}$ should also be similar.

5.2.1 Solving reference static problem

According to the above motivation, we start by addressing a *reference* static problem $S\_DARP(\bar{\mathcal{R}})$ as introduced in Section 4, where $\bar{\mathcal{R}}$ is a *virtual reference* request set generated so that it is consistent with our statistical knowledge of the requests. As much time as necessary is used to solve $S\_DARP(\bar{\mathcal{R}})$ to get a good *reference* route set $\bar{\Theta} = \{\bar{\theta}^v, v \in 1, \dots, V\}$. At this time, a point needs to be discussed. We do not know *a-priori* the fleet size V required to efficiently handle our original problem $D\_DARP(\mathcal{R})$. Experience shows that the way requests' submission times evolve from one day to another is far more volatile than the other characteristics of the requests. Therefore, we intuitively guess that, for the same number of requests, the *online* feature will make the fleet size larger than that obtained through the resolution of the corresponding static DARP. Consequently, we consider the size of $\bar{\mathcal{R}}$ as a parameter and generate $\bar{\mathcal{R}}$ that reproduces the spatio-temporal distribution of the real requests, so that the resulting fleet size V is close to the optimal size required by the real system. This requires some tuning processes that will be discussed in Section 6.2 presenting numerical experiments.

5.2.2 Deriving travel patterns

Next, from the reference route set $\bar{\Theta}$, we derive a set of *travel patterns* Γ. Specifically, for each vehicle $v \in \mathcal{V}$, a *travel pattern* $\gamma^v \in \Gamma$ can be derived from its reference route $\bar{\theta}^v = \{\bar{P}_0, \dots, \bar{P}_{M-1}\}$. We define the *travel pattern* γ^v as a list of *guiding points*, where each guiding point $G \in \gamma^v$ is a representation of a *key point cluster* in $\bar{\theta}^v$.

Definition 5.1 (*Key Point Cluster*). A sub-route $\{P_j, P_{j+1}, \dots, P_{j+m}\}$ of route $\theta^v = \{P_0, \dots, P_i, \dots, P_{M-1}\}$ with M key points forms a key point cluster, if and only if:

$$t(P_k, P_{k+1}) \leqslant \alpha, \ \text{ for } \ j \leqslant k \leqslant j+m-1$$

$$t(P_{j-1}, P_j) > \alpha, \ \text{ if } j \neq 0$$

$$t(P_{j+m}, P_{j+m+1}) > \alpha, \ \text{ if } \ j+m \neq M-1$$

where the parameter α is a temporal threshold defining the maximum travel time (distance) allowed for two consecutive key points belonging to the same cluster.

A guiding point G representing the cluster $[\bar{P}_j, \bar{P}_{j+1}, \dots, \bar{P}_{j+m}]$ is associated with a temporal neighborhood $[t_G^{min}, t_G^{max}]$, with $t_G^{min} = \min_{\bar{P} \in G} e_{\bar{P}}^a$, and $t_G^{max} = \max_{\bar{P} \in G} l_{\bar{P}}^d$. We note that $\bar{D}_1$ and $\bar{D}_2$, which correspond to the initial and terminal reference key points $\bar{P}_0$ and $\bar{P}_{M-1}$ respectively, are excluded from the travel pattern construction because no insertion happens at these points. For any reference key point $\bar{P}$ other than $\bar{D}_1$ and $\bar{D}_2$, its associated guiding point is denoted by $G(\bar{P})$.

As shown in Figure 1, given the reference route $\bar{\theta}^v$, the corresponding travel pattern is γ^v. $\{\bar{P}_2, \bar{P}_3, \bar{P}_4\}$, $\{\bar{P}_6, \bar{P}_7, \bar{P}_8, \bar{P}_9\}$ and $\{\bar{P}_{11}, \bar{P}_{12}\}$ are three key point clusters containing multiple key points. For example, the associated guiding point of $\bar{P}_3$ is $G(\bar{P}_3) = G_1$.

5.3 Phase 1: the guided insertion mechanism

The idea of GIM when solving the real dynamic problem is to encourage every SAV v to follow its pre-defined travel pattern γ^v. Let us consider a request $r \in \mathcal{R}^e$ to be inserted. In the following, we describe how GIM proceeds r.

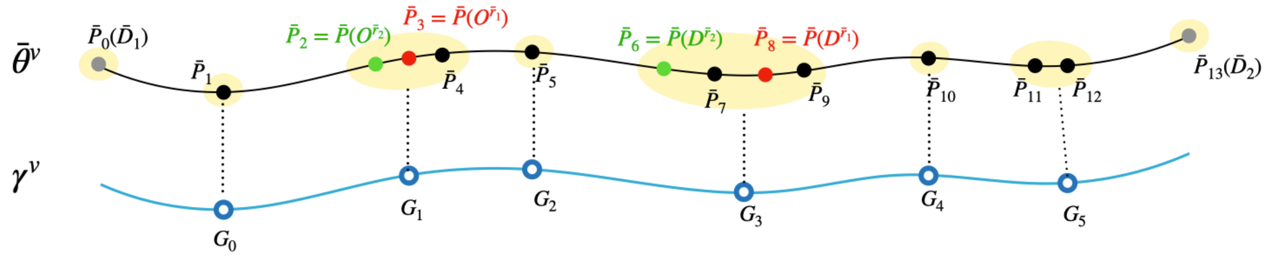


Figure 1: Reference route $\bar{\theta}^v$ and travel pattern γ^v

First of all, we retrieve a list of reference requests from $\bar{\mathcal{R}}$ which are *similar* to r. We say that a reference request $\bar{r}$ is similar to r if and only if:

$$\max(t(O^r, O^{\bar{r}}), t(O^{\bar{r}}, O^r)) \leqslant \delta^s$$

$$\max(t(D^r, D^{\bar{r}}), t(D^{\bar{r}}, D^r)) \leqslant \delta^s$$

and

$$|e_{O^r} - e_{O^{\bar{r}}}| \leqslant \delta^t$$

where parameter δ^s and δ^t are *similarity* parameters: δ^s bounds the travel time (distance) between the origins (resp. destinations) of two similar requests, and δ^t bounds the difference between the earliest pick-up times between r and $\bar{r}$. Let $\bar{\mathcal{R}}^r$ denote the set of reference requests similar to r. In order to speed the construction of $\bar{\mathcal{R}}^r$, we divide the transit networks into *zones* and the time horizon into *periods* (which are typically larger than what have been utilized in the *ST-Cluster* strategy). For every request r (reference or real), we quickly identify the zone $z(O^r)$ (resp. $z(D^r)$) related to O^r (resp. D^r) and the periods $H(O^r)$ (resp.$H(D^r)$) during which O^r (resp. D^r) can be serviced. Then we restrict the search for similar reference requests to those in $\bar{\mathcal{R}}$ whose origin and destination lie in $z(O^r)$ and $z(D^r)$ respectively, and whose time windows are consistent with $H(O^r)$ and $H(D^r)$.

Next, we retrieve a set of *guiding objects* $GO^r = \{\dots, (\gamma^v, G^o, G^d), \dots\}$ from $\bar{\mathcal{R}}^r$. If two reference requests are inserted in the same reference route and their insertion positions belong to the same key point clusters, then they correspond to the same guiding object. In such a case, we only keep one occurrence of each guiding object in GO^r. For example, in figure 1, if $\bar{r}_1$ and $\bar{r}_2$ are both similar to the target real request r, then only one occurrence of (γ^v, G_2, G_4) is kept in GO^r.

Once GO^r is constructed, we retrieve the candidate insertion parameters for r. Specifically, a key point P in a real route θ^v is called a *child* of a guiding point $G \in \gamma^v$, if and only if the time windows of P overlap with the temporal neighborhood of G, that means if:

$$e^a_P \leqslant t^{max}_G$$

and

$$l^d_P \geqslant t^{min}_G.$$

We notice that a guiding point $G \in \gamma^v$ may have multiple *children* in θ^v, these children being consecutive. Symmetrically, a key point can also have multiple *parents* in γ^v. For example, in Figure 2, P_2, P_3 and P_4 are children of G_2 and P_4 has two parents G_2 and G_3. For each guiding point G, we only memorize lc_G, the *predecessor* of the *left-most child* of G, and rc_G, the *right-most child* of G. In Figure 2 we have $lc_{G_1} = P_1$ and $rc_{G_1} = P_4$. If D_2 is the only key point satisfying the above condition then we set $rc_G = lc_G$. At the beginning, all real routes only consist of the two special key points D_1 and D_2. For every $G \in \gamma^v$, both lc_G and rc_G are initialized to be D_1. Then, given $(\gamma^v, G^o, G^d) \in GO^r$, the corresponding candidate insertion parameters are 5-tuples $(v, P_o, \epsilon_o, P_d, \epsilon_d)$ such that $P_o \in \mathcal{L}^o$ and $P_d \in \mathcal{L}^d$, where $\mathcal{L}^o$ is the list of candidate insertion positions for O^r, containing all the key points between lc_{G^o} (included) and rc_{G^o} (included), the same definition holding for $\mathcal{L}^d$. For the guiding object (γ^v, G^o, G^d) of Figure 2, we have $\mathcal{L}^o = \{P_1, P_2, P_3, P_4\}$ and $\mathcal{L}^d = \{P_7\}$.

Finally, the *best-fit insertion heuristic* is utilized over all the candidate insertion parameters given by GO^r. In other words, among all feasible insertion parameters, we keep the one that minimizes the insertion cost, and insert r for real. However, if r fails to be inserted via GIM, we put it aside and try the next request in $\mathcal{R}^e$. We wait until Phase 2 before coming back to r. Every time a request is inserted into v via GIM, time windows of key points along θ^v are modified, and lc_G and rc_G are updated for $G \in \gamma^v$.

5.4 Phase 2: the filtering system

At each decision epoch e, Phase 2 of the *GIM-FS* framework consists of trying to insert the requests $r \in \mathcal{R}^e$ rejected by GIM in Phase 1, while keeping with the best-fit insertion principle. Yet, the current state of the system may involve thousands of vehicles and hundreds of thousands of key points. Consequently, the number of potential insertion parameters $(v, P_o, \epsilon_o, P_d, \epsilon_d)$ may be too large regarding the computational time η allowed to execute the assignment/routing process. So we must filter the search for the insertion parameters. For that, we adapt the *Filtering System* (FS) introduced in [27] to the dynamic context. The trick is to maintain throughout the decision

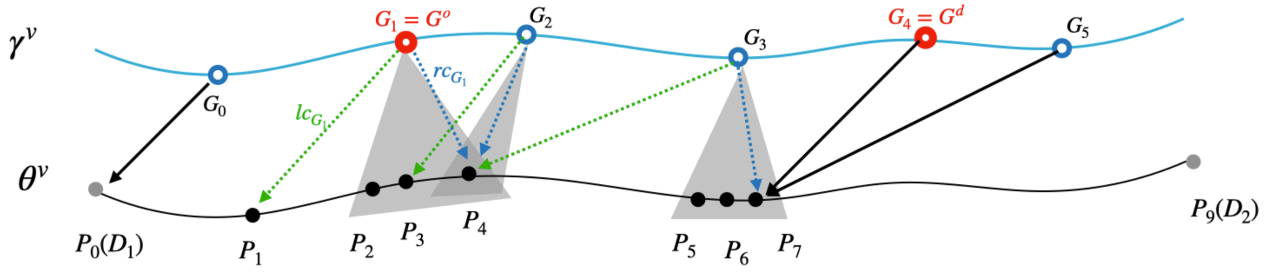


Figure 2: The projection between a travel pattern γ^v and the real route θ^v. For a point G, the dashed green arrow points at lc_G, the dashed green arrow points at rc_G, and the solid black arrow means that $lc_G = rc_G$. For the sake of illustration, we select several guiding points and use emitting gray shadows to elaborate their children

process a skeleton representation of the routes that involves a partition of the transit networks into *zones* and of the time horizon into *periods*. These periods and zones are the same as in Section 5.3 and allow us to quickly identify the vehicles and the key points likely to lead to feasible insertions of a given request. In the following, we briefly present FS.

As mentioned above, the transit network space is partitioned into a set of *zones*, $\mathcal{Z}$ and the time horizon is partitioned into a set of *periods*, $\mathcal{H}$, whose granularity may be adapted according to the context. Then the skeleton representation of the current state of the system is provided by a collection of *filtering* matrices: the *vehicle filtering matrix* $M^{\oplus}[\mathcal{Z} \times \mathcal{H}]$, where $\oplus$ is an operator to be clarified later, and the *insertion position filtering matrices* $M^v[\mathcal{Z} \times \mathcal{H}], v = 1, \dots, V$. The matrices are defined as follows.

- For any zone z, any period h and any key point P related to vehicle v, we define the *elastic duration* $El_{P,z,h}$ as $El_{P,z,h} = \min(ub_{P,z}, t_{h+1}) - \max(lb_{P,z}, t_h)$, where $lb_{P,z}$ (resp. $ub_{P,z}$) denotes the estimated earliest (resp. latest) time when v can arrive at (resp. depart from) at z while coming from (resp. going to) P. These two values are computed using the *four-references estimator* [27] which determines the estimated travel time between any node and any zone. Intuitively, $El_{P,z,h}$ provides us with an estimation of the *plausibility* that v may enter into z during period h while moving from P. The larger the value of $El_{P,z,h}$, the more plausible the move becomes.
- Then, for any zone z and any period h, the *vehicle filtering* matrix $M^{\oplus}$ identifies vehicles that are able to move through zone z during period h, together with the key points likely to allow this incursion. More precisely, we denote by $Pl^{\oplus}_{v,z,h}$ the quantity obtained by applying the associate operator $\oplus$ to the elastic durations $El_{P,z,h}$, $P \in \theta^v$. $Pl^{\oplus}_{v,z,h}$ provides us with the *plausibility* of the presence of vehicle v inside zone z during period h. The operator $\oplus$ may correspond to either the sum or the max of its arguments, or to the sum of the two largest elements. According to this, $M^{\oplus}[z, h]$ is defined as the set of the vehicles v such that $Pl^{\oplus}_{v,z,h}$ is not null:
$$M^{\oplus}[z, h] = \{(v, Pl^{\oplus}_{v,z,h}) | v \in \mathcal{V}, Pl^{\oplus}_{v,z,h} > 0\}$$
- For each vehicle v, the insertion position filtering matrix M^v contains tuples $(P, El_{P,z,h})$ for each $P \in \theta^v$ and z, h such that $El_{P,z,h} > 0$. Such a matrix may be understood as a reverse matrix of $M^{\oplus}$ and helps us in updating $M^{\oplus}$ every time an insertion is performed.

Identification of Candidate Vehicles: When we try to insert a new request r, we first utilize the vehicle filtering matrices $M^{\oplus}$ to quickly identify a subset of candidate vehicles $\mathcal{V}^r$. We denote by $z(O^r) \in \mathcal{Z}$ (resp. $z(D^r)$) the zone related to O^r (resp. D^r) and by $H(O^r) \subset \mathcal{H}$ (resp. $H(D^r)$) the periods when O^r (resp. D^r) may be serviced. The vehicle candidate set $\mathcal{V}^{r,\oplus}$ contains vehicles that appear in both $M^{\oplus}[z(O^r), H(O^r)]$ and $M^{\oplus}[z(D^r), H(D^r)]$.

Identification of Candidate Insertion Parameters: For each candidate v, potential insertion positions are determined from the matrix M^v. A key point P_o is regarded as a candidate insertion position for O^r if it appears in $M^v[z(O^r), H(O^r)]$ and meets conditions ensuring that independently inserting O^r at P does not violate load and time constraints on P_o. The same validation is applied for D^r. The candidate positions then undergo the complete feasibility test explained in Section 4.1, and the best-fitted one is retained. If no viable insertion is found, r is rejected.

Stopping mechanism A *stopping mechanism* can be integrated into FS to further accelerate the process. It relies on a trial threshold $\mathcal{T}$ and a counter τ, that is initialized to 0 every time we start inserting a request r and that is incremented every time some insertion parameter $(v, P_o, \epsilon_o, P_d, \epsilon_d)$ is tried for insertion. The increment consists in an estimation of the computational effort induced by this trial. In practice, this computational effort is set to 1 if the *constraint propagation procedure* is called and to 0 else. Then the candidate vehicles v in $\mathcal{V}^r$ are sorted according to a score, that combines $Pl^{\oplus}_{v,z(O^r),h \in H(O^r)}$ and

$Pl^{\oplus}_{v,z(D^r),h\in H(D^r)}$ in order to quantify the plausibility that v may service both the origin and destination of r. The search for insertion parameters $(v, P_o, \epsilon_o, P_d, \epsilon_d)$ proceeds by scanning the candidate vehicles v according to decreasing score values. It stops when counter τ reaches the threshold value $\mathcal{T}$.

Therefore, FS has two variations. If the stopping mechanism is not implemented, it is referred to as *Partial Filtering System* (PFS); otherwise, we denote it as *Complete Filtering System* (CFS).

6 Numerical experiments

We implemented all the tested algorithms in the C++ language and executed the computations on a machine equipped with an AMD EPYC 7452 32-Core Processor and 512 GB of RAM.

6.1 Instances

Transit network We consider the transit network in Clermont-Ferrand, a mid-sized city in France, and its suburban area. The underlined graph is downloaded from OpenStreetMap, containing $31,357$ arcs and $13,839$ nodes, among which $1,496$ are selected as pick-up and drop-off points. Each pick-up and drop-off point is associated with a label *working*, *residential*, or *undefined*. In the underlined graph, 6.7% of the points are of type *working*, 54% of type *residential*, and 39.3% of type *undefined* (see Figure 3).

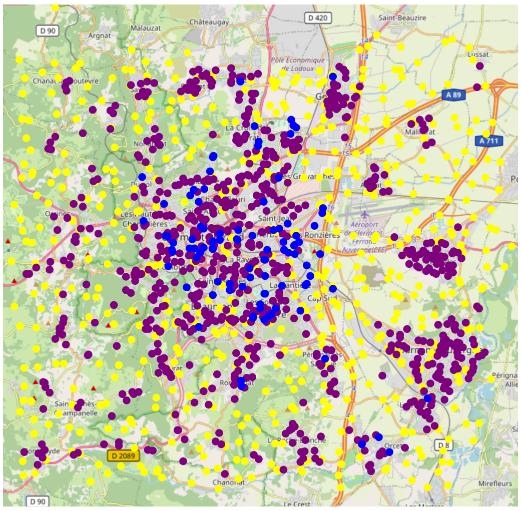


Figure 3: The pick-up and drop-off locations in the underlined transit network in Clermont-Ferrand. Blue, purple and yellow points represent points of type *working*, *residential*, and *undefined*, respectively

The vehicles The capacity of the vehicles is equal to 10. Determining the fleet size is part of the problem (remind that the request rejection rate and VTT are antagonistic criteria) and will be discussed in the next section.

Time horizon and decision epochs The *RSMoD* service period lasts for $T = 24$ hours, starting at 00:00 and ending at 24:00. Requests are collected and processed every 10 minutes. Therefore, within a time horizon of $T = 24$ hours, we end up with 144 decision epochs with $I_E = 10$ minutes. We use η as a parameter for sensitivity analysis.

Request distribution and configuration The large-scale *RSMoD* system studied here being still prospective, no suitable data is available to perfectly capture the usage of such a system. We have designed a passenger request generator to generate daily request instances that comply with the intended use case of the system. For each day, we consider five time slots: **Morning Slack (MS)**, from 00:00 to 06:00; **Morning Peak (MP)**, from 06:00 to 10:00; **Normal Hours (NH)**, from 10:00 to 15:00; **Evening Peak (EP)**, from 15:00 to 19:00; and **Evening Slack (ES)**, from 19:00 to 24:00. Then we ensure that 1% of the daily requests depart during **MS**, 20% during **NH**, 9% during **ES**, while **MP** and **EP** each accounts for 35% of the requests. The two peak periods correspond to passenger commuting trends. As for the spatial distribution of the requests, around 70% originate from *residential* (resp. *working*) points and terminate at *working* (resp. *residential*) points. They are referred to as "typical" passenger requests. Around 80% of "typical" requests traveling from o to d during **MP** (resp. **EP**) are related to a "symmetrical" passenger request traveling from d back to o during **EP** (resp. **MP**).

Each request r has a pick-up time window δ^r between 10 and 30 minutes, and a maximum ride-time $T^r = 2 \times t(o^r, d^r)$, where $t(o^r, d^r)$ denotes the fastest travel time from its origin to its destination. As for the submission times, 90% of the requests are *urgent* ones, with a difference between the submission time and the earliest pick-up time smaller than 1.5 hours. The rest are *relaxed* requests, that are submitted much earlier (at least 1.5 hours before their earliest pick-up time) or in advance. If we divide the time horizon $[0, T]$ into time bins of equal duration of 10 minutes, the distribution of requests submitted at each time bin is given in Figure 4.

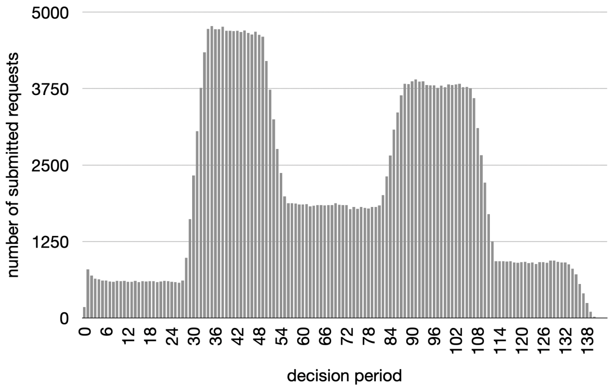


Figure 4: The number of requests submitted at each time bin

Recurrent Requests: We introduce a proportion β of $\mathcal{R}$ to identify the *recurrent* requests, which are considered to be almost stable from one day to another. This parameter, β, plays a key role in deducing the *reference* request set $\bar{\mathcal{R}}$ from $\mathcal{R}$. The *recurrent* requests should generate very similar reference requests, whereas the remaining requests will be generated to meet the statistical distribution of the requests.

Summary Table 1 summarizes the basic settings of the *RSMoD* system and the requests.

Table 1: Basic setting of the system and the requests

Param	Description	Value
$\lvert\mathcal{N}\rvert$	the number of pick-up and delivery locations	1,496
T	the planning horizon	24 h
I_E	the duration of each decision epoch of each decision epoch	10 min.
V	SAV fleet size	1,977
$\lvert\mathcal{R}\rvert$	the number of dynamic requests	300k
δ^r	the length of pick-up time window	15 min.
T^r	the maximum ride time of any request r	$2t(o^r, d^r)$
q^r	the load of request r	1
$\lvert\bar{\mathcal{R}}\rvert$	the number of reference requests	400k
β	the proportion of "random" part in $\mathcal{R}$	10%

6.2 Fleet sizing and the reference *S_DARP* problem

The *RSMoD* system studied in this paper aims to cater to around 300,000 dynamic requests daily. As explained in Section 5, we compute pertinent fleet sizes by proceeding as follows.

First, we generate six *static* instances of varying sizes that meet the statistical features previously described. These instance sizes range from 300,000 to 425,000, increasing in increments of 25,000. For any static instance, we address the associated static problem *S_DARP* with BF (see Section 4.2) and get the resulting fleet sizes. Table 2 shows the fleet sizes obtained from resolving each of the six static problems.

Next, we generate five trial *dynamic* instances $\{i0, i1, \dots, i4\}$, with 300,000 dynamic requests. For each instance, we address the six related problems D_DARP while relying on the basic algorithm BF and fixing the fleet size at the above-established six values. We get every time a request rejection rate. Table 3 presents the passenger rejection rates under different fleet sizes for different instances and the corresponding averaged values.

Then we consider 1,977 as the most reasonable fleet size: The passenger rejection rate is close to 0, under significantly smaller economic costs than the other tested size of 2,069. Of course, we might try fleet sizes between 1,977 and 2,069: Here we only propose a tentative approach for the fleet sizing issue.

An advantage of the above process is that it provides us with both the reference request set and the reference route sets required by *GIM*. The static request set of size 400,000 yielding the 1,977 SAVs is selected as the reference set $\bar{\mathcal{R}}$, and the related solution becomes the reference route set $\bar{\Theta}$.

6.3 The behavior of the RSMoD system

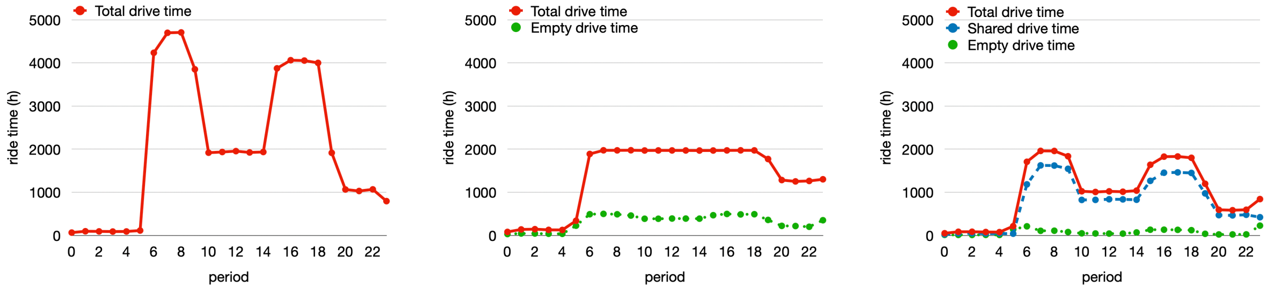
We compare here the performance of the *RSMoD* system to two other systems: *private* and *MoD*. In the *private* system, the 300,000 passenger requests are all supported by private vehicles. We estimate a fleet size of $240,951$ in *private*, which is the number of one-way requests plus the number of symmetric requests. In *MoD*, ride-sharing is not allowed. Its fleet size is 1,977, the same as in *RSMoD*. The baseline approach BF is used to solve the corresponding routing problems.

As shown in Table 4, in *private* system, the total drive time is much higher than *MoD* and *RSMoD*, because much more vehicles are utilized. Almost half of the requests are rejected in *MoD*, but its total drive time is still higher than that in *RSMoD*. Because no ride-sharing is allowed in *MoD*, and SAVs must make the re-positioning to reach different pick-up locations. We see that empty drive time contributes to 31.5% of the total drive time (the column **Relative empty drive time**) in *MoD*. Additionally, according to the results, to cater for all the rejected requests in MoD, an estimated minimum number of additional SAVs of around $2,000$ would be required, resulting in a great increase in the total drive time as well. Remind that most of the rejected requests are associated with peak hours, we would need many more SAVs than $2,000$. In contrast, thanks to ride-sharing, 78.1% of the total drive time is shared by more than one passenger in *RSMoD*, and only 7.4% is issued from empty relocation, with a smaller total drive time compared to the two other system configurations. These results prove several advantages of the *RSMoD* system. First, the on-route vehicles in the city can be largely reduced, thus the traffic congestion and occupancy of the city infrastructure can be improved. And SAVs are self-driving, reducing the cost of human labor. Furthermore, the fact that *RSMoD* decreases the total drive time in our simulation proves that the consumption of energy can be greatly economized by deploying such a novel system. Finally, the passengers' travel comfort remains the same in *MoD*, and is decreased in *RSMoD*, with an average of 16.63 minutes per passenger, suggesting that SAVs make detours to promote the ride-sharing so that the rejection of requests can be minimized.

Figure 5 provides a better vision of how drive times of each category (total drive time, empty drive time, and

Table 2: Fleet sizes to different static problem sizes

Instance size	300,000	325,000	350,000	375,000	400,000	425,000
Fleet size	1,621	1,667	1,768	1,869	1,977	2,069



(a) private (b) MoD (c) RSMoD

Figure 5: Accumulated drive time, empty drive time, and shared drive time (where appropriated) in each period of duration of one hour

Table 3: The rejection rate when solving static problems with different fleet sizes

Fleet size	**Instance**					**Avg**
	i0	**i1**	**i2**	**i3**	**i4**	
1,621	6.2%	6.4%	6.3%	6.2%	6.3%	6.3%
1,667	4.7%	4.8%	4.9%	4.7%	5.0%	4.8%
1,768	2.2%	2.2%	2.1%	2.3%	2.3%	2.2%
1,869	1.0%	0.8%	1.0%	1.0%	1.0%	1.0%
1,977	0.1%	0.0%	0.1%	0.1%	0.1%	0.1%
2,069	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

shared drive time) are distributed across the service horizon. In both *private* and *RSMoD* systems, the drive time during peak hours is much higher than in other time slots. In *MoD*, the total drive time remains almost the same during **MP**, **EP** and **NH**, because the number of requests that can be serviced during these time slots is almost the same (excessive requests during peak hours are rejected). Furthermore, as shown in Sub-figure 5b and Sub-figure 5c, empty drive time in *MoD* is higher than in *RSMoD*, especially during peak hours, because more detours are required to pick up new passengers. And a large proportion of trajectory during **MP**, **EP** and **NH** are shared in *RSMoD*.

6.4 The behavior of the GIM-FS algorithmic scheme

In addition to the two approaches, GIM-PFS and GIM-CFS, we try 4 variants: BF, PFS, CFS, and GIM-BF, of the GIM-FS algorithmic scheme. Whenever FS is utilized, it is configured as follows: the spatial-temporal partition results in 24 periods of equal duration and 20 zones of equal area. In CFS, the stopping criterion $\mathcal{T}^r$ set for each request r is equivalent to exploring at most 10% of the already-activated vehicles. In addition, a *regeneration mechanism* is employed, which proportionally expands the search space when no feasible insertion is present in the first candidate search space. These variants are summarized in Table 6. Furthermore, in GIM, the parameter α that defines the spatial neighborhood of pattern points is fixed at 10 minutes.

6.4.1 General results

Table 7 presents the general results when solving the dynamic problem with different approaches and different values of the request parameter β.

The column **CPU time** represents the accumulated execution time to solve *D_DARP*. Generally speaking, for every β, the two-phase approaches integrated with GIM are more efficient than their basic algorithm, thanks to the largely reduced search space provided by GIM. Meanwhile, when utilizing two-phase approaches, the quality of the solution remains almost the same, compared to those when applying the corresponding basic algorithms. Typically, GIM-BF and GIM-PFS reject slightly more requests and demonstrate a slightly longer total drive time than BF and GIM-BF, respectively. In addition, for every β, GIM-CFS outperforms CFS, in both execution time and solution quality, thanks to the guide provided by GIM in Phase 1. Specifically, although CFS can solve the problem in a very short time thanks to a largely reduced search space, the resulting solution is less satisfactory. Because this reduction is only deduced from the current status of the system in a rather myopic and greedy way, without any consideration of the global quality of the solution. However, GIM utilizes the additional information of the reference resolution and "forces" vehicles to stick to their pre-defined travel patterns extracted from optimal reference routes. This helps

Table 4: Performance of three systems

	Fleet size	**Rejection rate**	**Total drive time (h)**	**Relative[1] shared drive time**	**Relative[2] empty drive time**	**Average in -vehicle time (min)**
private	240,951	/	49,585	/	/	9.92
MoD	1,977	48.0%	33,343	/	31.5%	10.30
RSMoD	1,977	0.5%	24,035	78.1%	7.4%	16.63

[1] Relative shared drive time is calculated as the percentage of total drive time that is shared with other passengers.
[2] Relative empty drive time is calculated as the percentage of total drive time where the vehicle is empty.

Table 5: Sensitivity analyses of GIM

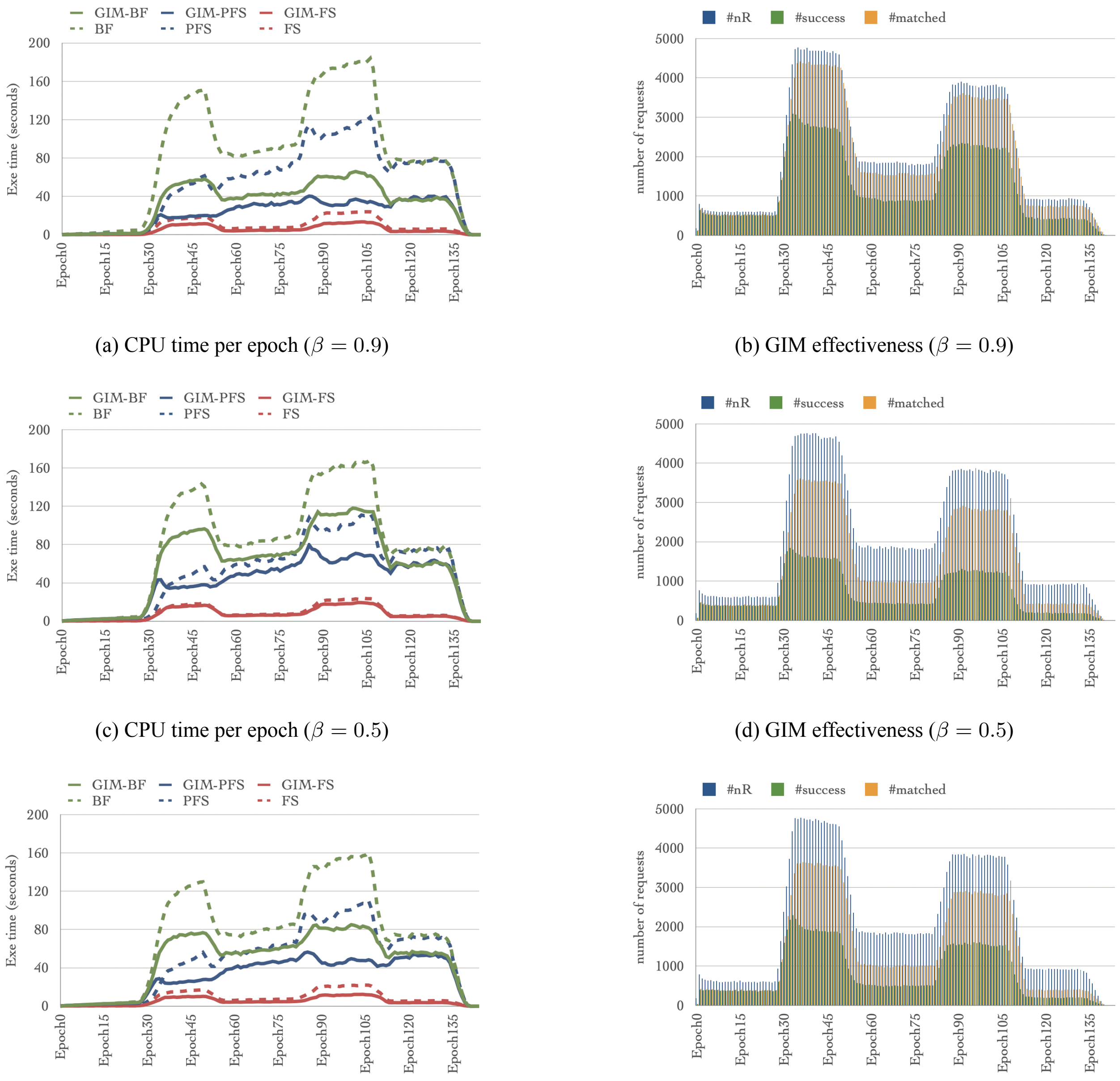
δ^s (min)	δ^t (min)	Approach	CPU time (min)	Rejection rate	Total drive time (h)	Average in-vehicle time (min)	matchGI rate	succGI rate
2	3	GIM-BF	137.1	0.7%	**23,571**	16.39	67.5%	41.0%
		GIM-PFS	93.5	**0.7%**	23,585	**16.38**		41.0%
		GIM-FS	17.9	2.3%	26,725	17.16		36.3%
	7.5	GIM-BF	134.2	0.7%	23,604	16.38	89.6%	40.0%
		GIM-PFS	92.1	0.7%	23,600	**16.37**		40.0%
		GIM-FS	17.6	2.3%	26,713	17.15		34.9%
	15	GIM-BF	85.3	0.9%	24,092	16.45	92.3%	62.0%
		GIM-PFS	56.0	0.9%	24,095	16.45		61.8%
		GIM-FS	13.3	1.7%	26,126	16.87		58.4%
3	3	GIM-BF	82.1	0.8%	24,129	16.40	82.7%	60.4%
		GIM-PFS	54.1	0.8%	24,109	16.41		60.3%
		GIM-FS	12.8	1.6%	26,117	16.8		56.6%
	7.5	GIM-BF	76.5	1.2%	24,497	16.58	95.4%	65.1%
		GIM-PFS	47.9	1.2%	24,498	16.57		65.2%
		GIM-FS	12.6	2.0%	26,190	16.99		62.4%
	15	GIM-BF	71.8	1.2%	24,527	16.53	97.1%	62.6%
		GIM-PFS	45.9	1.2%	24,527	16.5		62.6%
		GIM-FS	11.8	1.8%	26,166	16.87		59.3%
5	3	GIM-BF	68.5	3.1%	25,655	17.51	94.3%	64.2%
		GIM-PFS	40.3	3.2%	25,657	17.55		64.0%
		GIM-FS	12.5	4.0%	27,048	17.92		62.3%
	7.5	GIM-BF	44.8	2.7%	25,817	17.36	99.0%	76.6%
		GIM-PFS	26.3	2.7%	25,809	17.37		76.5%
		GIM-FS	**11.6**	3.0%	26,671	17.52		76.0%
	15	GIM-BF	37.3	2.8%	26,035	17.4	**99.5%**	**81.0%**
		GIM-PFS	22.7	2.8%	26,033	17.42		80.7%
		GIM-FS	12.4	3.0%	26,668	17.49		80.6%

The best results among all combinations are presented in bold font.

have a sort of pre-concern of future requests in the current insertion, thus making up for the inefficiency of CFS.

The reduction in CPU time by GIM-BF and GIM-PFS compared to BF and PFS, respectively, becomes stronger when there are more similar requests between the real request set and the reference request set. When more requests have similar reference requests, the proportion of requests that can be inserted via GIM should be higher, making the execution faster. However, this reduction effect between GIM-CFS and CFS across different scenarios is less prominent, for the already short enough execution time.

A closer analysis of the efficiency of the approaches can



(a) CPU time per epoch ($\beta = 0.9$)

(b) GIM effectiveness ($\beta = 0.9$)

(c) CPU time per epoch ($\beta = 0.5$)

(d) GIM effectiveness ($\beta = 0.5$)

(e) CPU time per epoch ($\beta = 0.3$)

(f) GIM effectiveness ($\beta = 0.3$)

Figure 6: Effectiveness of the algorithm. Each row corresponds to a value β. Sub-figures in the first column represent the CPU time spent to solve the sub routing problem for each decision epoch when using different approaches. Sub-figures in the second column represent the number of requests to be processed (blue), that are associated with at least one similar reference request in GIM (orange), and that are successfully inserted via GIM (green)

be done through Figure 6, which shows different statistics in every decision epoch. In terms of the CPU time spent on the insertion of requests in each decision epoch, we observe that approaches integrated with GIM spend much less time regarding their basic approach. For the purpose of thorough analyses, we did not set a specific value of η. From Sub-figures 6a, 6c and 6e, we see that when setting η at different values, several methods may fail to complete during certain epochs: When $\eta = 40$ seconds, only CFS and GIM-CFS can finish all insertions in every epoch.

The efficiency of two-phase approaches can be explained by the number of successful insertions via GIM. For example, from Sub-figures 6b, 6d and 6f, we note that with $\beta = 0.9$, the number of requests that have at least one similar reference request (orange bars in figures), and the successful insertions via GIM (green bars in figures) are higher than with the other two β values, especially during peak hours. Therefore, fewer requests are passed to Phase 2 when $\beta = 0.9$, and the insertion process becomes faster.

Table 7: General test results with different approaches under different β

(a) $\beta = 0.9$

Approach	CPU time (min)	#rejection	Rejection rate	Total drive time (h)	Average in-vehicle time (min)
BF	200.4	1,631	0.5%	24,035	16.63
PFS	130.6	1,624	0.5%	24,040	16.63
CFS	21.8	10,320	3.4%	27,486	15.99
GIM-BF	82.1	2,420	0.8%	24,129	16.40
GIM-PFS	54.1	2,493	0.8%	24,109	16.41
GIM-CFS	12.8	4,682	1.6%	26,117	16.80

(b) $\beta = 0.5$

Approach	CPU time (min)	#rejection	Rejection rate	Total drive time (h)	Average in-vehicle time (min)
BF	188.6	2,725	0.9%	24,306	16.67
PFS	123.1	2,674	0.9%	24,274	16.68
CFS	21.5	11,969	4.0%	27,594	16.02
GIM-BF	108.6	3,792	1.3%	24,189	16.71
GIM-PFS	72.6	3,762	1.3%	24,190	16.69
GIM-CFS	15.2	8,136	2.7%	26,746	17.43

(c) $\beta = 0.3$

Approach	CPU time (min)	#rejection	Rejection rate	Total drive time (h)	Average in-vehicle time (min)
BF	178.1	2,093	0.7%	24,131	16.73
PFS	118.0	1,973	0.7%	24,104	16.73
CFS	20.2	10,798	3.6%	27,512	16.10
GIM-BF	115.7	3,270	1.1%	23,978	16.69
GIM-PFS	76.0	3,302	1.1%	23,952	16.69
GIM-CFS	12.3	8,128	2.7%	26,839	17.49

Table 6: Description of tested approaches

Approach	**Description**
BF	Best-Fit insertion heuristic
PFS	Partial Filtering System
CFS	Complete Filtering System
GIM-BF	GIM is used in Phase 1, BF is used in Phase 2
GIM-PFS	GIM is used in Phase 1, PFS is used in Phase 2
GIM-CFS	GIM is used in Phase 1, CFS is used in Phase 2

6.4.2 Sensitivity analysis

From the above discussion, we see that the rate of successful insertions via GIM has a great impact on the efficiency of the two-phase methods. This rate is directly influenced by the number of requests that can be matched to at least one similar reference request. Two parameters, δ^s and δ^t, control the similarity measure, thus the match rate. From now on, we only focus on the $\beta = 0.9$, and analyze the sensitivity of the two-phase algorithms while varying the values of δ^s and δ^t.

Table 5 shows the performances of different methods under different δ^s and δ^t combinations.

Typically, the column *matchGI rate* represents the percentage of requests that have at least one similar reference request, and the column *succGI rate* computes the percentage of requests that are successfully inserted via GIM. We conclude that when the similarity measure becomes more relaxed (bigger δ^s and δ^t), the matchGI rate and succGI rate effectively become bigger. For example, in the most relaxed situation, 99.5% of requests are passed to GIM, and 81% of requests are successfully inserted from GIM. We confirm that the general trend is that a higher succGI rate implies a smaller CPU time. However, a bigger succGI rate is not equivalent to a better solution. We get the opposite situation instead. Because such an increase in succGI rate is gained from a more relaxed request similarity measure. Consequently, as long as a feasible insertion is found in Phase 1, GIM may force the request to be inserted by mimicking the insertion patterns of a not-very-similar reference request. In the short term, the insertion of such a request is based on a relatively bad candidate insertion parameter. In the long term, such insertion may lead to a distortion of the trajectory of the vehicle in question from its pre-defined pattern, preventing later perfectly matched requests from being inserted, thus worsening the quality of the solution.

7 Conclusion

In this paper, we studied a prospective large-scale *RSMoD* transportation system with SAVs. We showed that this SAV *RSMoD* system would provide on-demand services with a very low rejection rate and satisfactory passenger riding comfort, while, thanks to its ride-sharing feature, significantly reducing the total driving time, energy costs, and the number of on-route vehicles.

As for the operational management of such a system, we introduced a GIM: *Guided Insertion Mechanism*, that learns from a reference solution deriving from a statistical virtual version of the related DARP, and combined it with a Filtering System to form a two-phase algorithm framework, GIM-FS. We proved that GIM is likely to significantly help the manager of the system efficiently assign dynamic requests to the vehicles.

Yet, several issues remain that will motivate our future research. First, we must try (algorithmic issue) to improve the basic greedy insertion heuristic used to solve the statistical virtual version of the related DARP, in a way that will fit the very large-scale feature and that will allow us to make the GIM component more efficient. Next, we should integrate into our decision-making framework the energetic issue, and thus take into account that the schedule of a vehicle should include its recharge transactions and the time-varying costs of these recharge transactions. Finally, we should address the robustness issue, related to the non-deterministic features of any mass *RSMoD* system: traffic congestion, passenger no-shows, etc.

Acknowledgement

This work was supported by the International Research Center “Innovation Transportation and Production Systems” of the I-SITE CAP 20-25.

References

[1] Ransford A. Acheampong, Alhassan Siiba, Dennis K. Okyere, and Justice P. Tuffour. Mobility-on-demand: An empirical study of internet-based ride-hailing adoption factors, travel characteristics and mode substitution effects. Transportation Research Part C: Emerging Technologies, 115:102638, June 2020. https://doi.org/10.1016/j.trc.2020.102638.

[2] Niels Agatz, Alan Erera, Martin Savelsbergh, and Xing Wang. Optimization for dynamic ride-sharing: A review. European Journal of Operational Research, 223(2):295–303, December 2012. https://doi.org/10.1016/j.ejor.2012.05.028.

[3] Vincent Armant and Kenneth N. Brown. Fast optimised ridesharing: Objectives, reformulations and driver flexibility. Expert Systems with Applications, 141:112914, March 2020. https://doi.org/10.1016/j.eswa.2019.112914.

[4] Bilge Atasoy, Takuro Ikeda, Xiang Song, and Moshe E. Ben-Akiva. The concept and impact analysis of a flexible mobility on demand system. Transportation Research Part C: Emerging Technologies, 56:373–392, July 2015. https://doi.org/10.1016/j.trc.2015.04.009.

[5] Slim Belhaiza. A Hybrid Adaptive Large Neighborhood Heuristic for a Real-Life Dial-a-Ride Problem. Algorithms, 12(2):39, February 2019. https://doi.org/10.3390/a12020039.

[6] Gerardo Berbeglia, Jean-François Cordeau, and Gilbert Laporte. A Hybrid Tabu Search and Constraint Programming Algorithm for the Dynamic Dial-a-Ride Problem. INFORMS Journal on Computing, May 2011. https://doi.org/10.1287/ijoc.1110.0454.

[7] Kris Braekers, An Caris, and Gerrit K. Janssens. Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. Transportation Research Part B: Methodological, 67:166–186, September 2014. https://doi.org/10.1016/j.trb.2014.05.007.

[8] Roberto Wolfler Calvo and Alberto Colorni. An effective and fast heuristic for the Dial-a-Ride problem. 4OR, 5(1):61–73, March 2007. https://doi.org/10.1007/s10288-006-0018-0.

[9] CEU. MOVE., Univ. Eiffel., TRT., Panteia., GDCC., and STRATEC. Study on new mobility patterns in European cities: final report. Task A, EU wide passenger mobility survey. Publications Office, LU, 2022. https://data.europa.eu/doi/10.2832/728583.

[10] Jean-François Cordeau. A Branch-and-Cut Algorithm for the Dial-a-Ride Problem. Operations Research, 54(3):573–586, 2006. https://doi.org/10.1287/opre.1060.0283.

[11] Jean-François Cordeau and Gilbert Laporte. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. Transportation Research Part B: Methodological, 37(6):579–594, July 2003. https://doi.org/10.1016/S0191-2615(02)00045-0.

[12] Jean-François Cordeau and Gilbert Laporte. The dial-a-ride problem: models and algorithms. Annals of Operations Research, 153(1):29–46, June 2007. https://doi.org/10.1007/s10479-007-0170-8.

[13] Jean-François Cordeau, Gilbert Laporte, Jean-Yves Potvin, and Martin W. P. Savelsbergh. Chapter 7 Transportation on Demand. In Cynthia Barnhart and Gilbert Laporte, editors, Handbooks in Operations Research and Management Science, volume 14 of Transportation, pages 429–466. Elsevier, January 2007. https://doi.org/10.1016/S0927-0507(06)14007-4.

[14] Samuel Deleplanque and Alain Quilliot. Insertion techniques and constraint propagation for the DARP. In 2012 Federated Conference on Computer Science and Information Systems (FedCSIS), pages 393–400, September 2012.

[15] Daniel J. Fagnant and Kara M. Kockelman. Dynamic ride-sharing and fleet sizing for a system of shared autonomous vehicles in Austin, Texas. Transportation, 45(1):143–158, January 2018. https://doi.org/10.1007/s11116-016-9729-z.

[16] Jeffery B. Greenblatt and Susan Shaheen. Automated Vehicles, On-Demand Mobility, and Environmental Impacts. Current Sustainable/Renewable Energy Reports, 2(3):74–81, September 2015. https://doi.org/10.1007/s40518-015-0038-5.

[17] Martin L Hazelton. Inference for origin–destination matrices: estimation, prediction and reconstruction. Transportation Research Part B: Methodological, 35(7):667–676, August 2001. https://doi.org/10.1016/S0191-2615(00)00009-6.

[18] Sin C. Ho, W.Y. Szeto, Yong-Hong Kuo, Janny M.Y. Leung, Matthew Petering, and Terence W.H. Tou. A survey of dial-a-ride problems: Literature review and recent developments. Transportation Research Part B: Methodological, 111:395–421, May 2018. https://doi.org/10.1016/j.trb.2018.02.001.

[19] Zihan Hong, Ying Chen, Hani S. Mahmassani, and Shuang Xu. Commuter ride-sharing using topology-based vehicle trajectory clustering: Methodology, application and impact evaluation. Transportation Research Part C: Emerging Technologies, 85:573–590, December 2017. https://doi.org/10.1016/j.trc.2017.10.020.

[20] Esa Hyytiä, Lauri Häme, Aleksi Penttinen, and Reijo Sulonen. Simulation of a large scale dynamic pickup and delivery problem. In Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques, Malaga, Spain, 2010. ICST. https://doi.org/10.4108/ICST.SIMUTOOLS2010.8701.

[21] Siddhartha Jain and Pascal Van Hentenryck. Large Neighborhood Search for Dial-a-Ride Problems. In Jimmy Lee, editor, Principles and Practice of Constraint Programming – CP 2011, volume 6876, pages 400–413. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. https://doi.org/10.1007/978-3-642-23786-7_31.

[22] Jang-Jei Jaw, Amedeo R. Odoni, Harilaos N. Psaraftis, and Nigel H. M. Wilson. A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. Transportation Research Part B: Methodological, 20(3):243–257, June 1986. https://doi.org/10.1016/0191-2615(86)90020-2.

[23] Jiwon Kim and Hani S. Mahmassani. Spatial and Temporal Characterization of Travel Patterns in a Traffic Network Using Vehicle Trajectories. Transportation Research Procedia, 9:164–184, 2015. https://doi.org/10.1016/j.trpro.2015.07.010.

[24] Jiwon Kim, Kai Zheng, Jonathan Corcoran, Sanghyung Ahn, and Marty Papamanolis. Trajectory Flow Map: Graph-based Approach to Analysing Temporal Evolution of Aggregated Traffic Flows in Large-scale Urban Networks, December 2022. https://doi.org/10.48550/arXiv.2212.02927.

[25] Da Lei, Xuewu Chen, Long Cheng, Lin Zhang, Satish V. Ukkusuri, and Frank Witlox. Inferring temporal motifs for travel pattern analysis using large scale smart card data. Transportation Research Part C: Emerging Technologies, 120:102810, November 2020. https://doi.org/10.1016/j.trc.2020.102810.

[26] Michael W. Levin. Congestion-aware system optimal route choice for shared autonomous vehicles. Transportation Research Part C: Emerging Technologies, 82:229–247, September 2017. https://doi.org/10.1016/j.trc.2017.06.020.

[27] Chijia Liu, Alain Quilliot, Hélène Toussaint, and Dominique Feillet. A filtering system to solve the large-scale shared autonomous vehicles Dial-a-Ride Problem. Transportation Research Part C: Emerging Technologies, 161:104551, April 2024. https://doi.org/10.1016/j.trc.2024.104551.

[28] Benjamin Loeb, Kara M. Kockelman, and Jun Liu. Shared autonomous electric vehicle (SAEV) operations across the Austin, Texas network with charging infrastructure decisions. Transportation Research Part C: Emerging Technologies, 89:222–233, April 2018. https://doi.org/10.1016/j.trc.2018.01.019.

[29] J. Carlos Martínez Mori, M. Grazia Speranza, and Samitha Samaranayake. On the Value of Dynamism in Transit Networks. Transportation Science, 57(3):578–593, May 2023. https://doi.org/10.1287/trsc.2022.1193.

[30] Santhanakrishnan Narayanan, Emmanouil Chaniotakis, and Constantinos Antoniou. Shared autonomous vehicle services: A comprehensive review. Transportation Research Part C: Emerging Technologies, 111:255–293, February 2020. https://doi.org/10.1016/j.trc.2019.12.008.

[31] Julie Paquette, Jean-François Cordeau, Gilbert Laporte, and Marta M. B. Pascoal. Combining multicriteria analysis and tabu search for dial-a-ride problems. Transportation Research Part B: Methodological, 52:1–16, June 2013. https://doi.org/10.1016/j.trb.2013.02.007.

[32] Sophie N. Parragh, Jorge Pinho de Sousa, and Bernardo Almada-Lobo. The Dial-a-Ride Problem with Split Requests and Profits. Transportation Science, 49(2):311–334, 2015. https://doi.org/10.1287/trsc.2014.0520.

[33] Sophie N. Parragh, Karl F. Doerner, and Richard F. Hartl. Variable neighborhood search for the dial-a-ride problem. Computers & Operations Research, 37(6):1129–1138, June 2010. https://doi.org/10.1016/j.cor.2009.10.003.

[34] Sophie N. Parragh and Verena Schmid. Hybrid column generation and large neighborhood search for the dial-a-ride problem. Computers & Operations Research, 40(1):490–497, January 2013. https://doi.org/10.1016/j.cor.2012.08.004.

[35] Ronik Ketankumar Patel, Roya Etminani-Ghasrodashti, Sharareh Kermanshachi, Jay Michael Rosenberger, and Ann Foss. Mobility-on-demand (MOD) Projects: A study of the best practices adopted in United States. Transportation Research Interdisciplinary Perspectives, 14:100601, June 2022. https://doi.org/10.1016/j.trip.2022.100601.

[36] Ronik Ketankumar Patel, Roya Etminani-Ghasrodashti, Sharareh Kermanshachi, Jay Michael Rosenberger, and Ann Foss. Exploring willingness to use shared autonomous vehicles. International Journal of Transportation Science and Technology, 12(3):765–778, September 2023. https://doi.org/10.1016/j.ijtst.2022.06.008.

[37] Marco Pavone. Autonomous Mobility-on-Demand Systems for Future Urban Mobility. In Markus Maurer, J. Christian Gerdes, Barbara Lenz, and Hermann Winner, editors, Autonomes Fahren: Technische, rechtliche und gesellschaftliche Aspekte, pages 399–416. Springer, Berlin, Heidelberg, 2015. https://doi.org/10.1007/978-3-662-45854-9_19.

[38] Harilaos N. Psaraftis. A Dynamic Programming Solution to the Single Vehicle Many-to-Many Immediate Request Dial-a-Ride Problem. Transportation Science, 14(2):130–154, May 1980. https://doi.org/10.1287/trsc.14.2.130.

[39] Harilaos N. Psaraftis. An Exact Algorithm for the Single Vehicle Many-to-Many Dial-A-Ride Problem with Time Windows. Transportation Science, 17(3):351–357, August 1983. https://doi.org/10.1287/trsc.17.3.351.

[40] Lisa Rayle, Danielle Dai, Nelson Chan, Robert Cervero, and Susan Shaheen. Just a better taxi? A survey-based comparison of taxis, transit, and ridesourcing services in San Francisco. Transport Policy, 45:168–178, January 2016. https://doi.org/10.1016/j.tranpol.2015.10.004.

[41] M. Schilde, K. F. Doerner, and R. F. Hartl. Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. Computers & Operations Research, 38(12):1719–1730, December 2011. https://doi.org/10.1016/j.cor.2011.02.006.

[42] Susan Shaheen and Adam Cohen. Shared ride services in North America: definitions, impacts, and the future of pooling. Transport Reviews, 39(4):427–442, July 2019. https://doi.org/10.1080/01441647.2018.1497728.

[43] Shuo Ma, Yu Zheng, and O. Wolfson. T-share: A large-scale dynamic taxi ridesharing service. In 2013 IEEE 29th International Conference on Data Engineering (ICDE), pages 410–421, Brisbane, QLD, April 2013. IEEE. https://doi.org/10.1109/ICDE.2013.6544843.

[44] Sacha Varone and Vytenis Janilionis. Insertion heuristic for a dynamic dial-a-ride problem using geographical maps. In MOSIM 2014, 10ème Conférence Francophone de Modélisation, Optimisation et Simulation, Nancy, France, November 2014. https://hal.science/hal-01166662.

[45] Yuandong Wang, Hongzhi Yin, Hongxu Chen, Tianyu Wo, Jie Xu, and Kai Zheng. Origin-Destination Matrix Prediction via Graph Convolution: a New Perspective of Passenger Demand Modeling. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 1227–1235, Anchorage AK USA, July 2019. ACM. https://doi.org/10.1145/3292500.3330877.

[46] Salomon Wollenstein-Betech, Mauro Salazar, Arian Houshmand, Marco Pavone, Ioannis Ch. Paschalidis, and Christos G. Cassandras. Routing and Rebalancing Intermodal Autonomous Mobility-on-Demand Systems in Mixed Traffic. IEEE Transactions on Intelligent Transportation Systems, 23(8):12263–12275, August 2022. https://doi.org/10.1109/TITS.2021.3112106.

[47] Zixuan Yu, Ping Zhang, Yang Yu, Wei Sun, and Min Huang. An Adaptive Large Neighborhood Search for the Larger-Scale Instances of Green Vehicle Routing Problem with Time Windows. Complexity, 2020:1–14, October 2020. https://doi.org/10.1155/2020/8210630.

[48] Junbo Zhang, Yu Zheng, and Dekang Qi. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. Proceedings of the AAAI Conference on Artificial Intelligence, 31(1), February 2017. https://doi.org/10.1609/aaai.v31i1.10735.

[49] Linjiang Zheng, Dong Xia, Xin Zhao, Longyou Tan, Hang Li, Li Chen, and Weining Liu. Spatial–temporal travel pattern mining using massive taxi trajectory data. Physica A: Statistical Mechanics and its

Applications, 501:24–41, July 2018. https://doi.org/10.1016/j.physa.2018.02.064.

Factors Influencing Cloud Computing Adoption in Small and Medium-Sized Enterprises: A Systematic Review

Nathalie Holler, Markus Westner*
E-mail: markus.westner@oth-regensburg.de
* Corresponding author

Faculty of Computer Science & Mathematics, OTH Regensburg Seybothstr. 2, 93053 Regensburg, Germany

Keywords: cloud computing adoption, small and medium-sized enterprises (SMEs), technology-organization-environment (TOE) framework, systematic literature review, digital transformation

Received: September 10, 2024

This paper investigates the factors influencing cloud computing adoption in small and medium-sized enterprises (SMEs) through a systematic literature review. The analysis identified twelve key factors influencing the adoption of cloud computing in SMEs. Based on the Technology-Organisation-Environment (TOE) model and the Technology Acceptance Model (TAM), a conceptual framework was developed for future research. The most important factors are cost, organisational readiness, compatibility, relative advantage and top management support. Other influential factors include security, perceived usefulness, firm size, government support, perceived ease of use, vendor support and competitive pressure. The majority of studies were conducted in Asian countries, including developing countries, limiting the generalisability of the findings to SMEs in more developed economies. This research highlights the need for cloud computing solutions that not only reduce costs and ensure high levels of security and privacy, but are also easy to use and integrate. Further research is recommended to explore these factors within SMEs in more developed economies.

Povzetek: Analizirani so dejavniki, ki vplivajo na sprejetje računalništva v oblaku v malih in srednje velikih podjetjih (MSP). S sistematičnim pregledom literature so avtorji identificirali dvanajst ključnih dejavnikov, vključno s stroški, organizacijsko pripravljenostjo, združljivostjo, relativno prednostjo in podporo vodstva. Na podlagi modela tehnologija-organizacija-okolje (TOE) in modela sprejemanja tehnologije (TAM) so razvili konceptualni okvir za prihodnje raziskave.

1 Introduction

The ongoing digital transformation is exerting increasing pressure on businesses to adapt to a rapidly changing environment. This necessitates enhanced digital connectivity and restructuring of internal business processes. Small and medium-sized enterprises (SMEs) are particularly affected by this shift, as they can enhance their growth by adopting new products, services, and business models [24]. A robust internal IT infrastructure is crucial for companies to remain competitive in this landscape [16].

Cloud computing, which refers to the on-demand provision of resources over the internet, is a technology that can facilitate this transformation. Its flexibility and scalability allow businesses to adapt more quickly to changes and needs, providing location-independent access to business processes [39, 47].

Notwithstanding the aforementioned advantages, SMEs lag behind larger enterprises in the adoption of cloud computing. This phenomenon can be attributed to the distinctive characteristics of SMEs, including owner-managed operations, informal organisational structures, and limited resources in comparison to larger corporations [36]. Consequently, a study by Taş et al. [2022] revealed that only 40% of German SMEs currently utilise cloud computing [42].

Previous research has examined the factors influencing cloud computing adoption in SMEs [15, 18, 29]. However, these reviews only include studies up to 2020, necessitating an updated investigation. This is particularly important in light of recent changes driven by digitalisation and the impact of the global pandemic.

The objective of this paper is to investigate the factors influencing the adoption of cloud computing in SMEs

Table 1: Overview of existing literature reviews

Author	Time period	Databases and # Of articles	Focus	Results
Hasan et al. [2015]	2010 – 2015	n/a; 12 articles	Factors influencing the introduction of cloud computing in SMEs.	28 factors; of importance are security, perceived usefulness, perceived ease of use, cost, compatibility, and top management support.
Jayeola et al. [2022]	2011 – 2020	ACM, Emerald Insight, IEEE, Google Scholar, ProQuest, ScienceDirect; 76 articles	Current status of data analysis techniques, analysed services, further research questions and the most important factors.	8 factors; security and privacy are the most analysed, the biggest factor is cost savings.
Nguyen and Liaw [2022]	2011 – 2020	Academia, Emerald Insight, ResearchGate, ScienceDirect, Springer Link; 30 articles	Investigation of influencing factors using the TOE framework.	28 factors; top management support, technological readiness, security concerns and relative advantage are influential.

through a systematic review of recent literature. This research aims to provide an up-to-date understanding of the key determinants of cloud computing adoption in SMEs, summarising them in a conceptual model for cloud computing adoption by SMEs. This model will inform both practitioners and researchers in the field.

The structure of this paper is as follows: Section 2 provides the conceptual background, defining key terms and models, and reviews the current state of research. Section 3 details the methodology used for the systematic review. Section 4 presents the results of the review. Section 5 discusses the findings, including the development of a conceptual model to explain cloud computing adoption by SMEs. Section 6 concludes the paper, outlining its contributions, mentioning limitations of our paper, and describing avenues for further research.

2 Background

2.1 Cloud computing and SMEs

Cloud computing refers to the on-demand provision of computing resources, including applications and infrastructure, that are flexibly and scalably delivered over a network [33, 39]. This technology enables organizations to access and utilize computing resources without the need for significant upfront investments or ongoing maintenance responsibilities.

Cloud computing offers several advantages to businesses. It allows for cost and effort reduction by outsourcing the provision and maintenance of IT infrastructures, platforms, and software to specialized providers [46]. This outsourcing enables companies to focus more on their core competencies and internal business processes [12]. Additionally, cloud computing improves resource availability, allowing access to business processes independent of location [12, 46].

Despite these benefits, some companies express reservations about integrating cloud computing into their operations. Primary concerns relate to performance, security, data protection, and reliability [41, 48]. Furthermore, the adoption of cloud computing often necessitates new competencies and tasks for technical staff, potentially requiring a restructuring of the IT department [33].

Cloud computing services are typically categorized into three main types: Infrastructure-as-a-Service (IaaS), which provides scalable IT infrastructure; Platform-as-a-Service (PaaS), offering development frameworks and

Table 2: Results of existing literature reviews

Author	Identified adoption factors
Hasan et al. [2015]	1. Security, Perceived usefulness, Perceived ease of use 2. Cost, Compatibility, Top management support 3. Attitude towards technology, innovation, Competitive pressure, Relative advantage, Complexity, Organizational competence
Jayeola et al. [2022]	1. Security and privacy 2. Cost savings 3. Relative advantage 4. Compatibility 5. Top management support 6. Competitive pressure 7. Government support 8. Awareness
Nguyen and Liaw [2022]	1. Top management support 2. Technological readiness 3. Security concerns 4. Relative advantage 5. Organizational readiness 6. Knowledge and training, Compatibility, Competitive pressure 7. Vendor support, Cost issues 8. Firm size

environments; and Software-as-a-Service (SaaS), providing software applications accessible over the internet [1, 14, 39].

Cloud computing can be deployed through various models: Private Cloud, used exclusively by a single organization; Public Cloud, available to the general public over the internet; and Hybrid Cloud, combining private and public cloud use. These models offer different levels of control, security, and flexibility to meet diverse organizational needs [8, 33, 39].

SMEs play a crucial role in many economies. In Germany, for example, SMEs account for about 99.4% of all enterprises, of which 82.4% are micro-enterprises, 14.5% are small enterprises and 2.4% are medium-sized enterprises [40]. According to the European Union definition (2003), SMEs can have up to 249 employees and an annual turnover of up to 50 million euros or a balance sheet total of up to 43 million euros [13]. Despite their numerical dominance, SMEs contribute only around 50% of GDP in developed countries [25] and, for example, only 28.7% of total turnover in Germany in 2021, although they employ a significant proportion of the workforce, with 54.8% of all employees in Germany working in SMEs [40].

The adoption of cloud computing presents both opportunities and challenges for SMEs. While cloud services offer SMEs access to advanced technologies without significant upfront investments, the adoption rate among SMEs lags behind that of larger enterprises. A recent study by Taş et al. [2022], for example, reported that 40% of German SMEs now use cloud services, indicating a growing recognition of cloud computing's potential benefits among SMEs [42]. However, this also highlights that a significant portion of SMEs have yet to adopt cloud technologies, underscoring the need for a deeper understanding of the factors influencing adoption decisions in this sector.

2.2 Theoretical background

The adoption of technological innovations like cloud computing in organizational contexts is typically examined through the lens of several theoretical models. Our paper primarily draws upon three widely recognized frameworks: the Technology-Organization-Environment (TOE) framework, the Technology Acceptance Model (TAM), and the Diffusion of Innovation (DOI) theory.

The TOE framework, proposed by Tornatzky and Fleischer [1990], posits that technological innovation adoption is influenced by three contextual elements [45]: technological, organizational, and environmental. The technological context encompasses both internal and external technologies relevant to the firm. The organizational context refers to descriptive measures such as scope, size, and managerial structure. The environmental context is the arena in which a firm conducts its business, including industry characteristics, market structure, and regulatory environment [7].

The TAM, developed by Davis [1989], focuses on individual-level technology acceptance [10]. It proposes that perceived usefulness and perceived ease of use are primary determinants of an individual's intention to use a technology. While the TAM has been widely applied in information systems (IS) research, its limited scope in addressing organizational-level factors has led researchers to often combine it with other models for a more comprehensive analysis [6].

The theory of diffusion of innovation (DOI) by Rogers [1995] describes from a sociological perspective the extent to which the introduction of an innovation can be predicted from the perception of users [34]. The contributing factors are relative advantage, compatibility, complexity, trialability and observability [34]. However, for a more holistic approach, the model should be combined with other theories and factors [17]. The characteristics of the DOI model match the technological and organisational context of the TOE model. The TOE model is additionally complemented by the environmental context and can therefore better illustrate the introduction of innovations at the firm level [30].

These models provide complementary perspectives on technology adoption. The TOE framework offers a broad organizational view, the TAM delves into individual user acceptance, and the DOI theory bridges individual and organizational adoption processes. By integrating these theoretical perspectives, researchers can develop a more nuanced understanding of the complex dynamics involved in cloud computing adoption by SMEs.

2.3 Related works

Previous systematic reviews have examined the factors influencing cloud computing adoption in SMEs, providing a foundation for our paper. This section critically analyzes these three reviews [15, 18, 29] to establish the current state of knowledge and identify gaps that we aim to address. Table 1 provides an overview of the reviews discussed.

Hasan et al. [2015] conducted a review of 12 articles from 2010 to 2015 [15]. Their analysis identified 28 factors influencing cloud adoption in SMEs. The most significant factors included perceived ease of use, perceived usefulness, security, compatibility, costs, and top management support. However, the review lacked a comprehensive description of the methodology, particularly regarding inclusion criteria and database selection, which limits the reproducibility of their findings [15].

A more recent review by Jayeola et al. [2022] examined literature from 2011 to 2020 [18]. This study provided a more robust methodological approach, clearly outlining search strategies and quality assessment procedures. Their analysis of 76 articles revealed eight key factors, with security and privacy being the most frequently studied, while cost savings demonstrated the highest statistical significance. This review also explored data analysis techniques and cloud services investigated in the primary studies, offering a broader perspective on cloud computing research in SMEs [18].

Nguyen and Liaw [2022] analyzed 30 articles published from 2011 to 2020, focusing on the application of the TOE framework [29]. Their review identified 28 factors, with top management support, technological readiness, security concerns, and relative advantage emerging as the most influential. The organizational context within the TOE framework was found to have the highest impact on adoption decisions. However, this review lacked clear information on the time frame of the studies included and did not provide detailed quality assessment criteria for the selected studies [29].

These reviews collectively highlight several consistent factors influencing cloud adoption in SMEs, including security, cost, compatibility, top management support, and relative advantage. However, there are notable differences in how these factors are labelled, ranked and emphasised across the reviews, reflecting differences in the methodological approaches, timeframes and geographical focus of the primary studies included, as shown in Table 2.

It is important to note that these reviews primarily covered literature up to 2020 (Table 1). Given the rapid pace of technological advancement and the significant global changes since then, particularly the accelerated digital transformation driven by the COVID-19 pandemic, there is a clear need for an updated review. Our paper aims to bridge this gap by focusing on the most recent literature (2021-2024) to provide a current perspective on cloud computing adoption factors in SMEs.

Furthermore, the existing reviews predominantly included studies from Asian countries and developing economies. This geographical bias raises questions about the generalizability of findings to other contexts, particularly to countries with more advanced digital infrastructures. Our paper will pay particular attention to the geographical distribution of primary studies and discuss the implications for different economic contexts.

By building upon these previous works and addressing their limitations, this systematic review aims to provide a comprehensive, up-to-date analysis of the factors influencing cloud computing adoption in SMEs. This will not only contribute to the academic understanding of technology adoption in SMEs but also provide valuable insights for policymakers and cloud service providers seeking to facilitate wider adoption of cloud technologies in the SME sector.

3 Methodology

Our paper employs a systematic literature review to investigate the factors influencing cloud computing adoption in SMEs. The systematic review approach was selected for its rigor and capacity to synthesize existing research comprehensively and objectively [21]. The review process was guided by the recommendations of Templier and Paré [2015] and adhered to the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) guidelines [28].

In order to ensure transparency and reproducibility, a predefined review protocol was devised, delineating the research questions, search strategy, inclusion and exclusion criteria, quality assessment criteria, and data extraction and synthesis methods. The search was conducted in March 2024 utilising four major academic databases: Emerald Insight, EBSCOhost, ScienceDirect, and Scopus. These databases were selected with the objective of ensuring comprehensive coverage of peer-reviewed publications in information systems and management fields.

The search string employed was as follows: *(Cloud AND Adoption AND (SME OR SMEs OR "Small and Medium Businesses" OR "Small and Medium Enterprises") AND (Factors OR Determinants))*. This string was adapted as necessary to align with the syntax requirements of each database. The search was limited to publications from 2021 to 2024 to focus on the most recent developments in the field.

The studies included in this review were selected based on the following criteria: The studies were required to have been (1) published between 2021 and 2024, to have (2) focused on the adoption of cloud computing in SMEs, to have been (3) empirical studies (either quantitative or qualitative), to have been (4) peer-reviewed journal articles or conference proceedings, and to have been (5) written in either English or German. Studies were excluded if they focused solely on large enterprises, were non-empirical, did not specifically address adoption factors, or were grey literature.

The preliminary search returned 155 studies that may have been pertinent to the review. Following the removal of duplicates and the screening of titles and abstracts, 42 articles were selected for a full-text assessment. Following

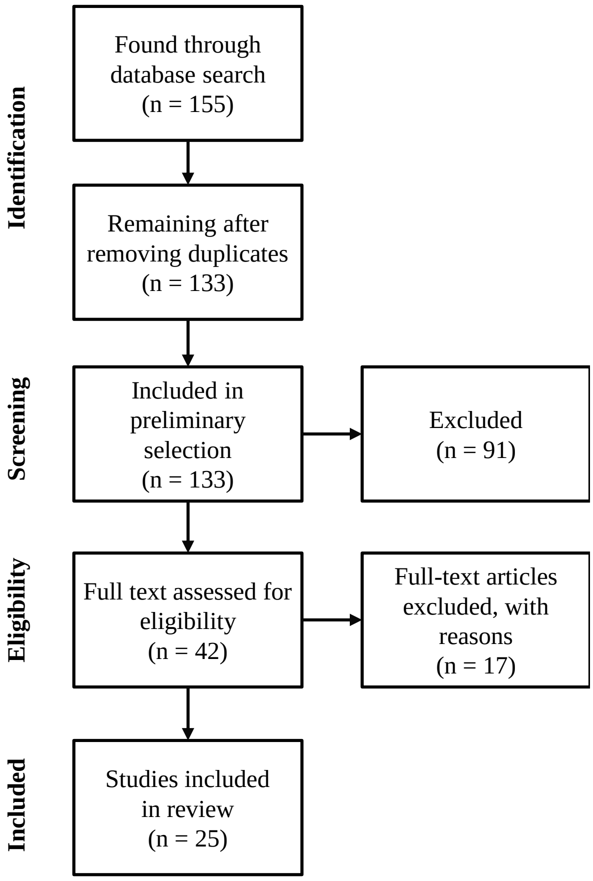

Identification
Found through database search (n = 155)
Remaining after removing duplicates (n = 133)
Screening
Included in preliminary selection (n = 133)
Excluded (n = 91)
Eligibility
Full text assessed for eligibility (n = 42)
Full-text articles excluded, with reasons (n = 17)
Included
Studies included in review (n = 25)

Figure 1: Selection process of the present systematic review in the form of a PRISMA flow chart (based on [28])

this assessment, 25 studies were ultimately included in the systematic review. Figure 1 presents a PRISMA flow diagram illustrating this selection process.

The quality of the included studies was evaluated using criteria adapted from Dybå and Dingsøyr [2008], which assessed aspects such as the clarity of the research aims, the appropriateness of the research design, and the adequacy of the data collection and analysis methods. A standardised data extraction form was used to collect relevant information from each study, including the study characteristics, the research methods, the context of cloud computing, the theoretical frameworks used, and the factors influencing cloud adoption.

The extracted data were synthesised using both quantitative and qualitative approaches. The quantitative synthesis involved calculating the frequency of factors across studies and their weighted importance based on reported significance, following the approach of Jeyaraj et al. [2006]. The qualitative synthesis involved thematic analysis to identify common themes and patterns across studies, interpreting findings in the context of existing theoretical frameworks (TOE, TAM, DOI) and identifying emerging themes or contradictions in the literature.

4 Results

4.1 Overview of included studies

Table 3 provides a comprehensive overview of all 25 studies included in this review. It details the authors, publication year, country, focus of the cloud service under investigation, research methodology, and theoretical models employed for each study.

The majority of studies (21 out of 25) were conducted in countries in Asia, with three in Africa and one in Europe. Fourteen studies examined cloud computing adoption in general, while eight focused specifically on cloud accounting, two on cloud Enterprise Resource Planning (ERP), and one on cloud Customer Relationship Management (CRM).

The quantitative data analysis methods employed were predominantly structural equation modelling (SEM), with some studies utilising multiple linear regression (MLR) or logistic regression. Qualitative research methods included semi-structured interviews and case studies. The most commonly used theoretical models were the TOE framework, the TAM, and the DOI theory.

4.2 Quantitative study results

The objective of the quantitative studies was to identify the positive, negative, or non-significant influences of various factors on the adoption of cloud computing in SMEs. A total of 171 factors (aka “independent variables”) were identified across all quantitative studies, comprising 104 for general cloud computing and 67 for specific cloud products. Following the categorisation process, 45 discrete factors were identified for further analysis (Tables 5 and 6 in the appendix).

Table 4 presents the factors that were the subject of the most extensive research, as evidenced by their appearance in at least five studies. This figure can be found in column A. Column B indicates the frequency with which

Table 3: Studies included in this literature review

Author	Country	Cloud service	Method	Model
Alasady et al. [2023]	Irak	Cloud Computing	PLS-SEM	DOI
Ali et al. [2023]	Somalia	Cloud Computing	SEM	TOE
Aligarh et al. [2023]	Indonesien	Cloud Computing	PLS-SEM	TOE
Al-Sharafi et al. [2023]	Malaysia	Cloud Computing	SEM-ANN	TOE
Ansong and Boateng [2023]	Ghana	Cloud Computing	Case study	TOE
Athambawa et al. [2023]	Sri Lanka	Cloud Computing	SEM	DOI, UTAUT2
Chen et al. [2023]	China	Cloud Computing	PLS-SEM, fsQCA	TOE
Forootani et al. [2022]	Iran	Cloud CRM	PLS-SEM	TAM, TOE
Homan and Beránek [2023]	Tschechien	Cloud Computing	Logistic regression	TOE
Kamal et al. [2023]	Malaysia	Cloud Accounting	PLS-SEM	TAM
Khayer et al. [2021]	Bangladesh	Cloud Computing	PLS-SEM	TOE, UTAUT
Lutfi [2021]	Jordanien	Cloud ERP	PLS-SEM	TAM
Lutfi [2022]	Jordanien	Cloud Accounting	PLS-SEM	TOE
Majengo and Mbise [2022]	Tansania	Cloud Computing	MLR, interviews	DOI, TOE
Matias and Hernandez [2021]	Philippinen	Cloud Computing	MLR, interviews	TOE
Mohammed et al. [2023]	Irak	Cloud ERP	PLS-SEM	DOI, TOE
Rawashdeh and Rawashdeh [2023]	Jordanien	Cloud Accounting	SEM	TOE
Rawashdeh et al. [2023]	Jordanien	Cloud Accounting	SEM	TOE
Saad et al. [2022]	Jordanien	Cloud Accounting	PLS-SEM	DOI, TOE
Sastararuji et al. [2022]	Thailand	Cloud Accounting	Interviews	TOEVO
Shetty and Panda [2023]	Indien	Cloud Computing	SEM	TAM, TCE, TOE
Sin et al. [2023]	Malaysia	Cloud Accounting	MLR	TAM
Syairudin and Nabila [2024]	Indonesien	Cloud Computing	PLS-SEM	TAM
Tawfik et al. [2023]	Oman	Cloud Accounting	SEM	TOE
Yaseen et al. [2023]	Jordanien	Cloud Computing	MLR	DOI, TOE

the factors were categorised as significant. The ratio of the frequency with which a factor was analysed and evaluated as significant is represented by the weighting from B/A in column C. Table 4 was finally sorted on the basis of this weighting. The minimum number of five occurrences (Table 5) was chosen in order to achieve a balanced and realistic weighting. Of these factors, those with a weighting of at least 0.5 (Table 6) were included in the further analysis, in accordance with the recommendations of Jeyaraj et al. [2006].

The most frequently examined factors were competitive pressure, relative advantage, top management support, compatibility, and security (Table 5). The factors with the highest weighted importance, exceeding 0.8, were cost, organisational readiness, compatibility, relative advantage, and top management support (Table 6).

A total of twelve factors were identified as being of significant importance in the context of cloud computing adoption by SMEs. These factors were selected based on a weighted importance threshold of 0.5, which was determined through a systematic review of the literature (Table 6). The twelve factors are as follows: cost, organisational readiness, compatibility, relative advantage, top management support, security, perceived usefulness, firm size, government support, perceived ease of use, vendor support, and competitive pressure.

4.3 Qualitative study results

Qualitative studies yielded a wealth of contextual insights into the factors influencing the adoption of cloud computing. Sastararuji et al. [2022] conducted semi-structured interviews with SME owners, cloud accounting service providers, and subject matter experts. The findings emphasised the crucial role of relative advantage, which was identified as a key factor leading to cost savings and increased efficiency. Complexity was identified as a negative influence, whereas compatibility was viewed in a positive light. Additionally, organisational characteristics, technological readiness, government support and competitive pressure were identified as influential factors.

Table 4: Most frequently analysed and found significant factors (aka “independent variables”) sorted by significance weight, C

Factor	A	B	C
Cost	7	6	0.86
Organizational readiness	6	5	0.83
Compatibility	11	9	0.82
Relative advantage	15	12	0.80
Top management support	15	12	0.80
Security	11	8	0.73
Perceived usefulness	6	4	0.67
Firm size	5	3	0.60
Government support	5	3	0.60
Perceived ease of use	5	3	0.60
Vendor support	7	4	0.57
Competitive pressure	17	9	0.53
Total	110	78	

In a case study presented by Ansong and Boateng [2023], a small enterprise that had adopted cloud computing was examined. The results, presented within the TOE model, identified technological readiness, technological trends, available digital resources, organisational characteristics, top management support, competitive pressure, and government support as the key influencing factors.

Matias and Hernandez [2021] found that while SMEs were aware of the benefits of cloud computing, these were not necessarily decisive factors in their decision to adopt cloud computing. Their study highlighted the importance of compatibility, cost effectiveness, and government measures in cloud computing adoption decisions.

5 Discussion

The following discussion section synthesises the findings, interprets their implications, and contextualises them within the broader landscape of cloud computing adoption research.

5.1 Synthesis and interpretation of findings

5.1.1 Geographical distribution and contextual considerations

A noteworthy finding from this review is the geographical concentration of studies. A significant bias is evident in the current research landscape, with 21 out of 25 studies (84%) conducted in Asian countries, many of which are developing economies such as Tanzania, Somalia, Indonesia, and Jordan. This concentration of studies gives rise to significant questions regarding the generalisability of findings to other contexts, particularly to more developed economies in Europe or North America.

The disparity in cloud computing adoption rates between countries serves to underscore this concern. For example, while studies indicate that cloud adoption in countries such as Jordan [2] and India [38] is progressing at a slow pace, research by Taş et al. [2022] suggests that 40% of German SMEs already utilise cloud services. This discrepancy underscores the necessity for caution when extrapolating findings from one economic context to another and highlights the importance of context-specific research in different regions.

5.1.2 Analysis of key adoption factors

Our analysis identified twelve significant factors that influence the adoption of cloud computing in SMEs. The most influential factors, based on their weighted importance, were cost, organisational readiness, compatibility, relative advantage, and top management support (Table 4).

Cost was identified as the most consistently significant factor, with a weighted importance of 0.86. This finding corroborates those of previous reviews (e.g., [18]) and highlights the pivotal role of financial considerations in the technology adoption decisions of SMEs. The high significance of cost factors suggests that cloud service providers should focus on demonstrating clear cost benefits and return on investment in order to encourage the adoption of their services among SMEs.

The second most influential factor was organisational readiness (weighted importance of 0.83). This finding emphasises the significance of internal capabilities and resources in facilitating cloud adoption. SMEs with superior technical infrastructure, qualified personnel, and financial support are more likely to adopt cloud computing [32]. This indicates that efforts to enhance the overall technological capabilities of SMEs could indirectly promote cloud adoption.

Additionally, the results indicated that compatibility (0.82) and relative advantage (0.80) were also of significant importance. The importance of compatibility indicates that cloud solutions that align well with existing business processes and values are more likely to be adopted. The high significance of relative advantage indicates that SMEs are driven by the perceived advantages of cloud computing, including enhanced business processes, flexibility, and increased productivity [4, 35].

Top management support (0.80) was also identified as a significant factor, particularly in the context of SMEs where decision-making is often centralised. This finding suggests that educational and awareness programmes targeting SME owners and top management could prove effective in promoting cloud adoption [38].

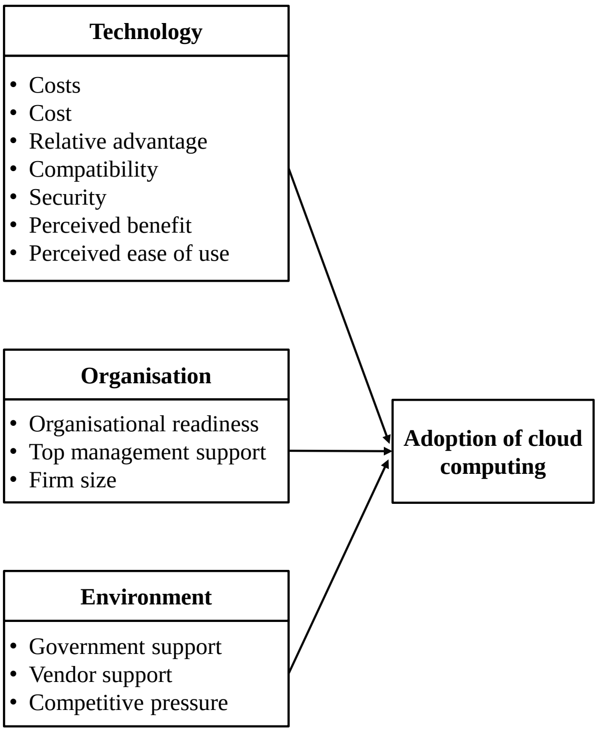
Notably, while security was frequently studied, the results were mixed (Table 5), with both positive and negative influences observed. This ambivalence may reflect the evolving perceptions of cloud security, where concerns about data protection are balanced against recognition of the robust security measures offered by cloud providers.

5.2 Comparison with previous research and theoretical implications

A comparison of our findings with those of previous systematic reviews reveals both continuities and shifts in the factors influencing cloud adoption in SMEs. The continued significance of factors such as cost, compatibility, and top management support across reviews spanning different time periods suggests that these are enduring concerns for SMEs considering cloud adoption.

However, our review also indicates some shifts in emphasis. For example, while Hasan et al. [2015] identified perceived ease of use and perceived usefulness as highly significant, our review suggests a relative decrease in the importance of these factors. This could indicate a growing familiarity with cloud technologies among SMEs, reducing usability concerns.

Figure 2: Tentative conceptual modul of cloud computing adoption factors in SMEs



The high significance of organisational readiness in our review, which was less prominent in earlier reviews, might reflect the maturing of cloud technologies and a shift in focus from technological to organisational factors in adoption decisions.

Our findings lend substantial support to the relevance of established theoretical frameworks, such as the TAM and the TOE, in understanding the adoption of cloud computing in SMEs. However, the complex interplay of factors revealed in our analysis suggests that an integrated theoretical approach combining elements from multiple frameworks may prove more effective in capturing the multifaceted nature of cloud adoption decisions in SMEs.

The prominence of cost and organisational factors in our results is consistent with the organisational context of the TOE framework. However, the significance of factors such as compatibility and relative advantage also highlights the continued relevance of the DOI theory in explaining cloud adoption. Based on our findings, we propose a tentative conceptual model (Figure 2) that combines elements from the TOE and TAM, incorporating the most significant factors identified in our review. This model provides a tentative framework for understanding cloud adoption in SMEs and can guide future research in this area.

This integrated model not only identifies the principal factors influencing the adoption of cloud computing, but also demonstrates the intricate interconnections between technological, organisational and environmental variables. It offers a more sophisticated comprehension of the adoption process, reflecting the complexities of decision-making in SMEs.

6 Conclusion

This systematic review was conducted with the objective of investigating the factors influencing the adoption of cloud computing in SMEs, with a particular focus on the most recent literature from 2021 to 2024. The analysis identified twelve key factors that significantly impact cloud adoption decisions in SMEs. The most influential factors were found to be cost, organisational readiness, compatibility, relative advantage, and top management support.

Our paper makes a number of contributions to the field. Firstly, we provide an up-to-date synthesis of the latest research on cloud adoption in SMEs, reflecting recent technological advancements and changing business environments. Secondly, the tentatively proposed conceptual model, combining elements from the TOE framework and the TAM, offers a framework for understanding the complex interplay of factors in cloud adoption decisions. This model can serve as a valuable tool for both researchers and practitioners in the field.

Our findings highlight the necessity for cloud computing solutions that not only provide cost savings and robust security measures but are also user-friendly and readily integrated into existing business processes. This indicates that cloud service providers should prioritise the development of flexible, cost-effective solutions that can be integrated seamlessly into the technological infrastructures of SMEs. For policymakers, our results indicate the potential influence of government assistance and regulatory frameworks on the adoption of cloud computing by SMEs.

It is important to acknowledge the limitations of our paper. The most significant limitation of the paper is the geographical bias in the reviewed literature. Given that 84% of the studies were conducted in Asian countries, many of which are developing economies, the generalisability of our findings to other contexts, particularly to more developed economies in Europe or North America, is limited. This geographical concentration gives rise to significant questions regarding the applicability of these findings to disparate economic and technological contexts. Furthermore, the accelerated pace of technological evolution in cloud computing may exceed the pace of academic publishing, potentially limiting the currency of some findings. The exclusion of non-English and non-German publications may also have resulted in the omission of pertinent studies from non-English-speaking regions.

These limitations indicate several potential avenues for future research. Firstly, there is a pressing need for further studies on the adoption of cloud computing in SMEs in developed economies, with a particular focus on Europe and North America. Such research would facilitate a more nuanced global perspective and enable comparative analyses across diverse economic contexts.

Secondly, future studies should investigate the temporal dynamics of factors influencing cloud adoption. Longitudinal studies could provide valuable insights into the evolution of the importance of different factors as SMEs progress through various stages of cloud adoption and as the technology itself matures.

Thirdly, the complexity of cloud adoption decisions revealed in our paper indicates a need for more in-depth qualitative research, such as case studies or ethnographic approaches, to provide a richer contextual understanding of how these factors interact in real-world settings.

Acknowledgements

Artificial intelligence (AI) tools, namely Claude 3.5 Sonnett and DeepL Write, were employed in the preparation of this paper. These AI tools were used to enhance the language, grammar, and overall readability of the manuscript. While the content, ideas, and research presented in this paper are the original work of the authors, it is important to acknowledge the use of these AI tools as a support in the writing process.

The authors have carefully reviewed and edited the AI-generated content to ensure the accuracy, integrity, and quality of the final paper. The use of AI tools was intended to improve the clarity and presentation of the research and does not diminish the academic contribution or originality of the work. The authors take full responsibility for the content and conclusions presented in this paper.

7 Appendix

A total of 104 adoption factors (aka “independent variables”) were identified through the analysis of all quantitative studies on cloud computing adoption by SMEs. In the case of specific cloud products, 67 were subjected to analysis. A total of 171 factors (“independent variables”) were thus identified. Following the categorisation process, a total of 45 factors were ultimately selected for analysis. In accordance with the methodology proposed by Jeyaraj et al. [2006], the data is presented in Table 5 and Table 6.

Table 5 provides an extended view of the number of factors (“independent variables”) that had a positive or negative effect, measured by positive or negative significance, on the cloud computing adoption by SMEs (“dependent variable”). The table was created based on Lacity et al. [2016] and Könning et al. [2019].

Table 6 presents the 45 aggregated factors and offers an analysis of the frequency with which a factor was identified as being significant for the decision to adopt cloud computing in an SME context. Column A indicates the frequency with which this factor was analysed in studies, while column B shows the proportion of studies in which the factor was categorised as significant. The ratio of factors analysed to those evaluated as significant is presented in column C, with a weighting derived from B/A.

For comparability, the sorting of both tables 5 and 6 is consistent with the frequency with which this factor was analysed in studies.

Relative advantage and top management support each have a predominantly positive effect on the adoption of cloud computing. Companies benefit from the use of cloud computing from improved business processes, flexibility, increased productivity and cost savings, among other things [4, 35]. Executives who are aware of the benefits are more likely to prepare the company for the introduction of cloud computing [31]. As SMEs often have flat hierarchies and are owner-managed, decisions can be made more quickly by the management level [38].

Compatibility is positive eight out of eleven times, but was weighted negatively once. A lack of compatibility can inhibit the decision due to a lack of technical knowledge and the time and cost required for implementation. A high level of compatibility between existing technical infrastructures and cloud services, on the other hand, can increase the likelihood of cloud computing being introduced. This means that new services can be implemented quickly and easily.

Competitive pressure (9 out of 17 times) and technology readiness (four out of nine times) were only categorised as significant in around half of the studies. Although competitive pressure was the most frequently investigated factor, there are some factors with a higher weighting, such as costs and organisational readiness. Technological readiness also has a low weighting in terms of frequency. Nowadays, many companies are usually already at an advanced technical level due to the digital transformation, which makes it possible to implement new technologies more quickly. These offer companies a strategic advantage in a competitive and constantly changing business environment [20]. As a result, these factors may have less influence on the decision, as the companies already benefit from the conditions.

Security was negatively significant five times, positive three times and three times no significance. According to the results, security and privacy are among the most important factors taken into account when adopting cloud computing. Companies have concerns about data security and worry about the protection of their data and that of their customers [43]. In order to protect companies, appropriate security and data protection measures must be taken by cloud providers [20]. This can increase companies' trust in cloud computing and improved security measures also lead to higher acceptance [49]. A high level of security therefore has a positive influence on implementation.

Complexity was significant once positively and once negatively, but five times without significance and has a weighting of 0.29. Due to insufficient knowledge of cloud technology, companies can perceive an introduction as too complex [9]. As the factor was predominantly confirmed as insignificant, it can be assumed that SMEs can cope with the use of innovative technologies such as cloud computing.

The cost factor proved to be relevant in the course of our analysis. In six out of seven studies, it was shown to have a positive effect on the introduction of cloud computing. By using cloud services, SMEs have low costs and save money by not having to have their software developed and maintained by external IT companies [26]. The high relevance attributed to costs can also be found in the studies analysed as an adoption factor with the highest or second-highest significance [3, 4, 26, 49].

Vendor support has a weighting of 0.57. The introduction of cloud services also depends on trust in the vendor, who can provide support in the absence of technical, human and financial resources and ensure the provision and maintenance of the services [35]. In three studies, vendor support was not significant. This may be due to the fact that companies are aware of the wide range of cloud services on offer [32] and have confidence in the vendors.

The organisational readiness also has a high weighting at 0.83. A company is more likely to decide in favour of an introduction if a good technical infrastructure, qualified personnel and financial support are available [32].

Other factors with a positive influence are perceived benefits with a value of 0.67 as well as government support, perceived ease of use and company size, each with a weighting of 0.6.

Table 5: Found significant adoption factors (“independent variables”) sorted by analysis frequency

Factor	−1	N	+1	Total
Competitive pressure		8	9	17
Relative advantage		3	12	15
Top management support		3	12	15
Compatibility	1	2	8	11
Security	5	3	3	11
Technologial readiness		5	4	9
Complexity	1	5	1	7
Cost		1	6	7
Vendor support		3	4	7
Organisational readiness		1	5	6
Perceived usefulness		2	4	6
Firm size		2	3	5
Government support		2	3	5
Perceived ease of use		2	3	5
Industry		2	2	4
Suppliers		1	2	3
Self-efficacy			3	3
Awareness			2	2
Effort expectancy		1	1	2
Marketing		2		2
Performance expectancy		1	1	2
Risks		2		2
Social influence			2	2
Trust			2	2
Absorptive capacity			1	1
Selected characteristics		1		1
Sufficient resources		1		1
Decision maker’s cloud knowledge			1	1
Pressure		1		1
Attitude towards technology		1		1
Employee experience			1	1
Hedonic motivation		1		1
Information intensity		1		1
Decision maker’s innovativeness			1	1
Internet infrastructure		1		1
Competitive advantage			1	1
Cost and benefit		1		1
Mimetic effect			1	1
Service quality			1	1
Computer anxiety		1		1
Trialibility			1	1
Training		1		1
Perceived uncertainty		1		1
Resistance to change				1
Reliability and availability			1	1
Total				171

Table 6: Ratio of found significant adoption factors related to their total analyses frequency sorted by analyses frequency

Factor	A	B	C
Competitive pressure	17	9	0.53
Relative advantage	15	12	0.80
Top management support	15	12	0.80
Compatibility	11	9	0.82
Security	11	8	0.73
Technologial readiness	9	4	0.44
Complexity	7	2	0.29
Cost	7	6	0.86
Vendor support	7	4	0.57
Organisational readiness	6	5	0.83
Perceived usefulness	6	4	0.67
Firm size	5	3	0.60
Government support	5	3	0.60
Perceived ease of use	5	3	0.60
Industry	4	2	0.50
Suppliers	3	2	0.67
Self-efficacy	3	3	1.00
Awareness	2	2	1.00
Effort expectancy	2	1	0.50
Marketing	2	0	0.00
Performance expectancy	2	1	0.50
Risks	2	0	0.00
Social influence	2	2	1.00
Trust	2	2	1.00
Absorptive capacity	1	1	1.00
Selected characteristics	1	0	0.00
Sufficient resources	1	0	0.00
Decision maker’s cloud knowledge	1	1	1.00
Pressure	1	0	0.00
Attitude towards technology	1	0	0.00
Employee experience	1	1	1.00
Hedonic motivation	1	0	0.00
Information intensity	1	0	0.00
Decision maker’s innovativeness	1	1	1.00
Internet infrastructure	1	0	0.00
Competitive advantage	1	1	1.00
Cost and benefit	1	0	0.00
Mimetic effect	1	1	1.00
Service quality	1	1	1.00
Computer anxiety	1	0	0.00
Trialibility	1	1	1.00
Training	1	0	0.00
Perceived uncertainty	1	0	0.00
Resistance to change	1	1	1.00
Reliability and availability	1	1	1.00
Total	171	109	

References

[1] Ali S. Alasady, Hayder S. Hashim, and Wid A. Awadh. 2023. Nexus between Iraqi SMEs cloud computing adoption intention and firm performance: Moderating role of risk factors. *Indonesian Journal of Electrical Engineering and Computer Science* 31, 2, 1128–1135. DOI: https://doi.org/10.11591/ijeecs.v31.i2.pp1128-1135.

[2] Radwan M. Al-Dwairi, Nadia Al-Tweit, and Kholood Zyout. 2018. Factors influencing cloud-computing adoption in small and medium e-commerce enterprises in Jordan. In *Proceedings of the 2018 International Conference on Internet and e-Business*. ACM Other conferences. ACM, New York, NY, 73–78. DOI: https://doi.org/10.1145/3230348.3230370.

[3] Abdifatah F. Ali, Abdikarim A. Hassan, Husein O. Abdullahi, and Rusli H. Abdulah. 2023. Analyzing the factors influencing the adoption of cloud computing by SMEs using the SEM approach. *International Journal of Advanced and Applied Sciences* 10, 7, 66–79. DOI: https://doi.org/10.21833/ijaas.2023.07.009.

[4] Mohammed A. Al-Sharafi, Mohammad Iranmanesh, Mostafa Al-Emran, Ahmed I. Alzahrani, Fadi Herzallah, and Norziana Jamil. 2023. Determinants of cloud computing integration and its impact on sustainable performance in SMEs: An empirical investigation using the SEM-ANN approach. *Heliyon* 9, 5, e16299. DOI: https://doi.org/10.1016/j.heliyon.2023.e16299.

[5] Eric Ansong and Sheena L. Boateng. 2023. Reaching for the “cloud”: The case of an SME in a developing economy. *International Journal of E-Business Research* 19, 1, 1–17. DOI: https://doi.org/10.4018/IJEBR.319324.

[6] Hart O. Awa, Ojiabo U. Ojiabo, and Bartholomew C. Emecheta. 2015. Integrating TAM, TPB and TOE frameworks and expanding their characteristic constructs for e-commerce adoption by SMEs. *Journal of Science & Technology Policy Management* 6, 1, 76–94. DOI: https://doi.org/10.1108/JSTPM-04-2014-0012.

[7] Jeff Baker. 2012. The technology–organization–environment framework. In *Information systems theory*. *Explaining and predicting our digital society*, Yogesh K. Dwivedi, Michael R. Wade and Scott L. Schneberger, Eds. Springer, New York, NY, 231–245. DOI: https://doi.org/10.1007/978-1-4419-6108-2_12.

[8] Jürgen Biebl. 2012. Wofür steht cloud computing eigentlich? *Wirtschaftsinformatik & Management* 4, 1, 22–29. DOI: https://doi.org/10.1365/s35764-012-0104-3.

[9] Mei Chen, Haiqing Wang, Yikai Liang, and Ge Zhang. 2023. Net and configurational effects of determinants on cloud computing adoption by SMEs under cloud promotion policy using PLS-SEM and fsQCA. *Journal of Innovation & Knowledge* 8, 3, 100388. DOI: https://doi.org/10.1016/j.jik.2023.100388.

[10] Fred D. Davis. 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly* 13, 3, 319–340. DOI: https://doi.org/10.2307/249008.

[11] Tore Dybå and Torgeir Dingsøyr. 2008. Empirical studies of agile software development: A systematic review. *Information and Software Technology* 50, 9-10, 833–859. DOI: https://doi.org/10.1016/j.infsof.2008.01.006.

[12] Mohamed A. A. Elmonem, Eman S. Nasr, and Mervat H. Geith. 2016. Benefits and challenges of cloud ERP systems: A systematic literature review. *Future Computing and Informatics Journal* 1, 1-2, 1–9. DOI: https://doi.org/10.1016/j.fcij.2017.03.003.

[13] Europäische Union. 2003. *Amtsblatt der Europäischen Union (2003/361/EG)*: *Empfehlung der Kommission vom 6. Mai 2003 betreffend die Definition der Kleinstunternehmen sowie der kleinen und mittleren Unternehmen*.

[14] Joel Gibson, Darren Eveleigh, Robin Rondeau, and Qing Tan. 2012. Benefits and challenges of three cloud computing service models. *Fourth International Conference on Computational Aspects of Social Networks*, 198–205. DOI: https://doi.org/10.1109/CASoN.2012.6412402.

[15] Lujain M. Hasan, Laith A. Zgair, Adoum A. Ngotoye, Hisham N. Hussain, and Chya Najmuldeen. 2015. A review of the factors that influence the adoption of cloud computing by small and medium enterprises. *Scholars Journal of Economics, Business and Management*, 842-848. DOI: https://doi.org/10.36347/sjebm.2015.v02i08.009.

[16] Raoul Hentschel, Christian Leyh, and Marc Egner. 2020. Motivationsfaktoren für oder gegen einen einsatz von cloud-lösungen in kleinstunternehmen. *HMD Praxis der Wirtschaftsinformatik* 57, 5, 961–975. DOI: https://doi.org/10.1365/s40702-020-00650-7.

[17] Erind Hoti. 2015. The technological, organizational and environmental framework of IS innovation adaption in small and medium enterprises: Evidence from research over the last 10 years. *International Journal of Business and Management* 3, 4, 1–14. DOI: https://doi.org/10.20472/BM.2015.3.4.001.

[18] Olakunle Jayeola, Shafie Sidek, Shouvik Sanyal, Syed I. Hasan, Nguyen B. An, Samuel-Soma Mofoluwa Ajibade, and Thi T. H. Phan. 2022. Government financial support and financial performance of SMEs: A dual sequential mediator approach. *Heliyon* 8, 11, e11351. DOI: https://doi.org/10.1016/j.heliyon.2022.e11351.

[19] Anand Jeyaraj, Joseph W. Rottman, and Mary C. Lacity. 2006. A review of the predictors, linkages, and biases in IT innovation adoption research. *Journal of Information Technology* 21, 1, 1–23.

DOI: https://doi.org/10.1057/palgrave.jit.2000056.

[20] Abul Khayer, Nusrat Jahan, Md. N. Hossain, and Md. Y. Hossain. 2021. The adoption of cloud computing in small and medium enterprises: A developing country perspective. *VINE Journal of Information and Knowledge Management Systems* 51, 1, 64–91. DOI: https://doi.org/10.1108/VJIKMS-05-2019-0064.

[21] Barbara Kitchenham and Stuart M. Charters. 2007. *Guideliness for performing systematic literature reviews in software engineering*. EBSE Technical Report EBSE-2007-01. Keele Durham University, U.K.

[22] Michael Könning, Markus Westner, and Susanne Strahringer. 2019. A systematic review of recent developments in IT outsourcing research. *Information Systems Management* 36, 1, 78–96. DOI: https://doi.org/10.1080/10580530.2018.1553650.

[23] Mary C. Lacity, Shaji A. Khan, and Aihua Yan. 2016. Review of the empirical business services sourcing literature: An update and future directions. *Journal of Information Technology* 31, 3, 269–328. DOI: https://doi.org/10.1057/jit.2016.2.

[24] Christian Leyh and Katja Bley. 2016. Digitalisierung: Chance oder risiko für den deutschen mittelstand?: Eine studie ausgewählter unternehmen. *HMD Praxis der Wirtschaftsinformatik* 53, 1, 29–41. DOI: https://doi.org/10.1365/s40702-015-0197-2.

[25] Diaan-Yi Lin, Sruthi N. Rayavarapu, Karim Tadjeddine, and Rebecca Yeoh. 2022. *Beyond financials*: *Helping small and medium-size enterprises thrive* (2022). Retrieved from https://www.mckinsey.com/industries/public-sector/our-insights/beyond-financials-helping-small-and-medium-size-enterprises-thrive.

[26] Riziki M. Majengo and Mercy Mbise. 2022. Determinants of software-as a-service adoption by small and medium enterprises in Tanzania. *African Journal of Information Systems* 14, 1, 23–43.

[27] Junrie B. Matias and Alexander A. Hernandez. 2021. Cloud computing adoption intention by MSMEs in the Philippines. *Global Business Review* 22, 3, 612–633. DOI: https://doi.org/10.1177/0972150918818262.

[28] David Moher, Alessandro Liberati, Jennifer Tetzlaff, Douglas G. Altman, and The PRISMA Group. 2009. Preferred reporting items for systematic reviews and meta-analyses. *PLOS Medicine* 6, 7, 1–8. DOI: https://doi.org/10.1371/journal.pmed.1000097.

[29] Giang T. Nguyen and Shu-Yi Liaw. 2022. Understanding the factors affecting the small and medium enterprises adoption of cloud computing: A literature review. *International Journal of Business, Management and Economics* 3, 2, 149–162. DOI: https://doi.org/10.47747/ijbme.v3i2.635.

[30] Tiago Oliveira and Maria F. Martins. 2011. Literature review of information technology adoption models at firm level. *The Electronic Journal Information Systems Evaluation* 14, 1, 110–121.

[31] Tiago Oliveira, Manoj Thomas, and Mariana Espadanal. 2014. Assessing the determinants of cloud computing adoption: An analysis of the manufacturing and services sectors. *Information & Management* 51, 5, 497–510. DOI: https://doi.org/10.1016/j.im.2014.03.006.

[32] Awni Rawashdeh and Badi S. Rawashdeh. 2023. The effect cloud accounting adoption on organizational performance in SMEs. *International Journal of Data and Network Science* 7, 1, 411–424. DOI: https://doi.org/10.5267/j.ijdns.2022.9.005.

[33] Jonas Repschläger, Danny Pannicke, and Rüdiger Zarnekow. 2010. Cloud computing: Definitionen, geschäftsmodelle und entwicklungspotenziale. *HMD Praxis der Wirtschaftsinformatik* 47, 5, 6–15. DOI: https://doi.org/10.1007/BF03340507.

[34] Everett M. Rogers. 1995. Diffusion of innovations. *New York: Free Press*.

[35] Mohamed Saad, Abdalwali Lutfi, Mohammed A. Almaiah, Ahmad F. Alshira'h, Malek H. Alshirah, Hamza Alqudah, Akif L. Alkhassawneh, Adi Alsyouf, Mahmaod Alrawad, and Osama Abdelmaksoud. 2022. Assessing the intention to adopt cloud accounting during COVID-19. *Electronics* 11, 24, 4092. DOI: https://doi.org/10.3390/electronics11244092.

[36] Noor A. Salleh, Husnayati Hussin, Mohd. A. Suhaimi, Asma M. Ali, and Asma Md Ali. 2018. A systematic literature review of cloud computing adoption and impacts among small medium enterprises (SMEs). *International Conference on Information and Communication Technology for the Muslim World*, 278–284. DOI: https://doi.org/10.1109/ICT4M.2018.00058.

[37] Dalinee Sastararuji, Danupol Hoonsopon, Pongsakorn Pitchayadol, and Pimsiri Chiwamit. 2022. Cloud accounting adoption in Thai SMEs amid the COVID-19 pandemic: An explanatory case study. *Journal of Innovation and Entrepreneurship* 11, 43. DOI: https://doi.org/10.1186/s13731-022-00234-3.

[38] Jayalaxmi P. Shetty and Rajesh Panda. 2023. Cloud adoption in Indian SMEs: An empirical analysis. *Benchmarking: An International Journal* 30, 4, 1345–1366. DOI: https://doi.org/10.1108/BIJ-08-2021-0468.

[39] Daniel Sonnleitner and Markus Westner. 2017. Quo vadis IT infrastructure: Decision support for cloud computing adoption from a business perspective. In *UK Academy for Information Systems Conference Proceedings 2017*.

[40] Statistisches Bundesamt. 2024. *55 % in kleinen und mittleren Unternehmen tätig* (May 2024). Retrieved June 3, 2024 from https://www.destatis.de/DE/Themen/Branchen-Unternehmen/Unternehmen/Kleine-Unternehmen-Mittlere-Unternehmen/aktuell-beschaeftigte.html.

[41] Nabil Sultan. 2010. Cloud computing for education: A new dawn? *International Journal of*

Information Management 30, 2, 109–116. DOI: https://doi.org/10.1016/j.ijinfomgt.2009.09.004.

[42] Serpil Taş, Lukas Wiewiorra, and Claus Mayerböck. 2022. *Strategische bedeutung von clouddiensten für die digitale Souveränität von KMU. Teil 3 - Empirische erhebung KMU zu clouddiensten und digitaler souveränität* (2022). Retrieved May 29, 2024 from https://www.econstor.eu/bitstream/10419/268807/1/1833893360.pdf.

[43] Omar I. Tawfik, Omar Durrah, Khaled Hussainey, and Hamada E. Elmaasrawy. 2023. Factors influencing the implementation of cloud accounting: evidence from small and medium enterprises in Oman. *Journal of Science and Technology Policy Management* 14, 5, 859–884. DOI: https://doi.org/10.1108/JSTPM-08-2021-0114.

[44] Mathieu Templier and Guy Paré. 2015. A framework for guiding and evaluating literature reviews. *Communications of the Association for Information Systems* 37, 6, 112–137. DOI: https://doi.org/10.17705/1CAIS.03706.

[45] Louis G. Tornatzky and Mitchell Fleischer. 1990. *The processes of technological innovation*. D.C. Heath & Company, Lexington.

[46] Frederik Wulf, Tobias Lindner, Markus Westner, and Susanne Strahringer. 2021. IaaS, PaaS, or SaaS? The why of cloud computing delivery model selection: Vignettes on the post-adoption of cloud computing. In *Proceedings of the 54th Hawaii International Conference on System Sciences. DOI: https://doi.org/*10.24251/HICSS.2021.758.

[47] Frederik Wulf, Markus Westner, Maximilian Schön, Susanne Strahringer, and Claudia Loebbecke. 2019. Preparing for a digital future: Cloud strategy at Continental AG. In *ICIS 2019 Proceedings*.

[48] Frederik Wulf, Markus Westner, and Susanne Strahringer. 2021. Cloud computing adoption: A literature review on what is new and what still needs to be addressed. *Communications of the Association for Information Systems* 48, 1, 523–561. DOI: https://doi.org/10.17705/1CAIS.04843.

[49] Husam Yaseen, Ahmad S. Al-Adwan, Muhamud Nofal, Hazar Hmoud, and Radwan S. Abujassar. 2023. Factors influencing cloud computing adoption among SMEs: The Jordanian context. *Information Development* 39, 2, 317–332. DOI: https://doi.org/10.1177/02666669211047916.

Learning Algorithm for LesserDNN, a DNN with Quantized Weights

Masashi Takemoto[1], Yasutake Masuda[1], Jingyong Cai[2] and Hironori Nakajo[2]
[1]BeatCraft, Inc. Tokyo Japan
[2]Tokyo University of Agriculture and Technology Tokyo, Japan
E-mail: lesser@beatcraft.com, masuda@beatcraft.com, kkkluoruo@hotmail.com, nakajo@cc.tuat.ac.jp

Keywords: Deep neural network, machine learning, simulated annealing, weight quantization

Received: September 12, 2024

This paper presents LesserDNN, a model that uses a set of floating-point values {-1.0, -0.5, -0.25, -0.125, -0.0625, 0.0625, 0.125, 0.25, 0.5, 1.0} as quantized weights, and a new learning algorithm for the proposed model. In previous studies on deep neural networks (DNNs) with quantized weights, because DNNs employ the gradient descent method as their learning algorithm, quantized weights were applied only during the inference stage. Due to differentiability properties, quantized weights cannot be used when the gradient descent method is applied during training. To address this issue, we devised an algorithm based on simulated annealing. Since simulated annealing has no differentiability requirements, LesserDNN can utilize quantized weights during training. With the use of quantized weights and this simulated annealing-based algorithm, the learning process becomes a combinatorial problem. The proposed algorithm was applied to train networks on the MNIST handwriting dataset. The tested models were trained with the simulated annealing-based algorithm and quantized weights, achieving the same level of accuracy as gradient descent-based comparison methods. Additionally, we conducted tests using the CIFAR-10 dataset, and achieved the good results to demonstrate the algorithm. Thus, LesserDNN has a simple design and small implementation scale because backpropagation is not applied. Moreover, this model achieves a high accuracy

Povzetek: LesserDNN je model globoke nevronske mreže z utežmi, kvantiziranimi na nabor vrednosti -1,0; -0,5; -0,25; -0,125; -0,0625; 0,0625; 0,125; 0,25; 0,5; 1,0. Za učenje tega modela so razvili algoritem, ki temelji na simuliranem žarjenju, saj gradientni spust zaradi lastnosti diferencirljivosti ni primeren za kvantizirane uteži. Ta pristop omogoča uporabo kvantiziranih uteži že med učenjem.

1 Introduction

Deep neural networks (DNNs) have contributed substantially to the development of machine learning and various machine learning-based applications. To improve accuracy, deeper networks, more complex structures, and more neurons have been introduced in various models. However, these large models are not suitable for deployment in small devices. To address this issue, weight quantization approaches have been introduced. To reduce model size while maintaining accuracy, less precise weights are used only for inference, while training still requires full-precision weights. Weight quantization successfully reduces the memory requirements and computational costs of DNNs while maintaining the same accuracy as full-precision weight DNNs. Lower precision weights are not applied during the DNN training process because of the learning algorithm, namely, the gradient descent method. To fully utilize less precise weights, a new learning algorithm must be developed.

Therefore, we devised LesserDNN, a novel approach that uses a set of floating-point values {- 1.0, -0.5, -0.25, -0.125, -0.0625, -0.0625, 0.125, 0.25, 0.5, 1.0} as quantized weights instead of arbitrary values such as continuous floating-point numbers, where w≦1.0 and w≧-1.0. (w: a weight). Moreover, we developed a new learning algorithm. We implemented a framework that utilizes these quantized weights during both the training and inference stages; thus, LesserDNN can build arbitrary networks of freely stacked layers containing neurons with quantized weights.

In conventional DNNs, the weights are set as arbitrary real numbers between -1.0 and 1.0, and infinite combinations of these weights are possible during training. On the other hand, in contrast to traditional DNNs, the learning process of LesserDNN is formulated as a combinatorial problem because the number of combinations of quantized weights is finite.

Although the number of combinations is mathematically finite, it is intractable for typical computer systems; thus, it is impossible to search for the optimal solution in a round-robin fashion within a practical time frame. Therefore, we devised a learning algorithm using simulated annealing (SA). SA imitates the most settled arrangement of heated molecules as they cool, with the weights representing the molecules and the number of selections representing the temperature. Randomly selected weights are changed and updated only when these changes improve the results. The combinations are evaluated according to the difference between the current result and the correct answer. The optimization proceeds by iterating; thus, this difference decreases, and the global optimum is eventually reached. The background of this study is described in Section 2. LesserDNN is explained in Section 3, and the details of the algorithm are presented in Section 4. The experiments conducted on the MNIST handwriting dataset are explained in Section 5, and the results are reported in Section 6, The results of additional experiments on the CIFAR-10 dataset is explained in Section 7. Then, we have discussions in Section 8.

The code of LesserDNN is available online at `https://github.com/BeatCraft/LDNN`, and the code for experiments also is available online at `https://github.com/BeatCraft/LDNN-mnist`.

2 Background

Weight quantization has been proposed as a method of increasing the inference speed while reducing memory footprint. Weight quantization involves statistically analyzing trained DNNs and replacing values. In previous studies, Vanhoucke et al. [1] initially applied weight quantization to reduce the computational burden of DNNs. In each layer, the weights were normalized to a signed integer in the range of -127 to 127, and the activations were quantized as 8-bit integers. As a result, the total memory footprint of the improved network was approximately 3 or 4 times smaller than that of the original network. Compared with DNNs, networks that applied quantization exhibited recognizable improvements in terms of speed while maintaining accuracy.

Similar to the initial study, many weight quantization studies have applied dynamic range quantization; however, there are some notable studies on fixed-point weight quantization. Courbariaux et al. [4] introduced BinaryConnect, in which the weights are aggressively reduced to a single bit (-1 or 1). Because of these binarized weights, approximately two-thirds of the multiplication operations can be replaced by addition and substitution operations. As the calculations are greatly simplified, the training speed is 3 times faster, yet the accuracy is reduced by only 19%. To improve the accuracy of BinaryConnect, Li et al. [5] developed ternary weight networks. These networks added zero as a binarized weight and introduced a threshold-based ternary function to transform full-precision weights to ternary weights. The threshold value for the ternary weights was determined by optimizing the threshold-based ternary function. This function minimizes the Euclidian distance between full-precision weights and ternary weights with a scaling factor. Since ternary weight networks require 2-bit storage for each weight unit, their model compression rates are higher than those of full-precision weight models. The experimental results showed that when compared with full-precision weight models, the accuracies of ternary weight networks were reduced by 0.4% or less.

The results of weight quantization experiments have been remarkable thus far; however, previous studies have had difficulty testing weight quantization models with quantized weights only. In general, these studies statistically analyze trained DNNs and replace values; thus, training is initially performed with full-precision weights. The full-precision weights are replaced by quantized weights after training, and these quantized weights applied only during the inference stage to retain high accuracy.

Vanhoucke et al. [1] applied quantization in pretrained networks. Courbariaux et al. [4] and Li et al. [5] employed binary and ternary weights (lower precision weights) in forward and back propagation and applied full-precision weights in the gradient method. In these weight quantization studies, quantized weights were not fully used during training.

Weight quantization approaches require full-precision weights because the gradient descent method is used the optimization method in the training process. Because of differentiability properties, DNNs are not differentiable with respect to quantized weights. As the changes in the weights are large, the gradient descent method cannot accurately determine the gradient and thus cannot differentiate changes in the weights. Thus, quantized weights cannot be used in the gradient descent method. To address this issue, this paper applies a non-gradient-based approach, namely, an SA-based method.

SA is known as a combinatorial optimization method. In 2017, Mousavi et al. [3] applied SA to an artificial neural network (ANN). In their empirical analysis, the ANN/SA model outperformed the ANN model. However, as discussed in the paper, the ANN/SA model had large time costs. As the weights are expressed as floating-point numbers, the number of combinations in the model is enormous and practically infinite, and the computations cannot be completed within a practical time range.

The combinatorial optimization method also faces this issue. For instance, as an implementation limitation, the number of combinations is varied between -1.0 and 1.0 in increments of 0.01, which is considered the precision level of the weights. The number of combinations is 200^n, where n is the number of weights in the model. For example, a small DNN with 10000 weights contains 200^{10000} combinations. Considering the large number of combinations, small DNNs with medium-precision weights have reduced time costs while retaining the large computational power required to complete the training process. In typical experiments, small DNN models have more than 1000 weights, and the number of combinations easily surpasses this example.

To address this issue, quantized weights are introduced. The quantized weights are designed as a set of sequential numbers, where the current number is divided by 2 to determine the subsequent number, such as {-1.0, -0.5, -0.25, -0.125, -0.0625, 0.0625, 0.125, 0.25, 0.5, 1.0}, because we found that logarithmic quantization of the weights achieved better classification results than linear quantization of the weights in a previous study [2]. This approach reduces the level of precision of each weight from 200 to 10.

Thus, when quantized weights are used, the total number of combinations in the DNN is 10^{10000}. While this is still a large number, it is 20^{10000} times smaller than 200^{10000}, and current computer systems can likely handle this number of combinations. When an SA-based algorithm is applied to a DNN, a set of quantized weights can be used for training because the gradient method is not used as the optimizer.

3 LesserDNN

LesserDNN includes layers, neurons, and weights and has basic mechanisms such as inference, activation functions, loss functions, and batches. LesserDNN is similar to existing DNNs, except that LesserDNN has no backpropagation. These characteristics and other differences, such as the lack of bias, and how batches are handled, are explained in this chapter.

The output of a neuron in a DNN is calculated by adding a $bias$ to the summation of the product, as shown in (1), where x is the input value and w are the weights; then, the output value is passed to an activation function such as ReLU (2). ReLU and leaky ReLU (3) can be selected as activation functions in each layer. In (1), y is the output of the neuron, and a in (3) is a coefficient that depends on the actual network applied in a given problem.

$$y = \sum_{i=1}^{n}(x_i \cdot w_i) + bias \tag{1}$$

$$f(x) = \begin{cases} x & (x \geqq 0) \\ 0 & (x < 0) \end{cases} \tag{2}$$

$$f(x) = \begin{cases} x & (x \geqq 0) \\ a \cdot x & (x < 0) \end{cases} \tag{3}$$

The bias has two functions: adjusting the output range of the neuron and ensuring that the output value is differentiable. Ensuring that a value is differentiable means guaranteeing that that value is not set to zero. LesserDNN has no bias since there is no need to consider whether the output of the neuron is differentiable since the gradient descent method is not used for training and layer normalization is applied. Layers containing neurons use a function to normalize the output. Inference is executed by performing calculations sequentially starting with the input layer. A layer receives the output of the previous layer, executes calculations and passes the results to the next layer. This basic propagation process is repeated until the output layer is reached. In the output layer, the softmax function may be applied. For classification problems, the output of the final layer is the probability. The softmax function adjusts the individual values in the output layer to ensure that the sum of the outputs is equal to 1.0.

The loss function is a function that is used to determine the magnitude of the discrepancy between the correct and predicted values. LesserDNN selects either the mean squared error (MSE) or the cross entropy (CE) to evaluate the model. The MSE is defined in (4), where n is the number of data points, y is the correct value, and $\hat{y}$ is the predicted value. In classification problems, n is the number of classes. The CE is defined in (5), where $p(x)$ is the correct probability and $q(x)$ is the predicted probability.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4}$$

$$H(p,q) = -\sum_{x} p(x) \cdot \log q(x) \tag{5}$$

The calculations are performed in each layer, and basic functions are implemented to take advantage of hardware systems such as GPUs or multicore CPUs. LesserDNN adapts CUDA [10] and OpenCL [9] to support GPUs. CUDA is an advanced computation library for NVIDIA [8] GPUs. OpenCL is an open standard for parallel programming on systems with different types of hardware, such as multicore CPUs, FPGAs, and GPUs. The input data for training and testing are grouped in two-dimensional arrays and treated as batches. The inference results on the batched input data are evaluated in bulk, and the average of the results is considered the optimization improvement for that batch.

A batch can be assigned to multiple processes by adapting a message passing library (MPI). An MPI is a standard for distributed-memory parallel processing, and Open MPI [7] is available as an implementation. Open MPI supports data transfer and synchronization between processes not only on single computers but also on multiple computers connected in a network. Thus, very large batches that consume considerable working memory can be handled.

4 Learning algorithm

The learning algorithm of LesserDNN includes triple nested loops, challenge loops, cooling loops, and a main loop, as shown in Fig. 1. The challenge loops determine which weights to update, and the cooling loop is crucial for applying the SA function in the learning algorithm. The cooling loop controls the temperature, which represents the number of weights to update during each iteration. Then, the main loop iterates to the next cooling loop. More iterations are required for more difficult problems.

4.1 Challenge loop

The challenge loop is the innermost loop and the core of the algorithm as shown in Fig. 2, and the pseudocode is shown in Algorithm 1. It includes four basic functions: selecting

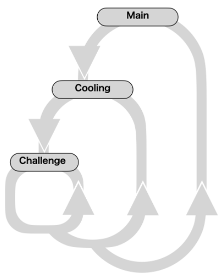


Figure 1: The triple nested loops

the weights, updating the network, evaluating the network, and reversing changes.

Before the loop, all weights in the LesserDNN network are initialized by randomly selecting a value in the set $\{$-1.0, -0.5, -0.25, -0.125, -0.0625, -0.0625, 0.125, 0.25, 0.5, 1.0$\}$. Inference is performed on the training data, and the difference between the current result and the correct answer is calculated as the loss function, Λ using either the MSE or CE. The loop takes a positive integer as a variable, and the number of weights to handle during each iteration is N. Then, N weights are randomly selected and changed, and the loss function, λ, is obtained. Since the training data include batches with multiple data points, multiple values are obtained. The average of these values is determined as the evaluation result for that batch. If λ is smaller than Λ,, the changed weights are maintained; otherwise, the changes are discarded. The loop stores ρ, which is the ratio of the number of successful updates to the total number of trials. The minimum number of iterations is set to 50 by default, and the loop is terminated when the hit rate falls below 1%.

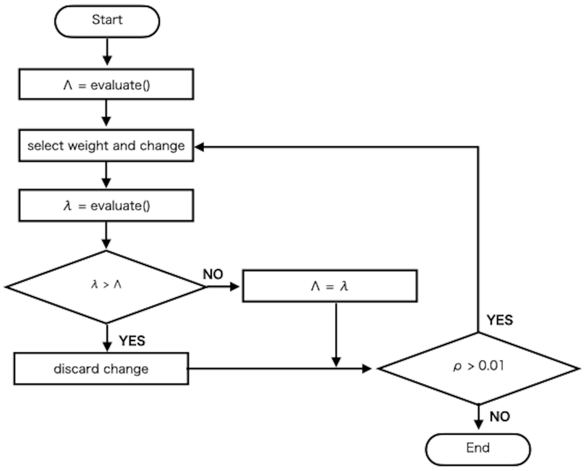


Figure 2: Flowchart of the challenge loop

Algorithm 1 Challenge Loop

```
function challenge($\Lambda$, $N$)
    $cnt \leftarrow 0$
    $hit \leftarrow 0$
    $\rho \leftarrow 1.0$
    while $\rho > 0.01$ do
        select($N$)            ▷ select N weights randomly
        update()
        $\lambda \leftarrow evaluate()$
        if $\lambda < \Lambda$ then
            $hit \leftarrow hit + 1$
            $\Lambda \leftarrow \lambda$
            keep the changes
        else
            discard the changes
        end if
        $cnt \leftarrow cnt + 1$
        $\rho \leftarrow hit/cnt$
    end while
    return $\Lambda$
end function
```

4.2 Cooling loop

The cooling loop controls N in the iterations of the challenge loop, as shown in Fig. 3, and the pseudocode is shown in Algorithm 2. N is the number of weights to be updated at once in the challenge loop. N is calculated according to the total number of weights M and the temperature in the cooling loop K.

The maximum value of temperature, θ is also calculated according to M. In the cooling loop, K decreases from θ to 0 as the temperature of the heated material decreases over time. In the cooling loop, K decreases as N decreases. We set the maximum number of weights to be changed during each iteration to 1%, and the maximum temperature, θ, was calculated as $\theta = \log_\epsilon(0.01 \cdot M)$, where ϵ is set to 2 as a default value.

Therefore, the number of weights to be changed at once at temperature k is $N = \epsilon^k$. ϵ is a hyperparameter that controls the convergence speed. The larger the value of ϵ is, the more rapidly the optimization process proceeds; thus, the possibility of falling into a local optimum increases. The appropriate range for ϵ is 1.1 to 2.0. The cooling loop starts at θ and is repeated until K reaches 1.

4.3 Main loop

The main loop is a loop that simply iterates the cooling loop, as shown in Algorithm 3. The number of iterations varies depending on the complexity of the problem. The number of iterations also depends on the number of training samples (batch size). Because LesserDNN is an SA-based method, the number of iterations is a hyperparameter.

In LesserDNN, the cycle of select(), update(), and evaluate() in the challenge loop must be performed sequentially. The computational costs of the main loop and cooling loop

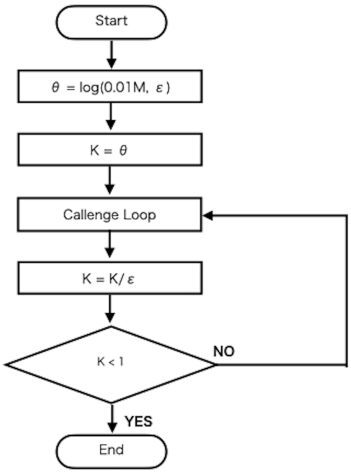


Figure 3: Flowchart of the coolong loop

Algorithm 2 Cooling Loop

```
function Cooling
    $M \leftarrow total\ number\ of\ the\ weights$
    $\epsilon \leftarrow 2$
    $\theta \leftarrow \log_{\epsilon}(0.01 \cdot M)$
    $\Lambda \leftarrow evaluate()$
    $K \leftarrow \theta$
    while $K >= 1$ do
        $N \leftarrow int(K)$
        $\Lambda \leftarrow challenge(\Lambda, N)$
        $K \leftarrow K/\epsilon$
    end while
end function
```

are low and may not change with the complexity of the problem or the batch size. The evaluation function, which includes the inference and loss functions, has the largest computational cost. The evaluation function can be parallelized by assigning divided batches to multiple processes, thus allowing LesserDNN to maintain the cycle in a single sequence and distribute the computational costs to as many processes as the hardware allows.

Algorithm 3 Main Loop

```
function Main($n$)                    ▷ $n$ : number of iterations
    for $i < n$ do
        cooling()
    end for
end function
```

5 Experiments

To validate the learning algorithm and examine the characteristics of LesserDNN, we conducted experiments using an MNIST dataset.

We constructed a network with the same layer and neuron configuration in TensorFlow and performed the same experiments for comparison. The performance was evaluated with stochastic gradient descent (SGD) and adaptive moment estimation (Adam) as backpropagation algorithms. The SGD algorithm is a first-order iterative optimization algorithm for determining the local minimum of a differentiable function. The strategy involves determining the steepest descent in a large or infinite space. By repeatedly applying the strategy, the algorithm eventually finds a local minimum. Adam was developed by Kingma et al. [6]. Adam is a stochastic gradient-based optimization method that calculates the exponential average of the gradient and the squared gradient and adapts the learning rate for each weight in the neural network. The hyperparameters control the decay rates of these moving averages. The moving averages are estimated according to the mean of the uncertain variance of the gradient.

The MNIST dataset is a well-known classification example in machine learning that contains 60000 training images and 10000 test images. These images are 28×28 pixel 8-bit grayscale images. All the images contain handwritten figures, namely, the digits 0 to 9, shown as Fig. 4. DNNs are trained with the training images and evaluated with the test images. A very basic structure, namely, a fully connected network with 2 hidden layers, was used. The input layer had 784 neurons to transform the 28×28 8-bit grayscale image to 784 floating-point values in the range of 0.0 to 1.0. The output layer had 10 neurons corresponding to the 10 classes.

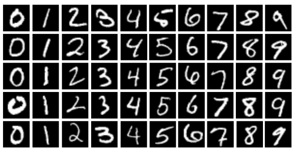


Figure 4: Examples of MNIST Dataset

5.1 Network size

The number of neurons in the two hidden layers was initially set to 256 and then changed to 128, 64, and 32.

5.2 Number of iterations

The number of iterations was set to 100. The accuracy and CE of the network, which has two hidden layers with 256 neurons, were determined with a learning algorithm that was iterated 100 times.

5.3 Training batch size

The networks with two 256-neuron hidden layers were trained on MNIST datasets with different numbers of images: 250, 500, 750, 1000, 2000, 3000, 4000, 5000, 10000, 20000, 40000, and 60000. The number of iterations was set to 100 for all configurations.

5.4 Base of temperature, ϵ

ϵ is a hyperparameter of LesserDNN that controls the speed of the temperature descent during each iteration of the SA-based learning algorithm. The networks with two 256-neuron hidden layers were trained with different ϵ values, including 2.00, 1.50, 1.25, and 1.10, to assess how this value influences training.

The number of iterations was set to 100 for all networks.

6 Results

6.1 Network size

Table 1 shows the results of training LesserDNN models with different network sizes and the results of equivalent networks in TensorFlow for comparison.

6.2 Number of iterations

Fig. 5 shows the changes in the accuracy and CE as the learning algorithm is iterated 100 times in the 256 network.

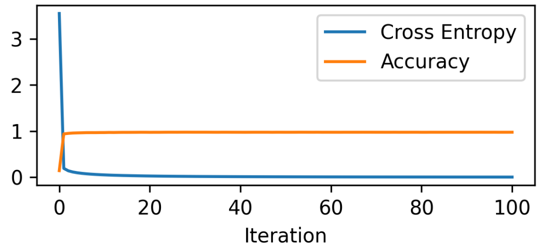


Figure 5: Changes in the accuracy and CE versus the number of iteration on the MNIST dataset

6.3 Training batch size

Table 2 shows the accuracies of networks trained with 250, 500, 750, and 1000 training images and the results of TensorFlow for comparison.

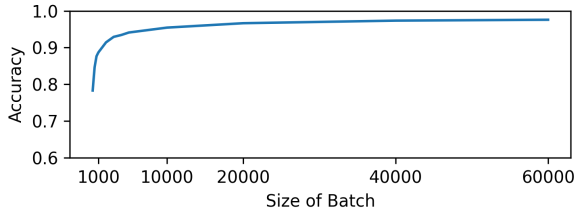


Figure 6: Changes in accuracy versus batch size for MNIST

6.4 Base of temperature, ϵ

Fig. 7 shows how the accuracy changes with the parameter, ϵ. The range of the Y axis varies from 0.90 to 1.00 because differences were observed only in this range.

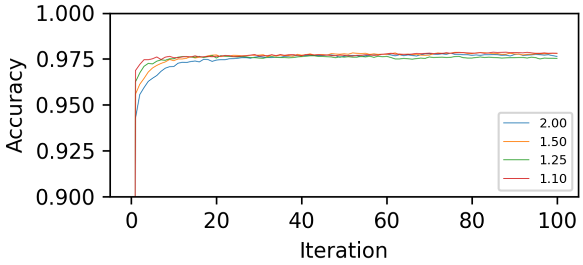


Figure 7: Accuracy versus ϵ

7 Additional experiments with the CIFAR-10 dataset

CIFAR-10 is a well-known image classification dataset with 60000 32 × 32 color images in 10 classes, including objects like airplanes, cars, and animals, shown as Fig. 8. Divided into 50000 training and 10000 test images, it is widely used to benchmark image recognition models due to the challenges posed by its diverse object orientations, lighting, and backgrounds.

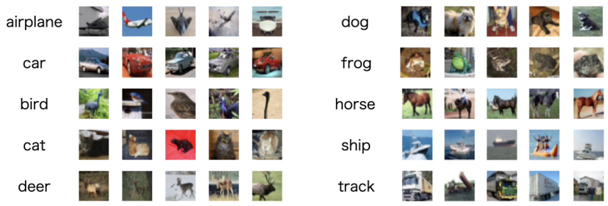


Figure 8: Examples of CIFAR-10 Dataset

DNNs with two hidden layers, each containing 64, 128, 256, and 512 neurons, were configured in both LesserDNN and TensorFlow for comparison. The accuracies achieved with the CIFAR-10 dataset were then compared between the two implementations, shown as Table3. In the TensorFlow model, features such as batch normalization and data augmentation were not used to ensure a direct comparison of the optimization algorithms.

8 Discussion

The experimental results showed that the SA-based learning algorithm can be applied to train LesserDNN, a DNN with quantized weights. Moreover, the fully connected

Table 1: Different network sizes for MNIST

	32	64	128	256
LesserDNN	0.9537	0.9675	0.9731	0.9776
TensorFlow/SGD	0.9618	0.9699	0.9734	0.9753
TensorFlow/Adam	0.9721	0.9789	0.9830	0.9848

Table 2: Different batch sizes for MNIST

	250	500	750	1000
LesserDNN	0.7832	0.8463	0.8766	0.8870
TensorFlow/SGD	0.6552	0.7689	0.8216	0.8347
TensorFlow/Adam	0.8082	0.8614	0.8850	0.8985

Table 3: Different network sizes for CIFAR-10

	64	128	256	512
LesserDNN	0.483200	0.4857	0.4706	0.4795
TensorFlow/SGD	0.4962	0.513	0.5231	0.5319
TensorFlow/Adam	0.5019	0.518	0.5333	0.5504

model achieved an accuracy comparable to that of existing DNNs on a classification task with the MNIST dataset.

The MNIST experimental results showed that LesserDNN has almost equivalent performance to TensorFlow, and we confirmed a decrease in accuracy of less than 1%. Furthermore, we confirmed that the accuracy decreased because the network size and batch size were reduced; however, even in such cases, considerable accuracy was maintained, indicating that a tradeoff between computational complexity and accuracy can be established.

When the batch size and number of iterations were fixed, the accuracy decreased slightly depending on the size of the network, as shown in Table 1. TensorFlow/SDG and TensorFlow/Adam both showed similar results. Thus, LesserDNN exhibits similar characteristics to DNNs.

Fig. 5 shows that no overfitting occurred in the experiments. After a certain number of iterations, the accuracy changed only slightly, although the loss function kept decreasing. The algorithm appears to switch between semi-optimal solutions of weights. The iterations should be terminated when the accuracy no longer increases. It is important that the method for determining the appropriate batch size be clear since the number of iterations is the hyperparameter of the LesserDNN model.

In Fig. 6, as the batch size varied from 60000 to 1000, the accuracy of LesserDNN decreased gradually from 97% to 88%. Then, when the batch size was 500, the accuracy decreased sharply to 84% and decreased further to 78% when the batch size was 250. The accuracy of TensorFlow decreased in the same trend with the both SDG and Adam. LesserDNN showed considerably equimbarent performance with TensorFlow.

ϵ moderates the learning rate. This parameter controls the numbers of neurons that are modified during each iteration in the cooling loop and is a major factor in determining the learning rate. A higher learning rate narrows the search direction, while a lower learning rate allows the search to proceed without narrowing the range of possibilities. Fig. 7 shows that the maximum accuracy does not change significantly as ϵ varies, although the accuracy in the 1st iteration of the main loop is higher when ϵ is smaller.

The additional evaluation on the CIFAR-10 dataset showed that the accuracy was lower than that of TensorFlow, but the results were still comparable.. The difference is minimal, especially when the network size is small. Since LesserDNN transforms learning into a weight combination problem, having fewer neurons makes the search easier and allows for more efficient use of computational resources.

The process of training LesserDNN iterates between inference and evaluation; thus, training is possible even in systems with limited resources, such as embedded systems and IoT devices. Additionally, training can be distributed among GPUs in the same system or among different systems over IP networks. Therefore, various systems can be constructed. For example, in one system, only inferences may be performed on edge devices, while training can be performed on GPUs in a cloud service.

9 Future work

We conducted performance comparisons with TensorFlow, for algorithm validation, and as a result, we did not precisely measure the computation times. However, there was a substantial difference, even in relatively simple problems like MNIST. This issue is expected to become more significant when dealing with more complex problems in the future. Therefore, it will be necessary to explore methods for efficient batch switching during training, similar to SGD.

In future work, we intend to address how to apply convolutional neural networks (CNNs) to LesserDNN as well as attempt to apply LesserDNN to more difficult problems, such as CIFAR-100 and Tiny ImageNet. In this paper, we assure that the small fully connected networks in LesserDNN are sufficient for small problems such as MNIST. However, CNNs are important for applying LesserDNN to practical problems.

Another issue we intend to address is accelerating the hardware with the algorithm. Multiplications of the weight and input values of the neurons can be replaced by SHIFT operations since quantized weights are representable by a factor of 2. This is advantageous for creating efficient logics in FPGA and converting networks to ASICs.

One important concept of LesserDNN is that the training process solves a combinatorial problem; thus, LesserDNN should have a high affinity with emerging quantum computers. Quantum computers are known to be great for solving combinatorial optimization problems. Therefore, we will attempt to apply our algorithm in quantum computers in future work.

Acknowledgement

To Tadasuke Furuya Ph.D. of the Faculty of Marine Technology at Tokyo University of Marine Science and Technology, and Takehiko Kashiwagi from Parallel Networks LLC, we would like to express our sincere gratitude for the support.

References

[1] Vincent Vanhoucke, Andrew Senior and Mark Z. Mao. Deep Learning and Unsupervised Feature Learning Workshop, NIPS 2011.

[2] Jingyong Cai, Masashi Takemoto, and Hironori Nakajo. A Deep Look into Logarithmic Quantization of Model Parameters in Neural Networks. In Proceedings of the 10th International Conference on Advances in Information Technology (IAIT '18). Association for Computing Machinery, Article 6, 1–8, 2018. https://doi.org/10.1145/3291280.3291800

[3] Seyyed Mohammad Mousavi, Elham S. Mostafavi, Pengcheng Jiao. Next generation prediction model for daily solar radiation on horizontal surface using a hybrid neural network and simulated annealing method, Energy Conversion and Management, Volume 153, Pages 671-682, 2017. https://doi.org/10.1016/j.enconman.2017.09.040

[4] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. BinaryConnect: training deep neural networks with binary weights during propagations. In Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'15), Vol. 2, 3123–3131, 2015.

[5] B. Liu, F. Li, X. Wang, B. Zhang and J. Yan. Ternary Weight Networks. ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1-5, 2023 https://doi.org/10.1109/ICASSP49357.2023.10094626

[6] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, arXiv, 2017, https://doi.org/10.48550/arXiv.1412.6980

[7] Open MPI https://www.open-mpi.org/

[8] NVIDIA Corporation https://www.nvidia.com/

[9] OpenCL https://www.khronos.org/opencl/

[10] CUDA Toolkit https://developer.nvidia.com/cuda-toolkit/

Efficient Line-Based Visual Marker System Design with Occlusion Resilience

Abdallah Bengueddoudj[1*], Foudil Belhadj[1], Yongtao Hu[2], Brahim Zitouni[3], Yacine Idir[1], Ibtissem Adoui[1], Messaoud Mostefai[1]
[1]LMSE Laboratory, Mohamed El-Bachir El-Ibrahimi University of Bordj Bou Arreridj, Algeria
[2] Guangdong Virtual Reality Technology Co., Ltd., Guangzhou, China
[3]Private Optical Institute of Bordj Bou Arreridj, Algeria
E-mail : abdallah.bengueddoudj@univ-bba.dz
*Corresponding author

Keywords: visual markers, homography, perspective distortion, occlusion, pose estimation

Received: October 2, 2024

Today, the most widely used visual markers, such as ArUco and AprilTag, rely on square pixel arrays. While these markers can deliver satisfactory detection and identification outcomes, they remain vulnerable to corner occlusion despite incorporating corrective codes. Conversely, line-based markers offer increased resilience against occlusions but are typically constrained in terms of codification capacities. The markers developed in this research leverage linear information to propose a pyramidal line-based structure that exhibits robustness to corner occlusion while providing enhanced coding capacities. Moreover, the projective invariance of the constituent lines enables the validation of a homography-less identification method that considerably reduces computation resources and processing time. We assembled an extensive test dataset of 169,713 images for evaluation, including rotation, distances, and different levels of occlusion. Experiments on this dataset show that the proposed marker significantly outperforms previous fiducial marker systems across multiple metrics, including execution time and detection performance under occlusion. It effectively identifies markers with up to 50% occlusion and achieves identification at a resolution of 1920×1080 in 17.20 ms. The developed marker generation and identification, as well as an extensive marker Database, are publicly available for tests at https://github.com/OILUproject/OILUtag

Povzetek: Predstavljen je izboljšan sistem vizualnih oznak, ki temelji na linijah in je odporen na delne zakritosti. Predlagana je piramidalna strukturaoznak, ki omogoča večjo kodirno zmogljivost in robustnost proti zakritju vogalov. Razvili so metodo identifikacije brez uporabe homografije, kar zmanjšuje računske zahteve in čas obdelave.

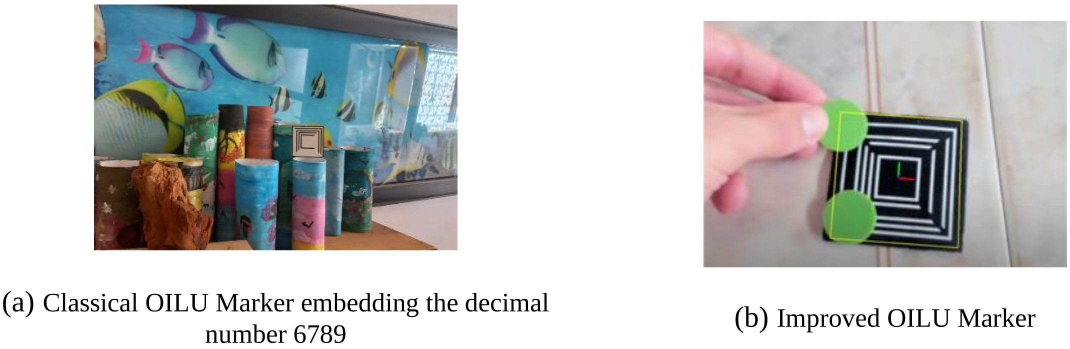
1 Introduction

Visual markers are artificial graphical codes representing numerical (or textual message) information that can be associated with objects to be uniquely identified. Computer vision applications use these tags to simplify the automatic perception of objects inside a scene and make their localization more precise. These are widely used in product labeling and tracking, robotics localization and mapping [1], camera calibration and pose estimation [2-3], augmented reality applications [4], automatic navigation [5] and medical positioning [6].

Today, the most prevalent visual markers, such as April [7] and ArUco [8] Tags, utilize square pixel arrays. Although these markers often yield satisfactory detection and identification results, they remain susceptible to external corners occlusion despite the inclusion of error correction codes. In contrast, line-based markers offer greater resilience against occlusions but are often constrained in coding capacities [9]. Recently, Chahir et al. [10] introduced a novel line-based marker called the OILU marker, addressing codification limitations. This marker utilizes groups of pyramidal-shaped lines to create highly distinguishable 2D markers (Figure 1.a).

While offering significant advantages in coding capabilities, the developed identification method, which relies on a time-consuming level set technique [11], slows down processing, particularly in scenarios where multiple markers are in the camera's field of view. The reported average processing time is approximately 40 ms per marker, making this solution unsuitable for constraining real-time applications. In addition, the proposed scheme (marker design and identification method), does not solve the problem of external corners occlusion, for which most square markers remain ineffective. In fact, if just one corner of these markers is occulted, the detection fails.

To overcome these challenges, we propose in this paper, a less computational identification method, based on cumulative histogram analysis that allows reducing processing times by almost half compared with the work of Chahir et al. [10]. However, as the method relies only on the external marker's corners for localization, it also remains vulnerable to external corners occlusion. Moreover, as the identification scheme integrates


(a) Classical OILU Marker embedding the decimal number 6789

(b) Improved OILU Marker

Figure 1: Classical OILU marker embedding the decimal number 6789 and improved OILU Marker identification and pose estimation under corners occlusion.

homography transform in its processing, computation performances decrease as the number of markers within the camera's field of view increases.

To remedy these limitations, a revised marker design (Figure 1.b), as well as a dedicated homography-less identification approach, are proposed. The adopted identification scheme exploits the marker's local properties to switch from a line-based representation to a more accurate and relatively fast dot-based one.

Deep tests on real and synthetic images highlight the performance and robustness of the proposed solution against challenging conditions, with a particular focus on corners occlusion. Despite this, the homography transform remains applicable for pose estimation, where improved markers demonstrate superior performances compared to state-of-the-art markers

In summary, the main contributions of this paper are:

- The layout design of the OILU Tag has been enhanced to offer more robustness to occlusion and overlapping objects.
- The proposal introduces a low computational homography-less identification method. The average execution time has been considerably reduced for both desktop and mobile architectures, making it suitable for constraining real-time applications.
- A dedicated OILU Tag Generator as well as a huge database are made available for comparative tests with the well-known state-of-the-art visual Tags.

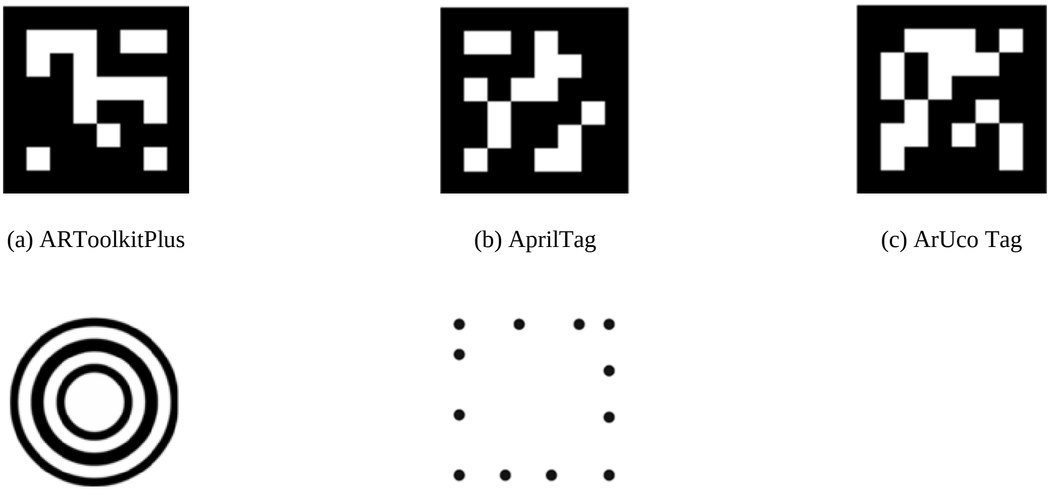
The remainder of this paper is organized as follows: Section 2 provides a quick literature review on well-known fiducial markers. Section 3 briefly presents the OILU code basics and highlights its key strengths as being an efficient visual marker. Sections 4 describe our primary OILU marker identification scheme, followed in Section 5 by the presentation of a revised marker design, as well as its validated homography-less identification approach. In section 6, extensive tests are conducted on real and synthetic images. Finally, section 7 concludes the paper with interesting perspective views.

2 Related works

There are many conceptions of visual markers in the literature (Figure 2). These can be clustered into three main categories: square-based, line-based, and dot-based tags. The first category regroups all QR-like tags that encode binary information in black/white cells arranged in square grid layouts. ARToolKit [12] is the oldest fiducial marker proposed for AR applications. It consists of a black-bordered square inside, which is embedded in a known image as a payload. Its limitation resides in the matching method that uses image correlation techniques to detect the embedded pattern. ARToolKitPlus and ARTag [13-14] are improved versions released to overcome these limitations. They use binary-coded patterns to encode the embedded identifier. Furthermore, the ARTag introduces additional information as an error-correction payload. Based on ARTag's idea, many efficient square markers were proposed, among them April Tag [7] and ArUco Tag [8], which became ubiquitous in the AR field. Both allow generating of user-customized dictionaries using some heuristics to maximize some criterion such as inter-marker distance and the number of bit-transitions.

Recently, a new square-like TopoTag was introduced by Yu et al. [15]. It offers a highly customizable marker shape. The fundamental structure of the marker consists of a black frame with black squares positioned on a white background. One notable advantage of TopoTag is its variable dictionary size. The authors claim that generating the dictionary is significantly faster compared to similar marker systems like April and ArUco Tags. Based on AprilTag, [16] proposed ChromaTag by using different colors to represent the internal bits to make the detection easier and speed up its decoding. Line-based markers apply some measurements on the basic forms like line-thicknesses and angle sizes to encode the elementary information. Usually, markers in this approach are robust against bad acquiring conditions such as blurring and variation in lighting. They perform well in occlusion situations. Based on the classical linear bar code, Calvet et al. [17] proposed a circular version called CCTag, in which the lines have been substituted by circles with different thicknesses.

Dot-based tags [18-19] enable the developing of projective invariants fiducial marker systems based on cross ratios computation. Even though these markers exhibit higher accuracy in camera calibration and pose



(a) ARToolkitPlus

(b) AprilTag

(c) ArUco Tag

(d) CCTag

(e) Pi-Tag

Figure 2: Examples of well-known visual markers.

Table1: Main markers specifications

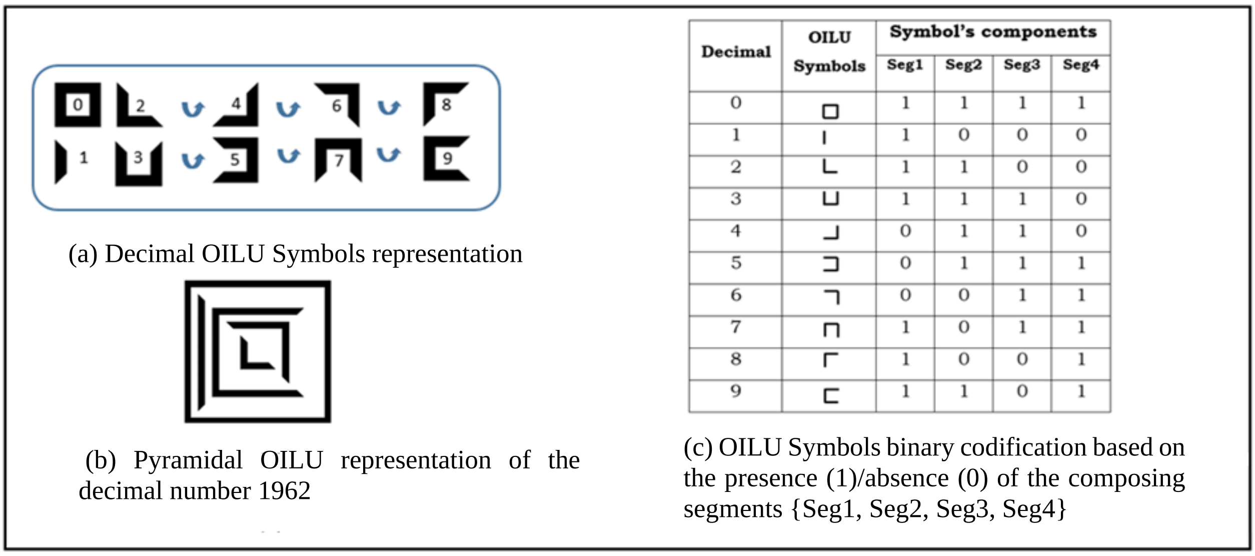
Marker	Family	Shape	Scalability	Key Findings	Flexibility
AprilTag [7]	Qr-like	Square	Limited	High dictionary size (5329); widely used in robotics and AR systems	Low occlusion resilience; limited scalability; no flexibility
ArUco [8]	Qr-like	Square	Limited	Moderate dictionary size (250); simple implementation	Low occlusion resilience; limited scalability; no flexibility
ARToolKit [12]	Qr-like	square	Limited	Developed the first generation of AR marker tracking tools	lacks flexibility and high computational complexity
CCTag [17]	Bar-like	Circular	Limited	Easy detection of circular shapes	Very limited dictionary size (39); low occlusion resilience; no flexibility
Pi-tag [18]	Dot-like	Square	Limited	Balanced dictionary size (300); efficient marker detection	Low occlusion resilience; limited scalability; no flexibility

estimation, they offer a limited number of distinctively recognizable patterns [20].

OILU Tag [10] is a distinct type of fiducial marker based on the two initial categories. It distinguishes itself from other fiducial markers in two main aspects: firstly, humans and machines can read it. Secondly, it exclusively employs lines as primary patterns to encode the elementary information. Moreover, the OILU marker introduces a novel approach to layout design, ensuring user accessibility and ease of implementation. It can even be accurately reproduced by hand drawing, eliminating OILU Tag [10] is a distinct type of fiducial marker, based on the need for specialized tools or intricate encoding schemes, making it ideal for a wide range of applications. Table 1 presents well-known markers along with their features, such as shape and dictionary size.

In summary, existing visual markers have significant limitations. Square-based markers, such as AprilTag and ArUco, are particularly susceptible to corner occlusion, resulting in poor performance even with minimal obstruction. While line-based markers are more resilient to occlusion, they are hindered by limited coding capacity and high computational requirements, especially when dealing with numerous markers in the field of view. Moreover, these markers often rely on complex homography transformations, reducing their efficiency.

The new marker design overcomes these challenges with several key improvements. By utilizing lines as the primary pattern, OILU markers offer inherent redundancy, making them more resistant to occlusion and blurring compared to dot- or square-based markers. Additionally,



(a) Decimal OILU Symbols representation

(b) Pyramidal OILU representation of the decimal number 1962

Decimal	OILU Symbols	Symbol's components			
		Seg1	Seg2	Seg3	Seg4
0	□	1	1	1	1
1	I	1	0	0	0
2	L	1	1	0	0
3	⊔	1	1	1	0
4	┘	0	1	1	0
5	⊐	0	1	1	1
6	┐	0	0	1	1
7	⊓	1	0	1	1
8	┌	1	0	0	1
9	⊏	1	1	0	1

(c) OILU Symbols binary codification based on the presence (1)/absence (0) of the composing segments {Seg1, Seg2, Seg3, Seg4}

Figure 3: OILU Symbols representation. The whole symbols are incorporated in a square to delimit their area in a real-world scene

lines serve as effective separators between contrasting regions, improving marker detection and recognition.

OILU's unique pyramidal line-based structure enhances occlusion resilience, ensuring reliable detection even in difficult conditions. The homography-less identification method reduces computational demands, making it more suitable for real-time applications. The shift from line-based to dot-based representations also boosts detection accuracy and processing speed. These innovations position the OILU marker as a superior alternative to existing visual markers, striking a balance between robustness, efficiency, and practicality.

3 OILU markers basics

OILU markers, as described in [10], are based on a set of four basic symbols {O, I, L, and U}, corresponding respectively to digits zero, one, two, and three (Figure 3 (a). The remaining decimal symbols, related to digits {4, 5, 6, 7, 8, 9}, are obtained by successive counter-clockwise rotations of the two symbols L and U. The important feature of these symbols is their ability to be concatenated in a pyramidal fashion, producing multi-faceted numbers that can be exploited as visual markers (Figure 3 (b)). Each OILU symbol is coded in bi-nary according to Figure 3 (c). In the following, we will detail our improved identification approach based on cumulative histograms analysis. Compared to the level set method presented in [10], the adopted approach is relatively simple and computationally efficient. It operates on classical OILU markers and incorporates homography in its processing [21, 22].

4 Standard OILU markers detection and identification

To facilitate detection, the visual OILU markers are designed with black-outlined segments on a white background (or inversely). The identification process follows the standard computer vision processes, which involves three key stages: pre-processing, code detection, and decoding. The complete process presented in Figure 4 is as follows:

4.1 Preprocessing

The primary objective of the pre-processing stage is to enhance the quality of the captured images, ensuring they are optimized for subsequent stages. To achieve this, classical image processing filters can be applied [23], though modern cameras often capture high-resolution images. In real-time applications, there is a necessary trade-off between speed and accuracy. Downsampling the captured images enables quick noise filtering and reduces the execution time, especially in the subsequent stages.

The output of this stage is an improved grayscale image (as shown in Figure 4 (b)). Its goal is to localize all possible quadrilaterals eligible to be square-OILU markers in the grayscale image. The process comprises three main steps:

4.2 Eligible markers detection

4.2.1 Image Thresholding

The first step after obtaining the enhanced grayscale image is to binarize it, separating the objects in the image from the background. This makes the extraction of contours possible in the subsequent step. Several methods can be used for binarization [23]. The simplest method is direct thresholding, where a global threshold is applied; however, this method performs poorly on images with

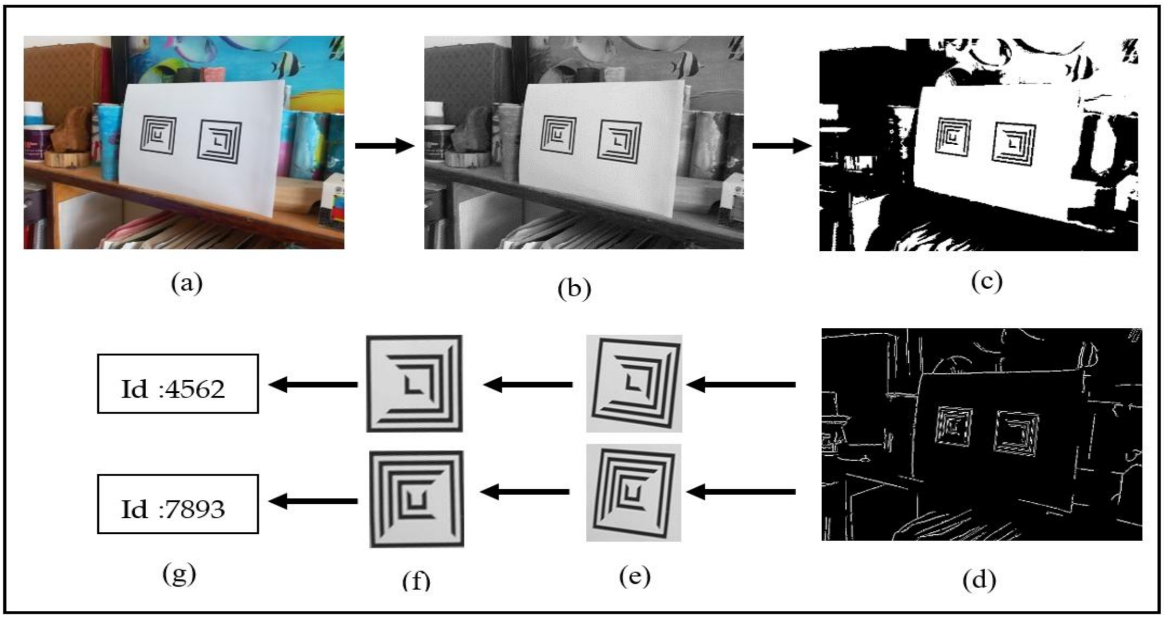


Figure 4: OILU Marker Detection process. (a) Input image acquiring. (b) input image processed and grayscale converted. (c) Binarized image. (d) Contours extracted. (e) Eligible markers extracted. (f) Perspective correction using homography. (g) Markers decoded.

multimodal histograms. The Canny method can be used, but it is time-consuming for real-time applications. For better performance, we utilize a local adaptive thresholding method, which is robust to varying lighting conditions and does not depend on a global threshold choice. Figure 4 (c) depicts the resulting binarized image.

4.2.2 Contour extraction

Given the square shape of the OILU Tag, we search for all potential quadrilateral shapes in the binarized image that could correspond to an OILU marker. To accomplish this, we first extract the contours of the image by tracing the transitions between black and white pixels, as described in [24]. Next, we approximate the obtained contours to the nearest polygonal shape using the Douglas-Peucker algorithm [25] (as shown in Figure 4 (d)). Only convex shapes with four corners are retained (Figure 4(e)). Some refinement steps are necessary to eliminate contours that are too small, too large, or too close to Each other [26].

4.2.3 Candidate markers determination and perspective adjustment

Although we retain all convex quadrilaterals with four corners in the previous step, not all of them are regular squares. Some may be subject to 2D transformation constraints such as rotations or perspective distortions. To correct these irregularities, a homography is applied to the sub-image framed by the quadrilateral. Once corrected, each obtained sub-image is resampled to a canonical grayscale image of size $Wc \times Wc$ using linear interpolation. The output of this step is a list of candidate square-shaped marker images (as depicted in Figure 4 (f)).

4.3 Marker Identification

Each candidate marker in the obtained list needs to be processed to confirm its content as an OILU marker and read its embedded identifier. As previously mentioned, each identifier digit is encoded in a separate layer using four segments that reflect the OILU symbolic. The challenging part is to identify the position of each layer, locate each segment within it, and extract its binary content, particularly in critical situations such as occlusions and noise. More formally, let K be an integer with N decimal digits, and M its corresponding OILU code. The segment-based binary codification of M is:

$$M = \left\{\left(s_0^i, s_1^i, s_2^i, s_3^i, s_4^i\right)\right\}_{i=1}^{N}, s_j^i \in \{0,1\}\ for\ j = 1..4 \quad (1)$$

The size of the embedded identifier (N), which corresponds to the number of layers, is unknown beforehand. Furthermore, no assumptions are made regarding the thickness of the segments, whether they are equal or not. When the segments are of equal width, the binary square image can be divided into a matrix of the same width and height as the segment width to isolate the segments easily. However, designing an OILU marker with different segment thicknesses and inter-layer space widths makes it more flexible and robust to a wide range of distortions, occlusions, and noise. In the subsequent paragraphs, we will consider this last case, which is more challenging. The decoding procedure, illustrated in Figure 5, involves several steps:

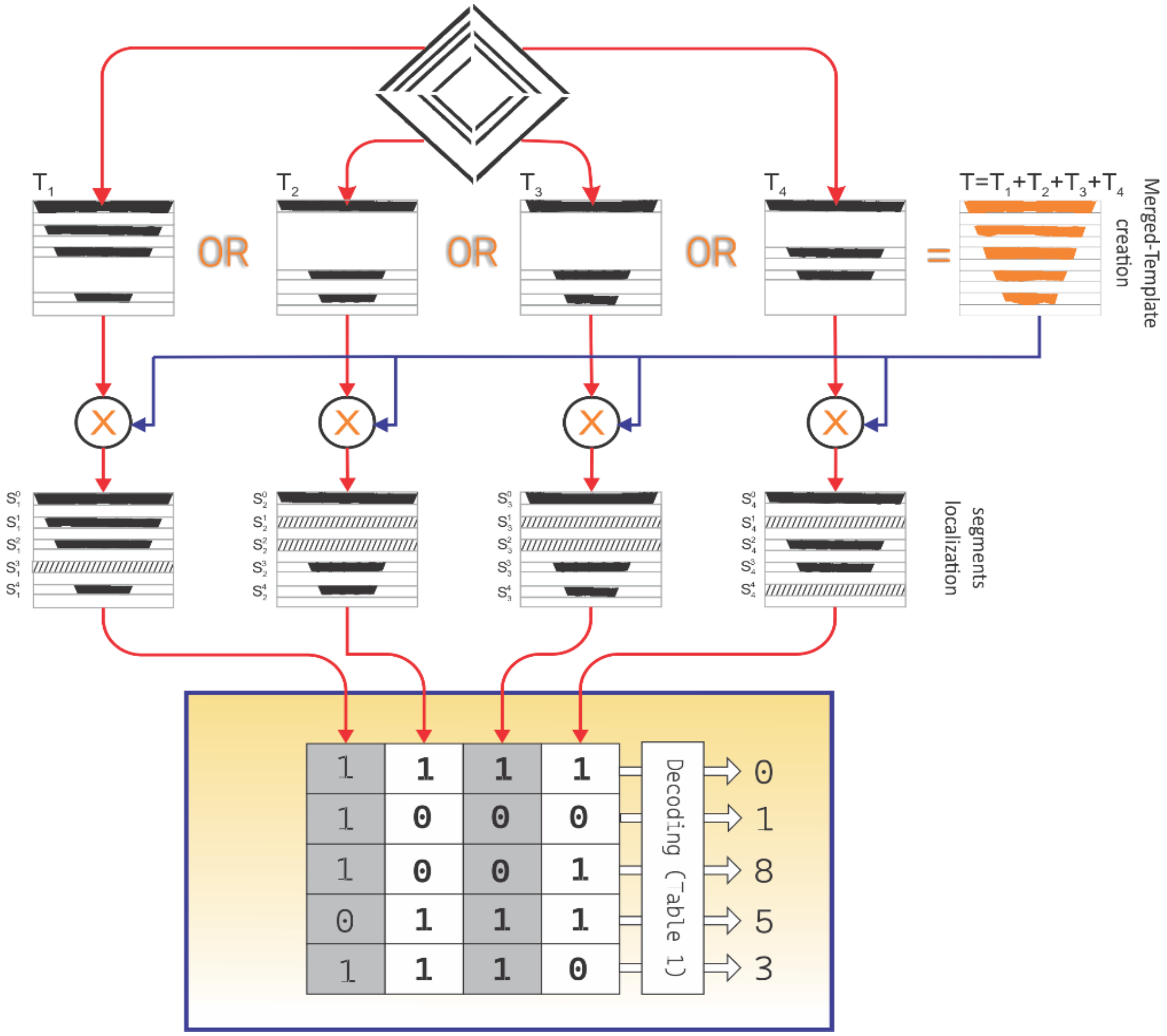


Figure 5: Decoding process: Each layer in a given OILU marker contains at least one black segment. Performing a bitwise-or operator on the four triangles (sectors) constituting the OILU -marker results in a merged template that contains a black segment on each layer. The merged template allows delimiting each black/white segment in each individual triangle.

4.3.1 Binarization

OILU tags exhibit bimodal histogram characteristics, making Otsu's thresholding method [27] the ideal choice for generating binarized images. This method determines the optimal threshold value to distinguish between the predominantly white background and the typically black OILU segments. In the resulting binarized image (Figure 4(c)), pixels within the segments (i.e., the region of interest or ROI) are assigned a value of ‘1’ while all other pixels are assigned a value of ‘0’.

4.3.2 Layers extraction and black segments localization

The binary image is first divided into four triangles or sectors, denoted as T1, T2, T3, and T4. Each triangle Ti comprises a set of alternating black and white bands that contain the encoded segments. A black band indicates a binary one '1', while a white band could represent zero '0' or multiple consecutive zeros '0' (as illustrated in Figure 5). To locate the segments within the image, we utilize a useful property of the OILU marker that states ‘each layer contains at least one black segment’. Therefore, combining all the triangles by performing a bitwise-OR operation between their contents yields a template triangle T (2) containing the exact number of black segments equal to the number of digits N in the encoded identifier (as depicted in Figure 5).

$$T = T1 + T2 + T3 + T4 \qquad (2)$$

The merged triangle T plays the role of a template guide that allows delimiting all the black/white segments in each layer by analyzing its horizontal and vertical cumulative

Table 2: OILU decoding matrix. Each column corresponds to one triangle in which each segment is coded in binary (s_i^j=1 means the segment is detected as being black)

T1	T2	T3	T4	Lookup Figure 3 (c)
s_0^1	s_0^2	s_0^3	s_0^4	digit$_0$ (must equal 1111)
s_1^1	s_1^2	s_1^3	s_1^4	digit$_1$
⋮	⋮	⋮	⋮	⋮
s_{N-1}^1	s_{N-1}^2	s_{N-1}^3	s_{N-1}^4	digit $_{N-1}$
s_N^1	s_N^2	s_N^3	s_N^4	digit $_N$

histograms (respectively HCH and VCH). The horizontal histogram HCH is the sum projection of pixel values along all rows inside the triangle T.

$$HCH_j = \sum_{k=1}^{w_c} \mathrm{T}(j,k)\,; j = 1..W_c/2 \tag{3}$$

It allows localizing the black segments by following the black-white transitions. Indeed, black segments coincide with high ridges (peaks) in the HCH, while white ones constitute low valleys. To handle occlusion situations and to be robust against noise, a percentage threshold $\omega_1 = 2/3$ regarding the whole line is set up to decide whether a horizontal-histogram value is black or white.

$$HCH_j \geq (\mathrm{w}_c - 2 * j) * \omega_1 \Rightarrow T(j) \equiv black\ line \tag{4}$$

It is worth noting to mention that ω_1 is dependent on the row position; outer rows correspond to high values of ω_1 and vice-versa. The VCH is the vertical projection of T over all columns; it allows the detection of the number of black bands, confirming the horizontal histogram analysis results.

$$VCH_j = \sum_{k=1}^{w_c/2} \mathrm{T}(k,\mathrm{j})\,; j = 1..w_c \tag{5}$$

The clustering of many adjacent black (respectively white) rows in the HCH constitutes a black (respectively white) segment, provided that the number of rows exceeds a threshold $\omega_2 = 25\%$. After creating the merged template and locating its segments, we utilize it to determine the position of each segment within the four triangles.

4.3.3 Marker validation

To confirm that the embedded data is an OILU code, we only need to verify that the marker satisfies the following criteria, which serve as an OILU signature:

- The strict alternation of bands: the merged triangle T comprises alternating black-white segments. The most outer (starting) band is black, and the most inner (ending) is white. The number of all bands is always even.
- Each black band in a triangle must correspond to a black band in the merged triangle.
- Each segment must be connected (no small fragments).

4.3.4 Marker decoding

To decode the content of the validated marker, we follow the reverse process of the encoding procedure (as shown in Figure 5, decoding step), which involves the following steps:
We affect the value “1” for each black segment and “0” for each white one, starting from the most outer segment to the most inner. Each triangle ($\mathrm{T_i}$) i=1..4 is composed of N+1 segments:

$$\mathrm{T_i} = \{s_i^k\}_{k=0}^{N}, s_i^k \in \{0,1\}\ for\ i = 1..4 \tag{6}$$

Next, we concatenate the binary values inside each triangle Ti to form a binary string:

$$str_i = s_0^i s_1^i .. s_{N-1}^i s_N^i, for\ i = 1..4 \tag{7}$$

After that, we arrange the four binary strings vertically to form a decoding matrix (Table 2) starting with the left triangle and going counterclockwise (left, bottom, right, then upper). Each line of the decoding matrix
represents a digit in the identifier whose decimal value can be obtained from the OILU codification table (Figure 3-c). These aforementioned steps are repeated for all eligible markers, and only the validated markers that have their IDs and Cartesian coordinates within the original image are retained after the detection process.

4.4 Processing time required for standard OILU markers identification

The identification scheme described in section 4 has been implemented and tested on a typical Laptop equipped with a 2.4 GHz Intel Core i7 processor with 16 GB RAM, running Windows 10. The processing time can be divided into three main steps: (1) finding marker candidates (including image processing, contours extraction and eligible squares determination), (2) perspective correction of all candidates, and (3) markers validation. The execution of the first step can be affected by the size S of

Table 3: Average processing time.

Architecture	Step	Proposed method		(Chahir et al., 2021) method
		Average time per step Image (640x480)	Total time /candidate	Total time /candidate
Typical laptop	1	17.33 ms	19 ms	40 ms
	2	1.24 ms		
	3	0.43 ms		
Typical Android smartphone	1	22.08 ms	25.06 ms	Not reported
	2	1.87 ms		
	3	1.10 ms		
Step 1 : Finding Marker Candidate – Step 2 : Perspective corrections – Step 3 : Marker validation				

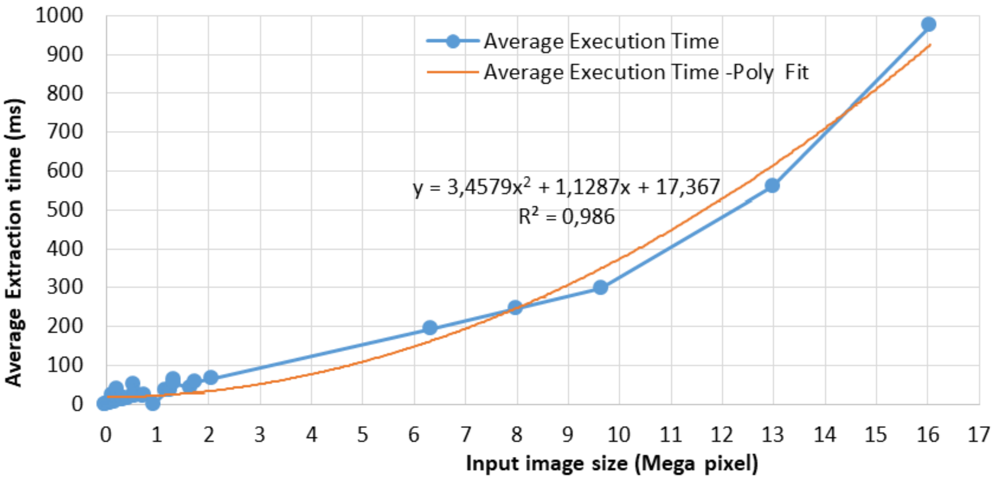


Figure 6: Evolution of the average detection time of the OILU code in function of the input-image size with its polynomial distribution fit.

the input image and the complexity of its texture in terms of contained contours, while the processing of the second and the third steps only depends on the marker canonical size $w_c \times w_c$. Table 3 resumes the average execution time of each step taken for multiple input images of size $S = 640 \times 480$ and canonical size of 256×256 pixels.

Figure 6 shows the evolution of the average extraction time in function of the image size and its best distribution fit which seems to be a polynomial distribution (with R2 goodness-of-fit = 0.986). As observed, the detection time increases significantly, reaching up to one second, when the input image has a high resolution.

While the adopted scheme reduces processing times by nearly half compared to the approach by Chahir et al. [10], it still depends on the external corners of the markers for localization, rendering it susceptible to occlusion. Additionally, the integration of homography transformations in the identification process leads to reduced computational performance as image resolution increases or as more markers appear within the camera's field of view.

In the following, an improved OILU marker system design is proposed (Figure 7). It involves enclosing the embedded identifier within two nested square-like quadrilaterals, enabling efficient marker detection even when the external marker's corners are occluded. The developed identification method considers OILU numbers as groups of locally parallel segments, treating them separately without the need for a homography transform, thereby reducing computation resources and minimizing processing time.

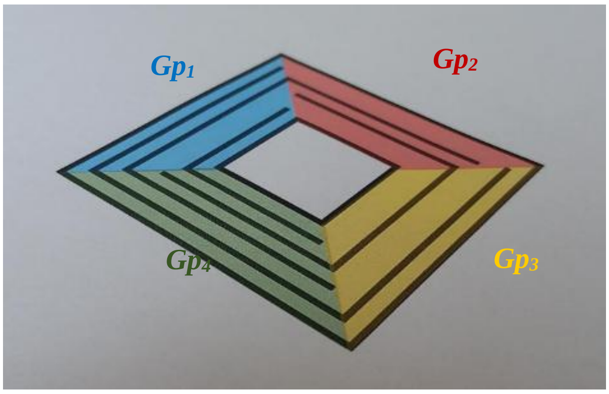
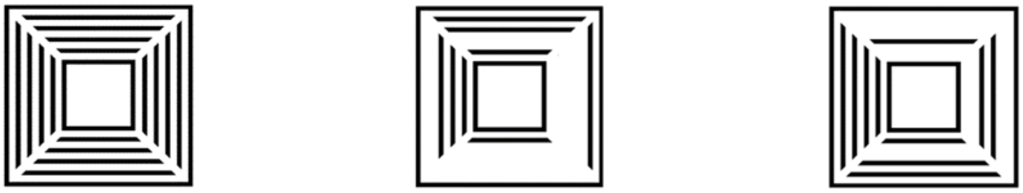


Figure 7: Groups of locally parallel segments.

5 Improved OILU markers system design

Common, well-known problems with state-of-the-art markers include detection failures when their corners are occluded, and lack of size adaptation to the camera's field of view (FOV), especially when the camera is in motion. This is evident, for example, when an autonomous drone attempts a landing based on its on-board camera. These issues have been addressed in various works [2-28-29]. An interesting approach presented in [2] involves designing fractal markers composed of imbricated quadrilaterals. In addition, being multi-scale markers, the latter are robust to partial occlusions. This inherent structure is characteristic of OILU markers, which are made up of nested square symbols, allowing their structure to be customized to overcome the above-mentioned problems. Hence, the adopted structure (Figure 8) is as follows:

- Two imbricated inner/outer square like – quadrilaterals as marker delimiters.
- A group of disconnected segments to embed the marker identifier



(a) $Id_1 = 0000$ (b) $Id_2 = 6819$ (c) $Id_3 = 2372$

Figure 8: Example of OILU markers with fixed inner and outer squares embedding different identifiers. Embedded symbols are drawn with disconnected segments (without corners) to distinguish them from the inner/outer quadrilaterals.

This arrangement simplifies marker detection, even in complex backgrounds, by focusing on identifying nested, similar square-like quadrilaterals within a filmed scene. This specific pattern streamlines OILU marker detection while filtering out non-OILU quadrilaterals. Furthermore, the nested structure enhances the markers' resilience to partial occlusion, which typically affects the outer quadrilateral. Since these imbricated quadrilaterals share a similar structure, any partial occlusion of the outer quadrilateral can be approximated and reconstructed through uniform rescaling. This advantageous feature is extensively leveraged in the experimental section to improve marker detection and pose estimation, even under occlusion.

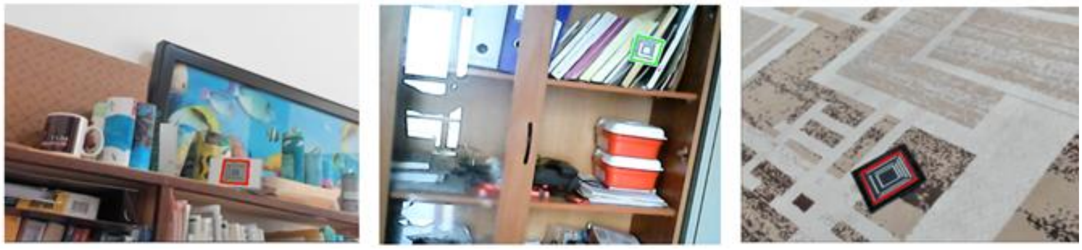
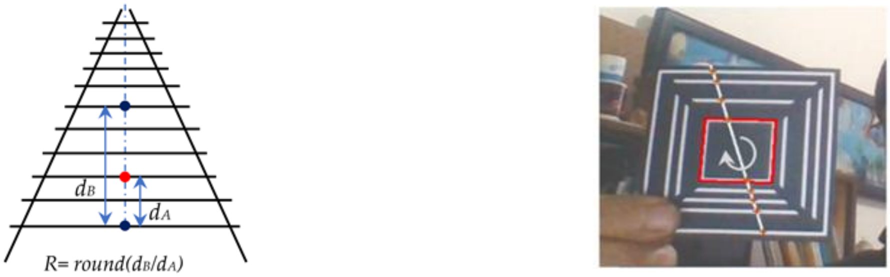


Figure 9: OILU markers detection within complex backgrounds.



(a) Computed railway interlines ratios

(b) Real OILU markers identification based on cross ratios computation

Figure 10: Analysis of Railway Interline Ratios and OILU Marker Identification

5.1 Candidate marker's location

As mentioned, OILU markers are composed of two imbricated square-like quadrilaterals.

Such composition eases markers' location tasks even within a complex background (Figure 9). For more selectivity, the fixed inner/outer quadrilateral's surface ratio is used to eliminate surrounding non-OILU quadrilaterals.

5.2 Marker identification

A deep evaluation of the perspective distortion levels within the adopted markers shows that these are more significant between the different groups of parallel segments (Gp1, Gp2, Gp3 and Gp4) than within the same group (Figure 7). Indeed, parallel lines in the same group remain locally parallel, even if the marker is acquired in perspective. In another way, each group of lines can be considered as railway ties (Figure 10 (a)) for which computed Euclidean length ratios remain invariant to perspective changes [30]. The challenge here is exploiting such ratios to generate the related embedded sub-codes without homography.

The adopted approach involves crossing the composing groups of lines (Gpi) with a virtual line centred on the middle of the marker (Figure 10 (b)). Crossed lines sections are then used to locate the corresponding cross points and generate associated sub-codes. The quantity of retained cross-point sets within each group can range from one to multiple, depending on the desired level of resilience against distortions, particularly occlusion. For example, a single set per group is adequate for marker identification when markers are fully visible. However, in scenarios where parts of the marker are obscured, multiple sets from various regions are required to confirm the most common ones. Notably, identification may fail even with multiple selected regions in case of significant occlusion. Deep tests in the experimental section will show the accuracy and limits of this approach. Globally, the adopted OILU marker code generation process is as follows: First, for each group Gpi (i= 1 to 4), the following metrics are computed:

<u>Group's metrics computation</u>

- *the number (N) of cross points,*
- *the (N -1) inter-dots Euclidian distances {d_j, j=1 to N -1},*
- *the group band width $W_i = \sum_{j=1}^{N-1} d_j$,*
- *the average dots spacing $A_i = W_i/S$, with S the number of code symbols,*

the ratios R_j = round (d_j/A_i).

Cross points positions are estimated according to their computed ratios Rj and marker's format (number of embedded symbols). In case of a four symbols marker, the possible configurations to be tested are as follows:

<u>Cross points position estimation</u>

```
    R_j = round (d_j/A_i)  // with Ai, the average cross points spacing of Gp_i
    if R_j = 1 the corresponding points are adjacent     // case (a)
    else if R_j = 2 related points are separated by one empty space     // case (b)
    else if R_j = 3 related points are separated by two empty spaces     // case (c)
    else related points are separated by three empty spaces     // case (d)
    end
```

The basic example (Figure 11) illustrates where a group's number of cross points equals three (two inner/outer boundary points and one symbol cross point). The number of inter-cross points Euclidean distances is equal to two (d_1, d_2). Computed metrics are: $W_i = d_1 + d_2$; $Ai = W_i/4$; $R_1 = round\ (d_1/A_i)$. Therefore, the presented red symbol cross point will be in one of the four cases $\{(a), (b), (c)\ or\ (d)\}$, according to R_1 value, equal to 1, 2, 3 or 4.

A second illustrative example (Figure 12) shows three views of a real marker (embedding decimal number 0389). Developed identification method, calculates corresponding cross points coordinates and metrics for each group of segments. It is worth mentioning that since processing is carried out separately on each group of cross

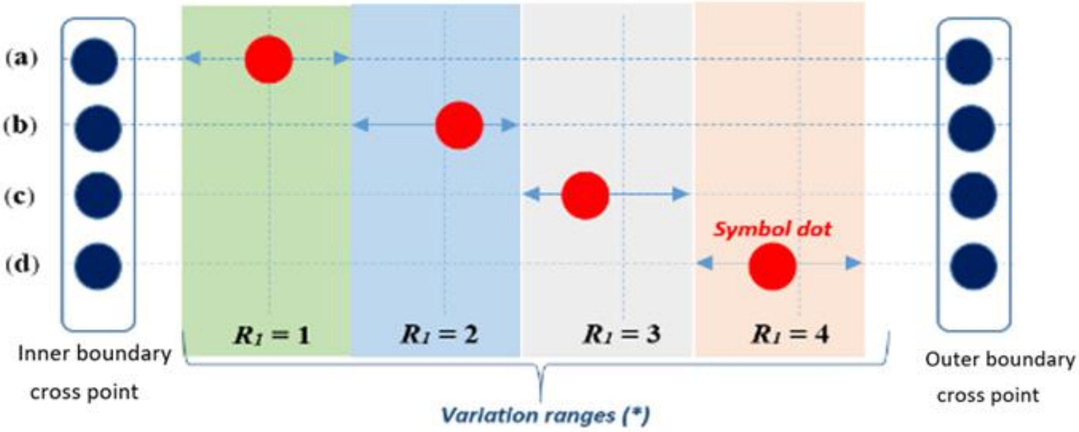
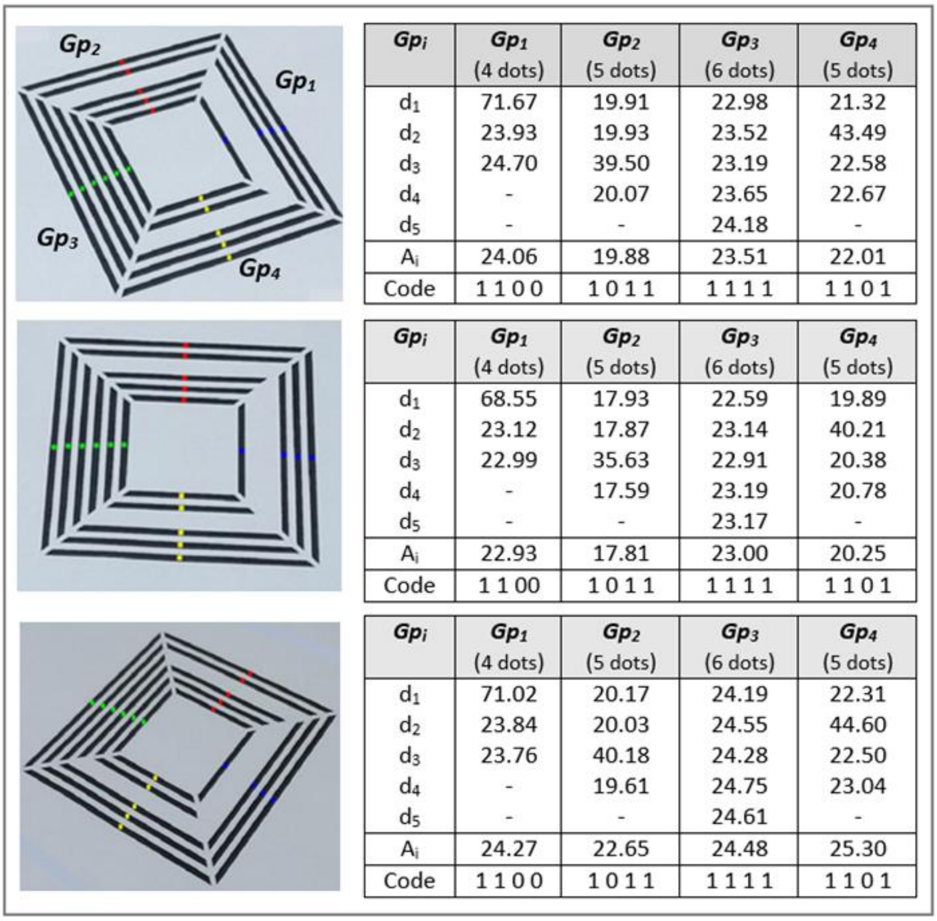


Figure 11: Cross points position estimation, (*) Colored bands delimit the cross-point variation intervals.

points, perspective distortions have no impact on the computed ratios and, consequently on the related embedded codes, making homography transformation unnecessary for marker identification. In the following, deep tests on real OILU markers are performed to evaluate the correctness and robustness of this approach against leading.


Gp_2
Gp_1
Gp_3
Gp_4

Gp_i	Gp_1 (4 dots)	Gp_2 (5 dots)	Gp_3 (6 dots)	Gp_4 (5 dots)
d_1	71.67	19.91	22.98	21.32
d_2	23.93	19.93	23.52	43.49
d_3	24.70	39.50	23.19	22.58
d_4	-	20.07	23.65	22.67
d_5	-	-	24.18	-
A_i	24.06	19.88	23.51	22.01
Code	1 1 0 0	1 0 1 1	1 1 1 1	1 1 0 1

Gp_i	Gp_1 (4 dots)	Gp_2 (5 dots)	Gp_3 (6 dots)	Gp_4 (5 dots)
d_1	68.55	17.93	22.59	19.89
d_2	23.12	17.87	23.14	40.21
d_3	22.99	35.63	22.91	20.38
d_4	-	17.59	23.19	20.78
d_5	-	-	23.17	-
A_i	22.93	17.81	23.00	20.25
Code	1 1 00	1 0 1 1	1 1 1 1	1 1 0 1

Gp_i	Gp_1 (4 dots)	Gp_2 (5 dots)	Gp_3 (6 dots)	Gp_4 (5 dots)
d_1	71.02	20.17	24.19	22.31
d_2	23.84	20.03	24.55	44.60
d_3	23.76	40.18	24.28	22.50
d_4	-	19.61	24.75	23.04
d_5	-	-	24.61	-
A_i	24.27	22.65	24.48	25.30
Code	1 1 0 0	1 0 1 1	1 1 1 1	1 1 0 1

Figure 12: Computed cross points metrics and related embedded codes.

6 Experiments

Tests are carried out on a huge database of synthetic and real markers, with nearly 5000 markers available in three groups of different sizes: small ($5\ cm \times 5\ cm$), medium ($10\ cm \times 10\ cm$), and large ($15\ cm \times 15\ cm$). Printed markers are placed on a rotating support using different types of cameras. Specifically, we employ a high-resolution Logitech camera (Figure 13). Deeper tests with many markers, displayed on a Surface Pro X tablet are also performed to assess the performance of our method under various distortion conditions (Figure 13 (b)). In our tests, we compared the performance of our developed marker with two well-known markers, ArUco and AprilTag. We gathered data for each tag family: 36h12 and 16h3 for ArUco and 25h9 and 36h10 for AprilTag.

Codes for the developed marker (generation/detection) and the OILU database (images and videos) are publicly available for download at the following link: *https://github.com/OILUproject/OILUtag*.

6.1 Marker to camera distance impact

Initially, we evaluate the impact of the marker-to-camera distance on the performances of marker detection. The camera was positioned in front of the marker at different distances d, ranging from 0.2m to 4 m. Obtained identification results are presented in Table 4. Compared with the ArUco and April Tags results, the proposed marker performs less well when using a fixed-focus camera (Logitech in our case). As distance increases, adjacent parallel lines expand, forming a uniform area that prevents accurate identification. This problem can be solved using an autofocus camera, such as a smartphone. Note that after a certain distance (superior than 3.5 m), marker identification became dependent on the camera resolution. The higher the resolution, the better the identification and vice versa.

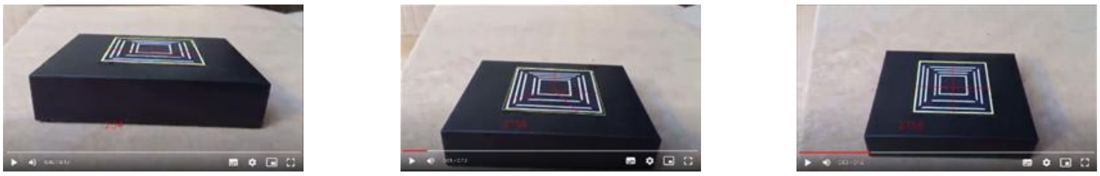
(a) A high-resolution front camera is used to record video sequences at various distances ranging from 0.2 to 4 meters

(b) A Surface Pro X tablet serves as a display platform to validate the identification scheme across a large database of markers

Figure 13: The experiment setup involves a rotating plate with embedded markers, which is controlled by a stepper motor to accurately capture the markers in a perspective view.

Table 4: Robustness to distance.

Cameras	Distance (m)	ArUco			April			OILU		
		T_1	T_2	T_3	T_1	T_2	T_3	T_1	T_2	T_3
Logitech Camera	2			×			×		×	×
	3					×				
	4	×	×		×			×		
Marker dimension : T_1 (5cm×5cm) – T_2(10cm×10cm) – T_3 (15cm×15cm)										



(a) Failing identification case (β =10)

(b) Successful marker identification (β = 20)

(c) Successful marker identification (β =40)

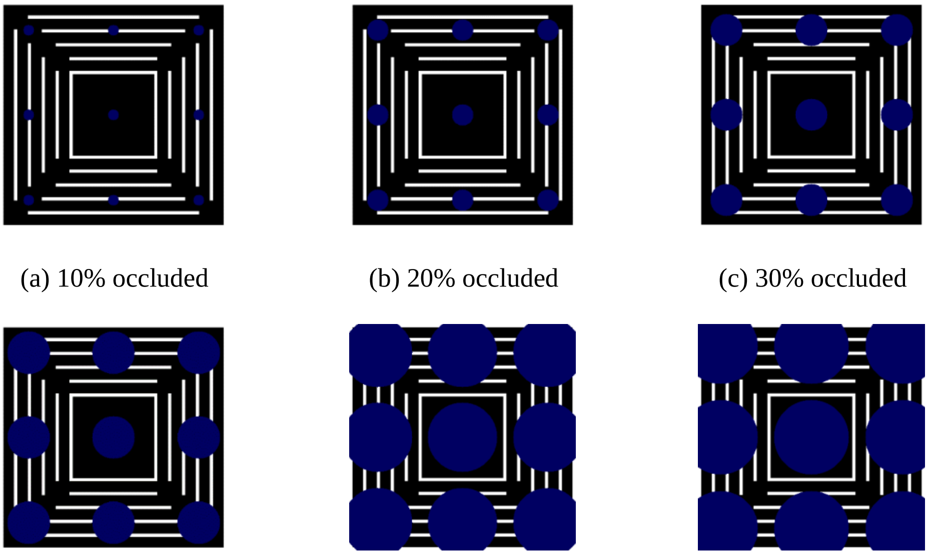
Figure 14: Snapshots of a live video showing different perspective views of a real marker embedding the value 2758

6.2 Robustness to viewing angle

The second test concerned robustness to viewing angle ‘β’. Markers were placed 1 m away from the camera and acquired with varying viewing angles β ∈ [10°: 90°]. The obtained results show that all the codes examined are indeed detected at angles greater than 15° (Figure 14). Beyond this angle, the proximity of neighbouring parallel lines increases, forming a homogeneous region that prevents accurate identification.

6.3 Robustness to occlusion

In these tests, we use a set of 50 unique OILU markers, each marked by a varying number of opaque circles ranging from 1 to 9 (Figure 15). By adjusting the size of these circles across seven different sizes, we generated a total of 3150 test images. The same process is adopted with the well-known April and ArUco Tags. Generated database is evaluated using dedicated exploitation codes. Obtained comparative tests (presented in Table 5) confirm that the suggested marker, characterized by its consistent line-based pyramidal structure, outperforms standard markers in handling difficult occlusion distortions. Identification fails if the occlusion rate exceeds 70% or both inner and outer quadrilaterals are partially occluded. Examples of snapshots from an available live video (Figure 16) show occulted markers identification cases in perspective view.



(a) 10% occluded

(b) 20% occluded

(c) 30% occluded

(d) 40% occluded

(e) 50% occluded

(f) 60% occluded

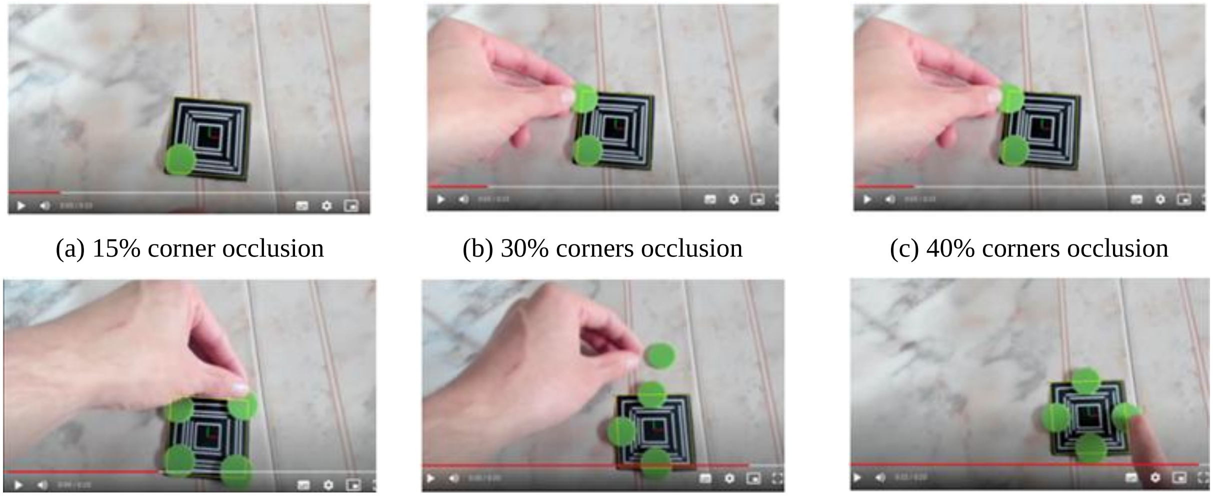
Figure 15: Occlusion tests using a set of 3150 synthetic markers. Opaque variable size circles are used for occlusion.

Table 5: Robustness to occlusion.

Occlusion (%)	Accuracy of the identification (in %)				
	OILU Tag	ArUco (36h12)	ArUco (16h3)	April (25h9)	April (36h10)
10%	100 %	72.67 %	100 %	100 %	100 %
20%	100 %	11.78%	100 %	23.17%	14.22 %
30%	100 %	10.44 %	100 %	17.87	10.89 %
40%	100 %	8.44 %	25.33 %	09.33 %	08.88 %
50%	100 %	1.56 %	17.78 %	04.53 %	01.78 %
60%	100 %	1.56 %	13.83 %	00.00 %	00.00 %

The tests shown in Figure 17 demonstrate the advantages of the OILU marker structure compared to the commonly used ArUco and AprilTag markers for pose estimation. While this task is relatively simple with is no occlusion (as shown in Figure 17 (a)), it becomes much more challenging when part of the marker is blocked. Markers like ArUco and AprilTag often struggle to maintain accuracy in these situations.

OILU markers, however, show strong resilience to partial occlusion, as illustrated in Figures 17 (b), (c), and (d). The design of OILU markers is particularly effective because, even if the outer quadrilateral is partially blocked, the pose can still be estimated using the inner quadrilateral. This feature makes OILU markers well-suited for environments where markers may not always be fully visible. However, OILU markers also exhibit certain limitations. When both the inner and outer quadrilaterals are occluded simultaneously, referred to as severe occlusion, pose estimation is no longer possible. Despite this, OILU markers significantly improve occlusion handling compared to ArUco and AprilTag markers, making them a more reliable choice in many real-world applications.



(a) 15% corner occlusion

(b) 30% corners occlusion

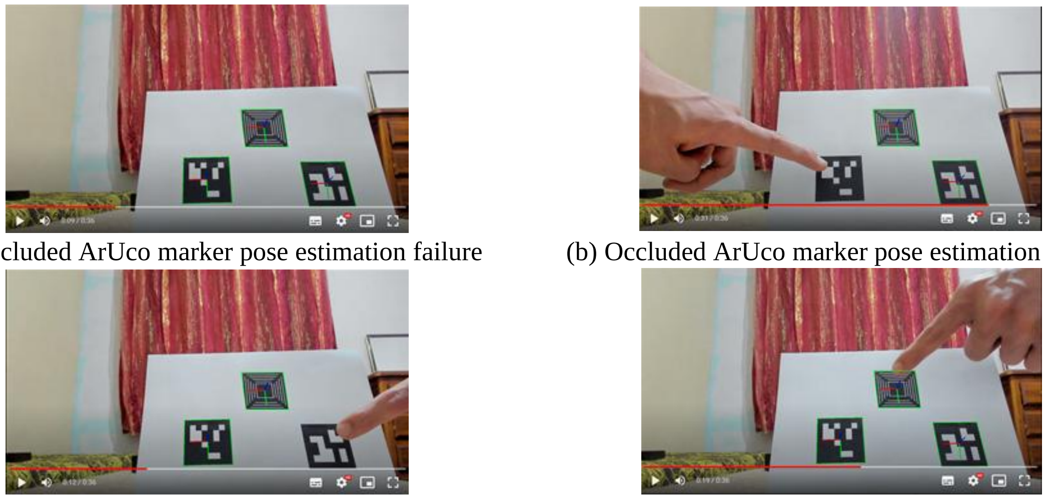
(c) 40% corners occlusion

(d) 50% corners occlusion

(e) 40% middle segments occlusion

(f) 50% middle segments occlusion

Figure 16: Snapshots of live demo showing occlusion tests with real markers acquired in perspective view.



(a) Occluded ArUco marker pose estimation failure

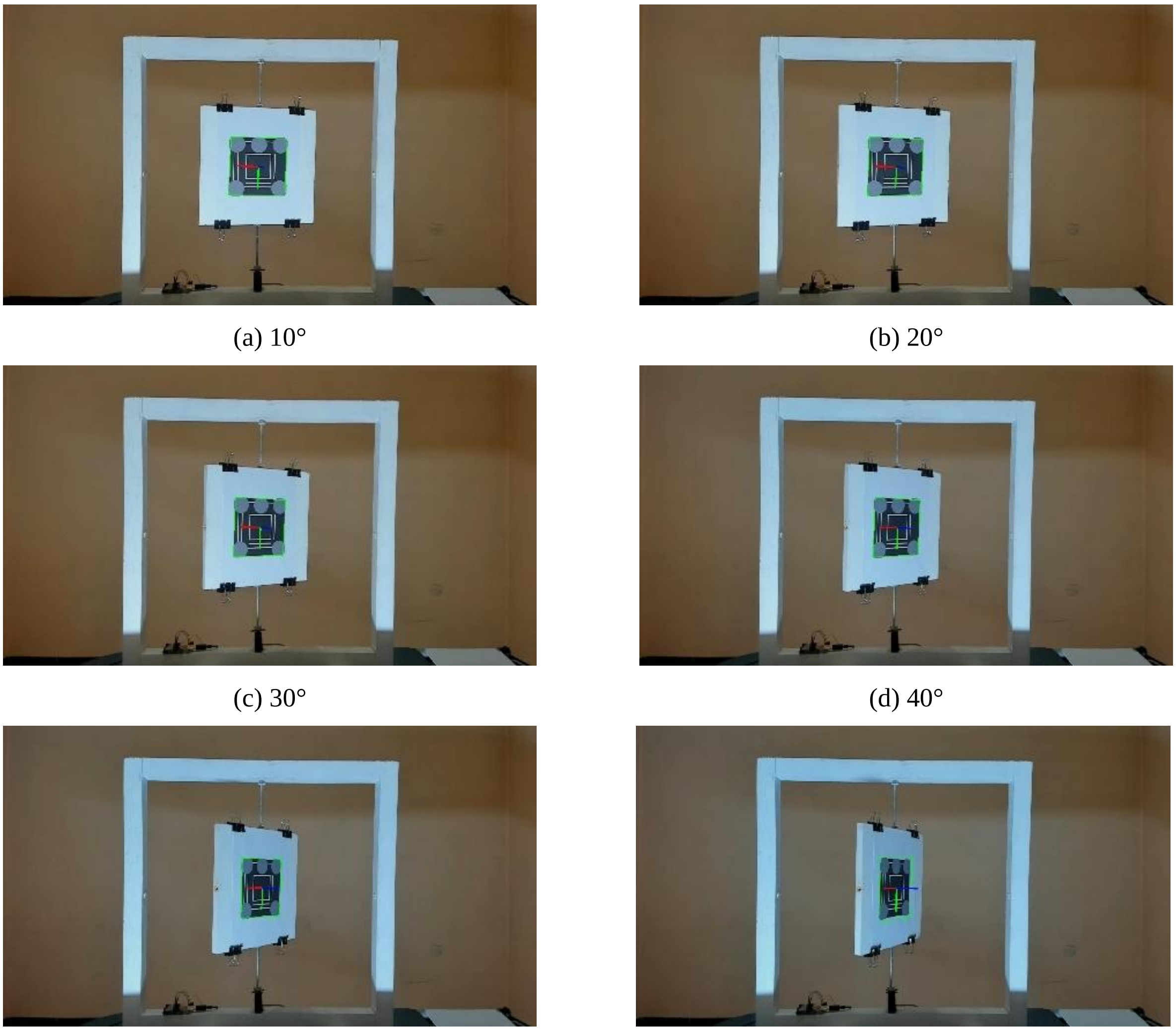
(b) Occluded ArUco marker pose estimation failure

(c) Occluded April marker pose estimation failure

(d) OILU Marker identification and pose estimation is possible even under occlusion

Figure 17: Snapshots from video comparing pose estimation using ArUco, April, and OILU Markers.

To further evaluate the robustness of the proposed OILU marker under occlusion, we conducted tests on real images with varying levels of occlusion and angular rotations. Figure 18 illustrates sample results for the OILU marker with 40% occlusion, where the marker was rotated at angles between 10° and 60°. Despite the challenging conditions, the OILU marker maintained detectable performance across all tested angles, showcasing its superior occlusion resilience. The images clearly demonstrate that the marker remains identifiable even at larger angles like 50° and 60°, where visibility is further reduced due to both occlusion and perspective distortion. Additionally, quantitative results presented in the angular error graph illustrated in Figure 19, confirm the impact of increasing occlusion levels (10%–50%) on pose estimation accuracy. Angular error tends to increase as occlusion levels rise, particularly beyond 30%. However, the proposed OILU marker still performs reliably. These results highlight the OILU marker's robustness in handling substantial occlusions and diverse viewing angles, demonstrating its reliability for pose estimation tasks in challenging real-world environments. The combination of visual examples and quantitative analysis emphasizes the marker's superior performance, particularly when other markers fail to deliver consistent results.



(a) 10°

(b) 20°

(c) 30°

(d) 40°

(e) 50°

(f) 60°

Figure 18:Pose estimation of OILU marker under 40% occlusion at different angles.

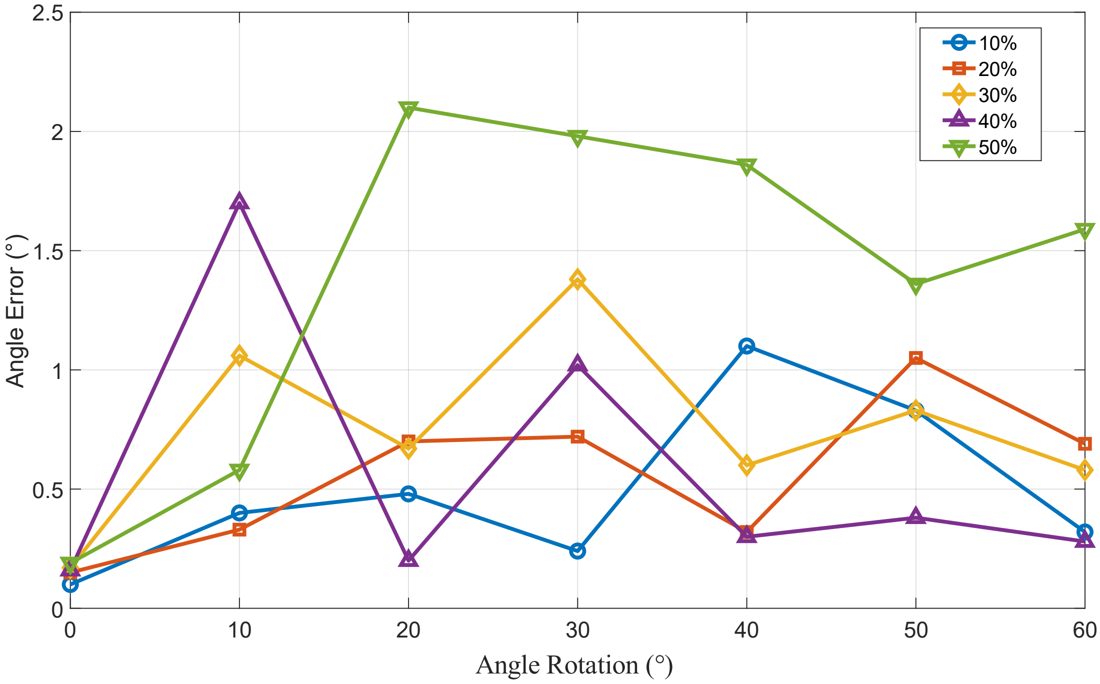


Figure 19: Effect of Occlusion (10%–50%) on Angular Error at Various Rotation Angles.

6.4 Execution time performance evaluation

The identification method described in Section 5.2 has been implemented and compared with the available ArUco [7] and April [8] tools using a laptop equipped with a 2.4 GHz Intel Core i7 processor with 16 GB RAM running on Linux. Reported processing times (Table 6) show that the proposed OILU system requires less processing time than the ArUco and April systems at all processed image resolutions. The gap between the different approaches is more important when dealing with multiple markers within the camera front of view (see Figure 20).

Such results confirm that the improved OILU solution outperforms state-of-the-art solutions in terms of rapid identification. The remaining challenge is developing a full hardware solution embedding the OILU marker identification process within a single System-on-Chip (SoC) device [31], ensuring fluid identification for highly constrained SLAM applications.

Table 6: Processing times with one marker (ms).

Image Resolution	OILU Tag	ArUco (36h12)	ArUco (16h3)	April (25h9)	April (36h10)
Low Resolution (640x480)	6,89	12,91	12,29	12,92	14,95
Medium Resolution (800x600)	8,51	12,98	12,99	13,60	15,44
High Resolution (1920x1080)	17,20	21,13	18,14	18,88	22,15

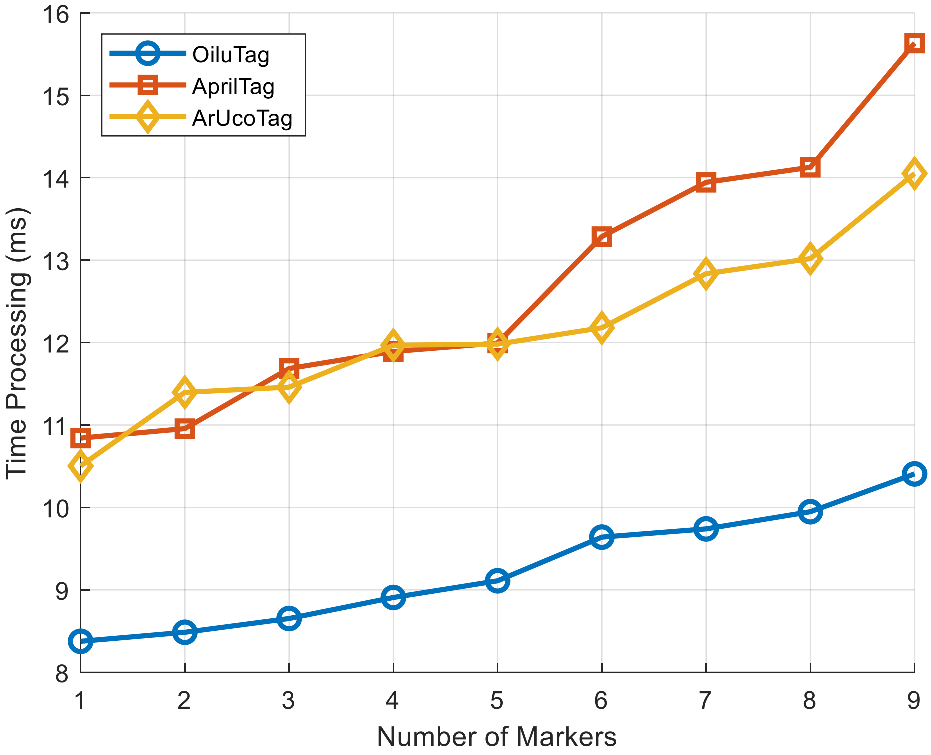


Figure 20: Processing time based on the number of markers within the camera front of view. Image resolution (1920×1080) and markers size (5cm×5cm).

7 Conclusion

An improved OILU marker system design is proposed for an accurate detection and identification scheme. Two approaches have been validated. The first one (based on cumulative histogram analysis) includes homography to process standard OILU markers. A second homography-less identification scheme is proposed to further improve marker detection and identification performances. The last involves enclosing the embedded identifier within two nested square-like quadrilaterals, allowing robust marker detection and identification even under challenging occlusion distortions.

Compared with the main state-of-the-art markers, the proposed approach presents approximately similar detection and identification results but with fewer computational resources and, consequently, less processing time. The suggested marker design, characterized by its consistent line-based pyramidal structure, surpasses standard markers in handling difficult occlusion distortions. Particular attention is paid to the possibility of identifying and estimating the pose of these markers, even if the external marker's corners are occluded.

At this stage, the developed marker does not integrate a corrector code. An improved OILU marker design, including a CRC code, is under development. It allows retaining or rejecting marker identifiers without significantly affecting the marker's codification capacities. Overall, the primary aim of this work is to underscore the potential benefits of employing uniform line-based 2D markers as a viable alternative to established state-of-the-art markers. Future work will extend the application of OILU markers to visual simultaneous localization and mapping (SLAM) projects, where markers are used to embed various environmental and orientation information exploited by uncrewed aerial vehicles (UAV) for accurate navigation and landing.

Acknowledgments

The authors would like to thank the Private Institute of Optics BBA for the support provided for the production and publication of this work.

References

[1] Alam, M.S., Gullu, A.I. & Gunes, A. Fiducial Markers and Particle Filter Based Localization and Navigation Framework for an Autonomous Mobile Robot. SN COMPUT. SCI. 5, 748 (2024) https://doi.org/10.1007/s42979-024-03090-y

[2] Romero-Ramire, F. J., Munoz-Salinas, R., Medina-Carnicer, R. Fractal Markers: a new approach for long-range marker pose estimation under occlusion. IEEE Access, (2019), 7, pp. 169908–

169919. http://dx.doi.org/10.1109/ACCESS.2019.2951204

[3] Zhenglong, G., Qiang, F., & Quan, Q. Pose estimation for multicopters based on monocular vision and AprilTag, 37th Chinese Control Conference (CCC), 2018, pp. 4717–4722. http://dx.doi.org/10.23919/ChiCC.2018.8483685

[4] Al-Dhaimesh, Sadeer Hel, and Nooriati Taib. "A review: investigation of augmented reality–BIM benefits in design process in AEC industry." Informatica 47.5 (2023). https://doi.org/10.31449/inf.v47i5.4671

[5] Sani, M. F., & Karimian, G. Automatic navigation and landing of an indoor AR. drone quadrotor using ArUco marker and inertial sensors. International Conference on Computer and Drone Applications (IConDA), 2017, 102–107. http://dx.doi.org/10.1109/ICONDA.2017.8270408

[6] Sarmadi, H., Muñoz-Salinas, R., M. Álvaro, B., Luna, A., Medina-Carnicer, R. 3D Reconstruction and alignment by consumer RGB-D sensors and fiducial planar markers for patient positioning in radiation therapy, Computer Methods and Programs in Biomedicine,Volume 180,2019,105004, https://doi.org/10.1016/j.cmpb.2019.105004

[7] Olson, E. AprilTag: A robust and flexible visual fiducial system. IEEE International Conference on Robotics and Automation, (2011), pp. 3400–3407. (2011) http://dx.doi.org/10.1109/ICRA.2011.5979561

[8] Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F., & Marín-Jiménez, M. Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recognition, 47, 2280–2292. (2014). http://dx.doi.org/10.1016/j.patcog.2014.01.005

[9] Rhijn, A., Jurriaan, M. Optical Tracking using Line Pencil Fiducials. (2004), 10.2312/EGVE/EGVE04/035-044.

[10] Chahir, Y., Mostefai, M., & Saidat, H. New Efficient Visual OILU Marker, The 25th International Conference on Image Processing Computer Vision, & Pattern Recognition (IPCV 2021), Book of Abstracts, 138, ISBN # 1-60132-514-2, (2021).

[11] Adalsteinsson, D., Sethian, J. A Fast Level Set Method For Propagating Interfaces, Comp Phys., (1995), Vol. 118, pp. 269-277. doi:10.1006/jcph.1995.1098

[12] Kato, H., & Billinghurst, M. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR’99),(1999), pp. 85–94. http://dx.doi.org/10.1109/IWAR.1999.803809

[13] Fiala, M. ARTag, a fiducial marker system using digital techniques. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), (2005), pp. 590–596. http://dx.doi.org/10.1109/CVPR.2005.74

[14] Fiala, M. Comparing ARTag and ARToolkit Plus fiducial marker systems. IEEE International Workshop on Haptic Audio Visual Environments and Their Applications, (2005). 6--pp. http://dx.doi.org/10.1109/HAVE.2005.1545669

[15] Yu, G., & Hu, Y., & Dai, J. TopoTag: A Robust and Scalable Topological Fiducial Marker System. IEEE Transactions on Visualization and Computer Graphics. (2020). http://dx.doi.org/10.48550/arXiv.1908.01450

[16] DeGol, J., Bretl, T., & Hoiem, D. Chromatag: A colored marker and fast detection algorithm. Proceedings of the IEEE International Conference on Computer Vision, (2017), pp. 1472–1481. https://doi.org/10.1109/ICCV.2017.16

[17] Calvet, L., Gurdjos, P., Griwodz, C., Gasparini, S. Detection and accurate localization of circular fiducials under highly challenging conditions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (2016), 562–570. http://dx.doi.org/10.1109/CVPR.2016.67

[18] Bergamasco, F., Albarelli, A., Torsello, A. Pi-tag: a fast image-space marker design based on projective invariants. Machine vision and applications, (2013), 24(6):1295–1310. http://dx.doi.org/10.1007/s00138-012-0469-6

[19] Bergamasco, F., Albarelli, A., Rodolà, E., Torsello, A. RUNE-Tag: A high accuracy fiducial marker with strong occlusion resilience. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (2011), pp.113 - 120. http://dx.doi.org/10.1109/CVPR.2011.5995544

[20] Birdal, T., Dobryden, I., Ilic, S. X-Tag: A Fiducial Tag for Flexible and Accurate Bundle Adjustment. (2016), pp. 556-564. http://dx.doi.org/10.1109/3DV.2016.65

[21] Burak, B., Cihan, T., Cuneyt, A. STag: A stable fiducial marker system, Image and Vision Computing, Vol 89, 2019, pp.158-169. https://doi.org/10.1016/j.imavis.2019.06.007

[22] Shingo, K., Hashimoto, K. Homography Estimation Using Marker Projection Control: A Case of Calibration-Free Projection Mapping, IFAC-Papers On Line, Vol 56, Issue 2, 2023, pp.

2951-2956. http://dx.doi.org/10.1016/j.ifacol.2023.10.1418

[23] Gonzalez, R. C., Woods, R. E. Digital image processing. Pearson Education limited, 4th Edition. (2017). https://doi.org/10.1117/1.3115362

[24] Suzuki, S., Be, K. , Topological structural analysis of digitized binary images by border following. Computer Vision, Graphics, and Image Processing, 30(1), (1985), 32–46. https://doi.org/10.1016/0734-189X(85)90016-7

[25] Douglas, D. H., & Peucker, T. K. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartographica: International Journal for Geographic Information and Geovisualization, 10(2), (1973), pp. 112–122. http://dx.doi.org/10.3138/FM57-6770-U75U-7727

[26] Li, Y., Zhu, S., Yu, Y., & Wang, Z. An improved graph-based visual localization system for indoor mobile robot using newly designed markers. International Journal of Advanced Robotic Systems, 15(2), (2018). https://doi.org/10.1177/1729881418769191

[27] Otsu, N. A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics, 9(1), (1979), pp. 62–66. http://dx.doi.org/10.1109/21.35351

[28] Xuancen, L., Shifeng, Z., Jiayi, T., Longbin, L. An Onboard Vision-Based System for Autonomous Landing of a Low-Cost Quadrotor on a Novel Landing Pad, Sensors, (2019), 19, 4703. https://doi.org/10.3390/s19214703

[29] Cong, Yingzi. "Image stitching technology for police drones using an improved image registration method incorporating ORB algorithm." *Informatica* 48.2 (2024). http://dx.doi.org/ 10.31449/inf.v48i2.5877

[30] Hartley, R., Zisserman, A. Multiple View Geomerty in Computer Vision. Cambridge University Press, second edition, 2003. https://doi.org/10.1017/s0263574700223217

[31] Vasileios, L., Panagiotis Minaidis, P., Lentaris, G., Dimitrios, S. Accelerating AI and Computer Vision for Satellite Pose Estimation on the Intel Myriad X Embedded SoC, Microprocessors and Microsystems, Volume 103, (2023). https://doi.org/10.1016/j.micpro.2023.104947

Q-Rung Orthopair Fuzzy Sets-Enhanced FMEA for COVID-19 Risk Assessment

Lazim Abdullah[1*], Noor Azzah Awang[2] and Muhammad Qiyas[3,4]
[1*]Special Interest Group of Modelling and Data Analytics, Faculty of Computer Science and Mathematics, Universiti Malaysia Terengganu, Kuala Nerus, 21030, Malaysia
[2]College of Computing, Informatics and Mathematics, Universiti Teknologi MARA, Shah Alam, 40450, Selangor, Malaysia
[3]Department of Mathematics, Riphah International University Failsabad Campus, Pakistan
[4]Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Besut Campus, 22200 Besut, Terengganu, Malaysia
E-mail: lazim_m@umt.edu.my, azzahawang@uitm.edu.my, m.qiyas160@gmail.com
[*]Corresponding author

Keywords: decision making, orthopair fuzzy sets, risk management, Covid-19, failure modes

Received: April 11, 2023

Failure Modes and Effects Analysis (FMEA) is a widely used tool for risk analysis, primarily to identify risk factors affecting system quality. Due to the limitations of the traditional FMEA model, several recent models incorporating advanced fuzzy set extensions have been developed to enhance the reliability of risk assessment outcomes. However, most of these models limit expert flexibility in expressing preferences and often overlook the impact of unequal expert weights and the stability of risk ranking results. This study introduces a new FMEA model based on Q-Rung Orthopair Fuzzy Sets (Q-ROFSs), termed Q-ROFSs-FMEA. Q-ROFSs, an extension of intuitionistic fuzzy sets, introduce a new linguistic term. The Q-ROFSs-FMEA model considers the unequal weights of experts, enabling a dynamic representation of expert preferences. These weights and the linguistic evaluation of risk factors are integrated through an aggregation operator, facilitating consensus among experts. The model is applied to a case study on COVID-19 risk factors, revealing that 'older age' (risk priority number 0.000012) is the highest risk factor, while 'gender' (risk priority number -0.0037) is the lowest. It is found that the ranking of risk factors determined by the Q-ROFSs-FMEA model is obtained as $FM_1 \succ FM_3 \succ FM_6 \succ FM_4 \succ FM_5 \succ FM_7 \succ FM_8 \succ FM_2$. *Furthermore, a comparative analysis indicates consistent ranking results across different models, demonstrating the reliability of the proposed model. The case study and comparative analysis validate the effectiveness and applicability of the Q-ROFSs-based risk assessment model.*

Povzetek: Razvili so model FMEA, izboljšan z uporabo Q-rung ortoparnih mehkih množic, za oceno tveganj COVID-19. Model omogoča dinamično izražanje preferenc strokovnjakov z neenakimi utežmi in združuje ocene tveganj prek agregacijskega operatorja. V študiji je bil kot najvišji dejavnik tveganja identificiran 'starejša starost', kot najnižji pa 'spol' (RPN -0,0037).

1 Introduction

Risk assessment is a crucial management tool for reducing project risks and promoting sustainable development. Risk assessment helps us to make the right decision especially when we are confronting problems with several alternatives and criteria [1]. There are numerous models available for determining risks and identifying hazards. The Failure Modes and Effects Analysis (FMEA) is one of the most widely used models since it is straightforward and efficient. It employs a proactive and systematic approach to identifying where and how it may fail [2]. Looking into the detailed part of FMEA, it is the process of analysing as many components, assemblies, and subsystems as possible to discover potential failure modes in a system and their causes and effects. FMEA has been widely applied across various sectors due to its proactive and systematic approach to failure identification. The FMEA is used to assess the relative impact of various failures such as in reducing medical errors [3], analysing the failure modes of nuclear-powered icebreakers, obtaining risk analysis for the textile industry's occupational safety and health [4], among many other applications. When mathematical failure rate models proposed by Tay and Lim [5] were linked with a statistical failure mode ratio database, the FMEA can be a qualitative analysis [6]. Therefore, FMEA is one of the earliest and most meticulously structured methods of failure analysis.

The FMEA was originally used in the aerospace sector in the 1960s, and it has been around for more than 60 years. Unlike other failure-prevention strategies, the FMEA was described in language that was universally understandable by those with minimal technical and/or systems knowledge. These encouraged the use of intelligent linguistic based approaches that are applicable to all enterprises and industries. The FMEA was brought into the mainstream by the automotive sector, which adopted it as the major mechanism for error and risk reduction. In recent years, the FMEA method has been widely used in a variety of fields, including manufacturing [7,8], aerospace [9], information technology risk assessment [10], healthcare risk management, [11,12] food industry [13,14], and maritime risk safety [15,16].

From a theoretical point of view, the FMEA comprises three components, namely the severity of the failure mode (*S*), the occurrence probability of a failure mode (*O*), and the detectability of the failure mode (*D*), which yields the risk priority number (RPN). The larger the RPN, the higher the risk of the related failure mode. The goal of RPN is to prioritise a product's or system's failure modes so that available resources can be appropriately allocated. The RPN can be expressed mathematically as the multiplication of *S, O* and *D* where they are risk parameters that are measured using a suitable point scale, such as Likert's scale [17]. According to Balaraju et al. [18] the FMEA team established an action approach based on the risk categories or risk rating level. For example, minor risk means no action is taken, moderate risk means some action is taken. Then, for high risk, corrective action will be taken, and for critical risk, corrective action will be taken, and major adjustments to the process/product will be necessary. However, in recent years it was argued that many types of risk assessments are difficult to obtain by the standard RPN. In an attempt to ease the assessment, Wang et al. [19] introduced the interval two-tuple linguistic representation model in FMEA. A dental manufacturing business uses the suggested linguistic FMEA technique to manufacture medical products. On the other hand, Huang et al. [20] employed probabilistic linguistic terms in FMEA instead of the normal linguistic term sets. The benefit of probabilistic linguistic terms is that they can handle the inherent ambiguity in FMEA team members' risk assessments without losing any information.

Risks are often associated with contradictory, subjective, ambiguous, or unclear information, making them well-suited for analysis using fuzzy set theory. Based on this assumption, the assessment model FMEA was integrated with fuzzy sets. The risk categories in the form of linguistics such as minor, moderate and high are closely related to the memberships of fuzzy set. The use of linguistic expressions to deal with uncertainty is one of the common aspects in fuzzy set-based risk models. For example, linguistics based on Pythagorean fuzzy set was employed in determining the risk performance of logistic service provider [21]. Recently, Huang et al. [22] proposed an integrated T-spherical fuzzy linguistic-FMEA. More works of FMEA that integrated with fuzzy set theory can be retrieved from Nie et al. [23] and Daneshvar et al. [24]. It was noticed that some of these integrated works used trapezoidal fuzzy set, [25, 26] triangular fuzzy set [27], and interval 2-tuple fuzzy linguistic variables [11, 28]. Recently, an integrated fuzzy set-FMEA was proposed by Ouyang et al. [29] where trapezoidal fuzzy numbers are used in defining linguistic variables. The use of linguistics in risk assessment has attracted many researchers because of its ability to deal with subjective and unclear notions.

In risk assessment models, fuzzy sets [30] that were represented by the membership functions allow the use of linguistic variables in FMEA to have a value between 0 and 1. However, some argued that a single membership function fails to address dual membership functions. Therefore, the intuitionistic fuzzy set (IFS) was proposed by Atanassov [31] where the total value of membership and non-membership may be greater than 1. However, in some real-world situations, the square sum of its dual memberships is equal to or less than 1 which is in violation of the condition of IFS. To solve the problem, Yager [32] developed the Q-rung Orthopair Fuzzy sets (Q-ROFSs) to overcome the ultimate limitation in which we can change the parameter value q to fulfil the value range requirement in a corresponding risk decision-making environment. The Q-ROFS is developed to deal with increasingly complex challenges where parameter value *q* is the notion of flexibility and variability. The capacity to evaluate a broader membership grade space with the parameter value *q* is the main benefit of these sets [33]. In other word, Q-ROFS is a new set for studying ambiguous information in a system. Compared to fuzzy sets, intuitionistic fuzzy sets, and Pythagorean fuzzy sets, this set is more potent and complete. Due to the inclusion of the parameter value *q*, the space of uncertain information described by the Q-ROFS is found to be enormous and flexible [34]. The current literature serves as the motivation for this paper to propose a novel FMEA model that can successfully address uncertainty issues. Some experts might not appreciate utilising crisp numbers to evaluate the failure modes while employing FMEA. They frequently employ linguistic variables or interval numbers to convey their ideas more effectively. In these circumstances, our suggested method is heavily emphasis the use of FMEA to combine heterogeneous information. Table 1 summarizes related research, highlighting current literature gaps that motivate this paper’s proposed FMEA model to address uncertainty more effectively.

Table 1: Summary of literature review

Authors	Year of publication	Type of sets used	Type of linguistic representation
Wang et al. [19]	2019	NA	Interval two-tuple
Huang et al. [20]	2022	NA	Probabilistic linguistic
Yalcinkaya and Cebi [21]	2022	Pythagorean fuzzy set	NA
Huang et al. [22]	2022	NA	T-spherical fuzzy linguistic
Nie et al. [23]	2018	NA	Multi-granular linguistic
Daneshvar et al. [24]	2020	Triangular and trapezoidal fuzzy set	NA
Wang et al. [25]	2017	Trapezoidal fuzzy set	NA

Wang et al. [26]	2017	Trapezoidal fuzzy set	NA
Testik and Unlu [27]	2022	Triangular fuzzy set	NA
Bhuvanesh Kumar and Parameshwaran [28]	2018	NA	Interval 2-tuple fuzzy linguistic
Ouyang et al. [29]	2021	NA	Trapezoidal fuzzy linguistic
Proposed method		Q-ROFSs	NA

Note: NA is an acronym for Not Available

The development of Q-ROFSs provides a valuable integration with FMEA, as both approaches address uncertainty in risk assessment. This paper proposes a Q-ROFS-FMEA model, where uncertainty is expressed through the linguistic variables of Q-ROFS. In this model, the linguistic variables for severity (*S*), occurrence (*O*), and detectability (*D*) in FMEA are replaced with Q-ROFS memberships, allowing for a nuanced representation of uncertainty. To illustrate the proposed work, a case study of risk factors of Coronavirus disease 2019 (COVID-19) will be implemented. In more detail, our proposed method can convert various assessment data into four-tuple linguistic variables that can be used to compute RPN. As a collective decision tool, FMEA requires input from a group of experts using linguistic terms. The uncertainty of information in FMEA is dealt with appropriately in the proposed model. The novel approach can handle the fuzziness and subjectivity in an uncertain environment, cover the diversity of viewpoints on the FMEA group of experts, and prevent the loss of crucial data throughout the risk assessment process. This method has the advantage of considering heterogeneous information as opposed to information of a single type.

The contributions of this paper are three-fold. First, we define ten linguistic terms for failure modes and develop two equations to transform interval-valued memberships into single-valued memberships and non-memberships within the Q-ROFSs framework. These terms are specifically applied to the *S*, *O*, and *D* components of FMEA. Second, the paper addresses heterogeneous expert input by assigning unequal weights to experts, reflecting differences in their opinions. Third, we demonstrate the model's application by identifying critical failure modes associated with COVID-19 risk factors. This paper is organised as follows. The next section recalls some prerequisite definitions and operations of Q-ROFSs. Section 3 presents the proposed Q-ROFS-FMEA risk assessment model. A case study of the risk factors of COVID-19 is illustrated in Section 4. In this section, detailed computational steps and results are presented. Finally, Section 5 concludes.

2 Preliminary

This section presents the definition of Q-ROFSs and its related operations.

Definition 2.1 Q-Rung Orthopair Fuzzy sets [32].

Let X be the universe of discourse. A Q-ROFSs $\tilde{Q}$ in X is denoted by

$\tilde{Q} = \left\{ < x, \mu_{\tilde{Q}}(x), v_{\tilde{Q}}(x) > | \, x \in X \right\}$,

where $\mu_{\tilde{Q}} : X \to [0,1]$ and $v_{\tilde{Q}} : X \to [0,1]$ signify the membership degree and the non-membership degree of the element $x \in X$ to the set $\tilde{Q}$, respectively with the limited condition $0 \le \mu_{\tilde{Q}}^q(x) + v_{\tilde{Q}}^q(x) \le 1$. The indeterminacy degree $\pi_{\tilde{Q}}(x) = \sqrt[q]{1 - \mu_{\tilde{Q}}^q(x) - v_{\tilde{Q}}^q(x)}$.

For convenience, Yager [32] termed $\left(\mu_{\tilde{Q}}(x), v_{\tilde{Q}}(x)\right)$ a Q-rung Orthopair Fuzzy number (Q-ROFN), which is signified as $\ddot{q} = \left(\mu_{\tilde{Q}}, v_{\tilde{Q}}\right)$.

Definition 2.2 Accuracy value $H(\tilde{Q})$ [35].

Let $\tilde{Q} = \left(\mu_{\tilde{Q}}, v_{\tilde{Q}}\right)$ be a Q-ROFN. The score value $S(\tilde{Q})$ of the Q-ROFN $\tilde{Q} = \left(\mu_{\tilde{Q}}, v_{\tilde{Q}}\right)$ is defined as $s(\tilde{Q}) = \mu_{\tilde{Q}}^q - v_{\tilde{Q}}^q$, where $S(\tilde{Q}) \in [-1,1]$ and $q \ge 1$. The accuracy value $H(\tilde{Q})$ of the Q-ROFN $\tilde{Q} = \left(\mu_{\tilde{Q}}, v_{\tilde{Q}}\right)$ is defined as $H(\tilde{Q}) = \mu_{\tilde{Q}}^q + v_{\tilde{Q}}^q$, where $H(\tilde{Q}) \in [0,1]$ and $q \ge 1$.

Definition 2.3 Accuracy values of the Q-ROFNs [35]

Let $\tilde{Q}_1 = \left(\mu_{\tilde{Q}_1}, v_{\tilde{Q}_1}\right)$ and $\tilde{Q}_2 = \left(\mu_{\tilde{Q}_2}, v_{\tilde{Q}_2}\right)$ be any two Q-ROFNs, and let $S(\tilde{Q}_1)$ and $S(\tilde{Q}_2)$ be the score values of the Q-ROFNs $\tilde{Q}_1$ and Q-ROFNs $\tilde{Q}_2$ respectively.

Let $H(\tilde{Q}_1)$ and $H(\tilde{Q}_2)$ be the accuracy values of the Q-ROFNs $\tilde{Q}_1$ and Q-ROFNs $\tilde{Q}_2$, respectively,

(1) If $S(\tilde{Q}_1) > S(\tilde{Q}_2)$, then $\tilde{Q}_1 > \tilde{Q}_2$.

(2) If $S(\tilde{Q}_1) = S(\tilde{Q}_2)$ and $H(\tilde{Q}_1) > H(\tilde{Q}_2)$, then $\tilde{Q}_1 > \tilde{Q}_2$.

(3) If $S(\tilde{Q}_1) = S(\tilde{Q}_2)$ and $H(\tilde{Q}_1) = H(\tilde{Q}_2)$, then $\tilde{Q}_1 = \tilde{Q}_2$

Many scholars have studied and expanded mathematical operations over Q-ROFSs, a fascinating topic with many obstacles. The following are the basic activities outlined by Peng and Luo [36].

(1) Complement, $\ddot{q}^c = (\mu, v)$

(2) Union, $\ddot{q}_1 \cup \ddot{q}_2 = \left(\max\{\mu_1, \mu_2\}, \min\{v_1, v_2\}\right)$

(3) Intersection, $\ddot{q}_1 \cap \ddot{q}_2 = \left(\min\{\mu_1, \mu_2\}, \max\{v_1, v_2\}\right)$

(4) Subset, $\ddot{q}_1 \subseteq \ddot{q}_2 iff \, \mu_1 \le \mu_2, v_1 \ge v_2$

(5) Addition,

$$\ddot{q}_1 \oplus \ddot{q}_2 = \left(\left(\mu_1^q + \mu_2^q - \mu_1^q \cdot \mu_2^q \right)^{\frac{1}{q}}, v_1 \cdot v_2 \right)$$

(6) Multiplication.

$$\ddot{q}_1 \otimes \ddot{q}_2 = \left(\mu_1 \cdot \mu_2, \left(v_1^q + v_2^q - v_1^q \cdot v_2^q \right)^{\frac{1}{q}} \right)$$

These definitions of concepts and their operations are directly used in the computational implementation of the proposed work.

3 Proposed Q-ROFSs-FMEA

The FMEA is a computational tool that proactively strategized for examining a process whether it might fail. The tool is also used to analyse the relative impact of various failures in which process aspects that need to be altered the most can be identified. This section presents a new proposed Q-ROFS-FMEA where Q-ROFS and FMEA are combined. The computational procedures of fuzzy sets-FMEA method that was proposed by Ouyang et al. [29] become the basis in this work. To make it compatible with Q-ROFSs setting, several innovations to FMEA are made. The first innovation is the use of Q-ROFSs in defining linguistic terms where four-tuple number is used instead of one single number. To recognise the difference in experts' opinions and heterogenous information, unequal relative weight of experts is introduced as the second innovation. Finally, an aggregation operator is introduced to merge expert opinions of which a consensus RPN can be obtained. Details of these innovations are further explained in the computational procedures of the proposed Q-ROFSs-FMEA. The computational procedures of the proposed work are presented as follows.

Step 1: Determine the failure modes

To identify all probable failure modes denoted by $FM = \{FM_1, FM_2, \ldots, FM_m\}$ indicates the m failure modes that results in system failure, the experts $e_k (k = 1, 2, \ldots, l)$ with suitable expertise and experience are invited.

Step 2: Estimate the failure modes by linguistic terms.

Assessment scale of failure modes are made using linguistic terms due to uncertainty and ambiguity of human perceptions and heterogenous information. In this step, a new linguistic term is proposed. The linguistic terms proposed by Jin et al. [37] becomes the basis in this effort. Interval number of memberships in the work of Jin et al. [37] is simplified and transformed into memberships of Q-ROFSs. This transformation is made using Equation (1) and Equation (2) subjected to the condition $0 \le \mu_{\tilde{Q}ROF}^q(x) + v_{\tilde{Q}ROF}^q(x) \le 1$.

$$\left(\mu_{\tilde{Q}ROF}(x)\right) = \left(\left| \frac{\mu_{\tilde{Q}_1}(x) + \mu_{\tilde{Q}_2}(x)}{2} \right| \right) \tag{1}$$

$$\left(v_{\tilde{Q}ROF}(x)\right) = \left(1 - \mu_{\tilde{Q}ROF}\right) \tag{2}$$

where, $\mu_{\tilde{Q}ROF}(x)$ is a membership degree corresponding Q-ROFSs, and $v_{\tilde{Q}ROF}(x)$ is non-membership degrees corresponding Q-ROFSs.

For example, if the interval membership is ([0.99, 0.99], [0.01, 0.01]) then, by using Equation (1), $\left(\mu_{\tilde{Q}ROF}(x)\right) = \left(\left| \frac{0.99 + 0.99}{2} \right| \right) = 0.99$. Then using Equation (2), we have $\left(v_{\tilde{Q}ROF}(x)\right) = (1 - 0.99) = 0.01$. The similar transformations are made for other linguistic terms. Summarily, the new linguistic terms are presented in Table 2.

Table 2: The linguistic terms of Q-ROFSs

Scales	Linguistic terms	Q-ROFSs
0	Exceptionally high	(0.99,0.01)
1	Extremely High	(0.90,0.10)
2	Very High	(0.80,0.20)
3	High	(0.675,0.325)
4	Medium High	(0.525,0.475)
5	Medium	(0.50,0.50)
6	Medium Low	(0.40,0.60)
7	Low	(0.30,0.70)
8	Very Low	(0.175,0.825)
9	Extremely Low	(0.10,0.90)

To measure the risks and to make it compatible with the FMEA model, the linguistic terms are changed to linguistic of Severity (*S*), Occurrence (*O*) and Detection (*D*). Table 3 provides the linguistic terms for *S*, *O*, and *D* with Q-ROFSs.

Table 3: The linguistic terms for Severity (*S*), Occurrence (*O*) and Detection (*D*) with Q-ROFSs

Scale	Severity (*S*)	Occurrence (*O*)	Detection (*D*)	Q-ROFSs
9	Hazardous	Almost certain	Almost impossible	(0.99,0.01)
8	Serious	Very High	Very Remote	(0.90,0.10)
7	Very High	High	Remote	(0.80,0.20)
6	High	Moderately High	Very Low	(0.675,0.325)
5	Moderate	Moderately	Low	(0.525,0.475)
4	Low	Moderately low	Moderate	(0.50,0.50)
3	Very Low	Low	Moderately high	(0.40,0.60)
2	Slight	Slight	High	(0.30,0.70)
1	Very Slight	Remote	Very High	(0.175,0.825)
0	None	Almost Never	Almost certain	(0.10,0.90)

Step 3: Determine the weights of experts

There is an innovation in this step where weights of experts are introduced. Differently from Ouyang et al. [29] where no weight was introduced, this step introduces weights in which these weights are crucial as it represents the difference of human perceptions and heterogenous information.

The weights, λ_k for kth experts, are calculated using Equation (3) with q is a constant. The total weight of experts must equal to one.

$$\tilde{\lambda}_k = \frac{1-((1-\mu_{\tilde{Q}ROF}(x))+v_{\tilde{Q}ROF}(x))/2)}{\sum_{k=1}^{l}1-((1-\mu_{\tilde{Q}ROF}(x))+v_{\tilde{Q}ROF}(x))/2)} \qquad (3)$$

where $\sum_{k=1}^{l}\tilde{\lambda}_k = 1$

The linguistic terms used in finding the weight of experts is presented in Table 4.

Table 4: The weight of expert's preference scale

Scale	Linguistic terms	Corresponding Q-ROFSs
9	Exceptionally Important	(0.99,0.01)
8	Extremely Important	(0.90,0.10)
7	Very Important	(0.80,0.20)
6	Important	(0.675,0.325)
5	Medium Important	(0.525,0.475)
4	Neutral	(0.50,0.50)
3	Medium not Important	(0.40,0.60)
2	Not Important	(0.30,0.70)
1	Very Not Important	(0.175,0.825)
0	Extremely Not Important	(0.10,0.90)

The weights obtained here will be used for the next step of computational procedures.

Step 4: Aggregate the assessment of experts

In this step, the Q-rung Orthopair fuzzy weighted averaging operator (q-ROFWA) proposed by Liu and Wang [35] is used to aggregate the assessment of experts. An aggregated matrix to represent assessments made by k-th experts are calculated using Equation (4).

$$q-ROFWA(E_1,E_2,\ldots,E_l)=\left\langle \sqrt[q]{1-\prod_{k=1}^{l}\left(1-\mu_k^q\right)^{\tilde{\lambda}_k}}, \prod_{k=1}^{l} v_k^{\tilde{\lambda}_k} \right\rangle \qquad (4)$$

Differently from Ouyang et al. [29] where no aggregation equation is used, an aggregation is inserted at this step. This aggregation operation is significant as it combines all expert opinions to become a consensus decision.

The q-ROFWA operator is employed because it effectively incorporates the weights of experts, making it well-suited for our context. Its simplicity allows for a balanced aggregation that accurately reflects expert consensus without adding unnecessary complexity. Additionally, the q-ROFWA operator was selected over other averaging methods due to its compatibility with q-ROF numbers, which supports a more precise representation of expert opinions.

Step 5: Determine score function values of failure modes

Score function, $s(\tilde{Q})$ is used for the defuzzification process. Equation (5) is used to find a crisp value.

$$s(\tilde{Q}) = \mu_{\tilde{Q}ROF}(x) - v_{\tilde{Q}ROF}(x) \qquad (5)$$

Step 6: Calculate the RPN of failure modes using the multiplication operator of *S*, *O* and *D* (See Equation (6)).

$$RPN = S \times O \times D \qquad (6)$$

where *S*, *O*, and *D* are risk parameters.

Step 7: Rank the failure modes using RPN results.

The final RPN results can be ranked in ascending order and the highest failure mode can be identified. The proposed computational procedures will be implemented in a case study investigating risk factors of COVID-19. Detailed computations and results will be discussed in the following section.

4 A case study of COVID-19 failure modes

This section describes the failure modes of COVID-19, the experts who are giving their assessment, and the proposed computational model used to implement the computation.

4.1 Failure modes

The list of failure modes for COVID-19 disease is defined. Table 5 shows the failure modes considered in this study and their respective literature sources.

Table 5: Selected failure modes of COVID-19

No.	Failure mode	Source of Literature
1	Older age (FM_1)	Rashedi, et al. [38] and Jordan et al. [39]
2	Gender (FM_2)	Gebhard et al. [40], Rashedi, et al. [38] and Ambrocino et al. [41]
3	Individual medical condition (FM_3)	De Sousa Lima et al. [42]
4	Occupational factors (FM_4)	Leso et al. [43]
5	Poor ventilation (FM_5)	Rashedi, et al. [38]
6	Low education (FM_6)	Rashedi, et al. [38]
7	Transmissibility (FM_7)	Rashedi, et al. [38]
8	Viral load COVID-19 and its receptor, ACE2 (FM_8)	Rashedi, et al. [38]

4.2 Experts' information

Five experts were invited to contribute their insights in assessing COVID-19 failure modes. A summary of their profiles is provided in Table 6.

Table 6: Biographical data of experts

Expert	Designation	Experience (year)	Academic
E_1	Senior Nurse	10	B,Sc Nursing, Community Health Nursing Certification
E_2	Senior Nurse	19	B,Sc Nursing, Community Health Nursing Certification
E_3	Public Health Expert	16	MBBS, MPH
E_4	Public Health Expert	11	MBBS, MPH
E_5	Nurse	5	B,Sc Nursing, Community Health Nursing Certification

The experts provide an assessment of failure modes and then analyse using FMEA.

4.3 Data

Specifically, eight failure modes for COVID-19, denoted as $(FM_1, FM_2, FM_3, FM_4, FM_5, FM_6, FM_7, FM_8)$ were evaluated by a group of experts $(E_1, E_2, E_3, E_4, E_5)$. Each expert assessed the failure modes based on severity, occurrence, and detection using a scale from zero to nine. To ensure consistency and minimize subjective bias, the experts were provided with Table 3, which outlines the numerical scale alongside its corresponding linguistic terms. Additionally, a brief training session was conducted to standardize the experts' understanding of these linguistic terms, enhancing alignment throughout the evaluation process.

Table 7: Assessment of severity, occurrence, and detection

Expert	FM	Severity (S)	Occurrence (O)	Detection (D)
E_1	FM_1	8	8	8
	FM_2	0	0	0
	FM_3	8	7	7
	FM_4	8	9	5
	FM_5	7	9	5
	FM_6	2	4	8
	FM_7	8	9	3
	FM_8	9	9	1
E_2	FM_1	9	9	8
	FM_2	8	6	1
	FM_3	9	7	8
	FM_4	9	8	6
	FM_5	7	9	4
	FM_6	3	8	8
	FM_7	9	9	3
	FM_8	9	9	2
E_3	FM_1	9	9	8
	FM_2	6	2	0
	FM_3	9	9	9
	FM_4	9	9	4
	FM_5	9	9	0
	FM_6	7	9	8
	FM_7	9	9	2
	FM_8	9	9	2
E_4	FM_1	9	9	8
	FM_2	6	2	2
	FM_3	9	9	6
	FM_4	9	9	6
	FM_5	9	9	0
	FM_6	9	7	9
	FM_7	9	9	3
	FM_8	9	9	3
E_5	FM_1	7	6	8
	FM_2	1	1	3
	FM_3	8	8	5
	FM_4	8	5	6
	FM_5	6	6	5
	FM_6	7	5	6
	FM_7	8	7	4
	FM_8	6	8	1

The heterogenous information from the above table are regarded as the input data in which these data are then computed in accordance with the proposed Q-ROFSs-FMEA (see Section 3).

4.4 Computation and results

The Q-ROFSs-FMEA method is implemented for the case of failure modes of COVID-19 disease. This subsection presents the detailed computations of the input data using the Q-ROFSs-FMEA method.

Step 1: Determine the failure modes
The list of COVID-19 failure modes is provided in Section 4.1, and the experts' biographical information is detailed in Section 4.2.

Step 2: Estimate the failure modes by using linguistic terms.
The linguistic 0-9 scales from Table 7 are converted to matrix form in Q-ROFSs information and the resulting $\ddot{S}, \ddot{O}, \ddot{D}$ matrices are shown as,

$$\ddot{S} = \begin{bmatrix} \langle 0.900, 0.100 \rangle & \langle 0.990, 0.010 \rangle & \cdots & \langle 0.800, 0.200 \rangle \\ \langle 0.100, 0.900 \rangle & \langle 0.900, 0.100 \rangle & \ldots & \langle 0.175, 0.825 \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle 0.990, 0.010 \rangle & \langle 0.990, 0.100 \rangle & \cdots & \langle 0.675, 0.325 \rangle \end{bmatrix}_{8\times 5}$$

$$\ddot{O} = \begin{bmatrix} \langle 0.900, 0.100 \rangle & \langle 0.990, 0.010 \rangle & \cdots & \langle 0.675, 0.325 \rangle \\ \langle 0.100, 0.900 \rangle & \langle 0.675, 0.325 \rangle & \ldots & \langle 0.175, 0.825 \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle 0.990, 0.010 \rangle & \langle 0.990, 0.010 \rangle & \cdots & \langle 0.900, 0.100 \rangle \end{bmatrix}_{8\times 5}$$

$$\ddot{D} = \begin{bmatrix} \langle 0.900, 0.100 \rangle & \langle 0.900, 0.100 \rangle & \cdots & \langle 0.900, 0.100 \rangle \\ \langle 0.100, 0.900 \rangle & \langle 0.175, 0.825 \rangle & \ldots & \langle 0.400, 0.600 \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle 0.100, 0.900 \rangle & \langle 0.300, 0.700 \rangle & \cdots & \langle 0.175, 0.825 \rangle \end{bmatrix}_{8x5}$$

Step 3: Determine the weights of experts.

The linguistic terms defined in Table 4 are used to determine the weights of experts. The importance of experts is represented by a linguistic term and its corresponding Q-ROFSs. Table 8 shows the linguistic terms and their respective Q-ROFSs which reflect the importance of experts.

Table 8: Importance of experts in Q-ROFSs

Expert	Linguistic term	Q-ROFSs
E_1	Neutral	$\langle 0.500, 0.500\rangle$
E_2	Very Important	$\langle 0.800, 0.200\rangle$
E_3	Important	$\langle 0.675, 0.375\rangle$
E_4	Neutral	$\langle 0.500, 0.500\rangle$
E_5	Medium not Important	$\langle 0.400, 0.600\rangle$

With the assumption that weights of experts are unequal, then Equation (3) is used to compute relative weights of experts. Given the information in Table 8, weight for the first expert, $\tilde{\lambda}_1$ for example is computed as

$$\tilde{\lambda}_1 = \frac{1-((1-0.500^3)+0.500^3)/2)}{\begin{bmatrix}\left(1-((1-0.500^3)+0.500^3)/2)\right)+\left(1-((1-0.800^3)+0.200^3)/2)\right)\\ +\left(1-((1-0.675^3)+0.375^3)/2)\right)+\left(1-((1-0.500^3)+0.500^3)/2)\right)+\\ \left(1-((1-0.400^3)+0.600^3)/2)\right)\end{bmatrix}}$$

$$\tilde{\lambda}_1 = 0.178$$

Similarly, the weights for other experts are calculated and summarised in Table 9.

Table 9: Weight of experts

Expert	Weights
E_1	0.178
E_2	0.268
E_3	0.224
E_4	0.178
E_5	0.151

Step 4: Aggregate the evaluation from different experts using q-ROFWA aggregation operator of matrices, $\bar{S},\bar{O},\bar{D}$. The aggregated matrices to represent assessments made by *k-th* experts are calculated using Equation (4). For example, the aggregated value of FM_1 is computed as follows.

$$\bar{s}_{FM_1} = \left\langle \sqrt[3]{1-\prod_{k=1}^{l}\begin{bmatrix}\left(\left(1-0.900^3\right)^{0.178}\right)\left(\left(1-0.990^3\right)^{0.268}\right)\left(\left(1-0.990^3\right)^{0.224}\right)\\ \left(\left(1-0.990^3\right)^{0.178}\right)\left(\left(1-0.800^3\right)^{0.151}\right)\end{bmatrix}}, \atop \prod_{k=1}^{l}\left[\left(0.100^{0.178}\right)\left(0.010^{0.268}\right)\left(0.010^{0.224}\right)\left(0.010^{0.178}\right)\left(0.200^{0.151}\right)\right] \right\rangle$$

$$= \langle 0.9771, 0.0237\rangle$$

$$\bar{o}_{FM_1} = \left\langle \sqrt[3]{1-\prod_{k=1}^{l}\begin{bmatrix}\left(\left(1-0.900^3\right)^{0.178}\right)\left(\left(1-0.99^3\right)^{0.268}\right)\left(\left(1-0.99^3\right)^{0.224}\right)\\ \left(\left(1-0.99^3\right)^{0.178}\right)\left(\left(1-0.675^3\right)^{0.151}\right)\end{bmatrix}}, \atop \prod_{k=1}^{l}\left[\left(0.1^{0.178}\right)\left(0.01^{0.268}\right)\left(0.01^{0.224}\right)\left(0.01^{0.178}\right)\left(0.325^{0.151}\right)\right] \right\rangle$$

$$= \langle 0.9758, 0.0255\rangle$$

$$\bar{d}_{FM_1} = \left\langle \sqrt[3]{1-\prod_{k=1}^{l}\begin{bmatrix}\left(\left(1-0.900^3\right)^{0.178}\right)\left(\left(1-0.900^3\right)^{0.268}\right)\left(\left(1-0.900^3\right)^{0.224}\right)\\ \left(\left(1-0.900^3\right)^{0.178}\right)\left(\left(1-0.900^3\right)^{0.151}\right)\end{bmatrix}}, \atop \prod_{k=1}^{l}\left[\left(0.100^{0.178}\right)\left(0.100^{0.268}\right)\left(0.100^{0.224}\right)\left(0.100^{0.178}\right)\left(0.100^{0.151}\right)\right] \right\rangle$$

$$= \langle 0.90, 0.10\rangle$$

It is good to note that while parameter value q can be varied, in this computation q=3 is chosen as to cushion the impact of non- membership with negation of membership. The aggregated matrices $\bar{S},\bar{O},\bar{D}$ are shown as

$$\bar{S} = \begin{array}{c}\langle 0.977, 0.0237\rangle\\ \langle 0.732, 0.327\rangle\\ \langle 0.979, 0.021\rangle\\ \langle 0.979, 0.021\rangle\\ \langle 0.941, 0.065\rangle\\ \langle 0.844, 0.197\rangle\\ \langle 0.979, 0.021\rangle\\ \langle 0.984, 0.017\rangle\end{array} \qquad \bar{O} = \begin{array}{c}\langle 0.976, 0.026\rangle\\ \langle 0.471, 0.611\rangle\\ \langle 0.949, 0.054\rangle\\ \langle 0.969, 0.033\rangle\\ \langle 0.984, 0.017\rangle\\ \langle 0.01, 0.114\rangle\\ \langle 0.985, 0.016\rangle\\ \langle 0.986, 0.014\rangle\end{array}$$

$$\bar{D} = \begin{array}{c}\langle 0.9, 0.1\rangle\\ \langle 0.255, 0.791\rangle\\ \langle 0.905, 0.106\rangle\\ \langle 0.623, 0.383\rangle\\ \langle 0.438, 0.623\rangle\\ \langle 0.924, 0.079\rangle\\ \langle 0.403, 0.604\rangle\\ \langle 0.299, 0.719\rangle\end{array}$$

Step 5: Determine score function values of $\bar{S},\bar{O},\bar{D}$.

Score function is used for the defuzzification process (see Equation (5)). The score function values of $\bar{S},\bar{O},\bar{D}$ are shown in Table 10.

Step 6: Calculate the RPN of failure modes using the product of *S*, *O*, *D* using Equation (6). For example, RPN of FM_1 in the last column of Table 10 can be calculated as

$$RPN_{FM_1} = 0.953 \times 0.95 \times 0.8$$
$$= 0.725$$

The similar operation is implemented to other failure modes.

Step 7: Rank the failure modes using RPN results.

The ranking of failure modes using Q-ROFS-FMEA method is obtained as shown in Table 10.

Table 10: The score function of $\bar{S}, \bar{O}, \bar{D}$, RPN and ranking

FM	Score function			RPN	Rank
	$\bar{S}$	$\bar{O}$	$\bar{D}$		
FM$_1$	0.953	0.950	0.800	0.725	1
FM$_2$	0.405	-0.140	-0.536	0.030	5
FM$_3$	0.958	0.895	0.800	0.686	2
FM$_4$	0.958	0.936	0.240	0.216	4
FM$_5$	0.876	0.967	-0.185	-0.157	6
FM$_6$	0.647	0.787	0.845	0.430	3
FM$_7$	0.958	0.969	-0.201	-0.187	7
FM$_8$	0.967	0.972	-0.420	-0.395	8

The above result shows the RPN of each failure modes in which eventually can portray the rank of RPN. It indicates that *FM$_1$* (older people) is the highest risk factor and *FM$_2$* (gender) is the lowest risk factor of COVID-19. The final results are subjected to comparative analysis of which will be explained in the following section.

5 Comparative analysis

The same data used to determine the ranking using the proposed Q-ROFSs-FMEA is then computationally reiterated using the existing FMEA methods such as crisp FMEA, Triangular Fuzzy Number FMEA (TFN-FMEA), and Intuitionistic Fuzzy Set FMEA (IFS-FMEA). It is good to note here that the existing FMEA method is the method used without considering the Q-ROFSs. Table 11 shows the comparison ranking of failure modes based on Q-ROFS-FMEA method alongside other FMEA methods.

Table 11: The ranking of failure modes

FM	RPN			
	FMEA	TFN-FMEA	IFS-FMEA	Q-ROFS-FMEA
FM$_1$	450.378(1)	0.846 (1)	0.572 (1)	0.725 (1)
FM$_2$	91.456(8)	0.021 (8)	0.006 (5)	0.030 (5)
FM$_3$	437.326(2)	0.696 (2)	0.393 (2)	0.686 (2)
FM$_4$	412.574(3)	0.559 (3)	0.212 (3)	0.216 (4)
FM$_5$	235.316(7)	0.231 (6)	-0.220 (6)	-0.157 (6)
FM$_6$	381.611(4)	0.419 (4)	0.033 (4)	0.430 (3)
FM$_7$	369.809(5)	0.246(5)	-0.222 (7)	-0.187 (7)
FM$_8$	334.708(6)	0.174 (7)	-0.331 (8)	-0.395 (8)

It can be seen that the RPN values obtained from FMEA are much higher compared to the RPN obtained from TFN-FMEA, IFS-FMEA and Q-ROFSs-FMEA. The main reason behind this big difference is because the type of numbers used. In the FMEA method, assessments are made using real numbers from 0 to 9, whereas fuzzy numbers between 0 and 1 are utilized in TFN-FMEA, IFS-FMEA, and Q-ROFS-FMEA. The RPN values in Q-ROFS-FMEA are significantly lower due to the use of four-tuple values, which represent the membership degrees within Q-ROFSs.

Furthermore, the RPNs obtained from the methods are used to compare the ranking of risk factors (*FMs*). The comparison of these ranks and their respective RPNs can be seen in Figure 1.

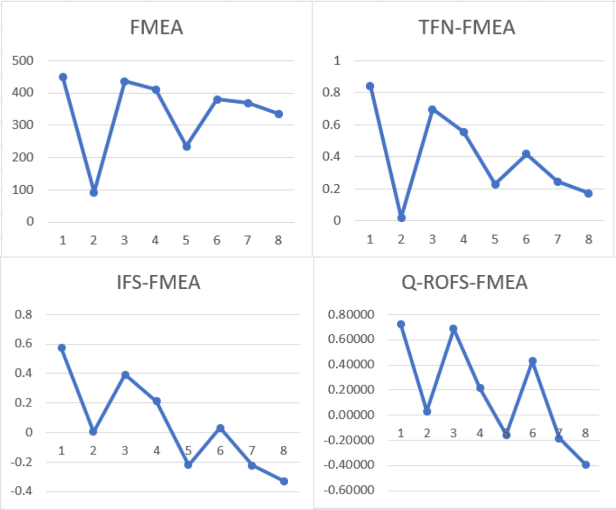


Figure 1: Comparison of RPN values obtained using Q-ROFS- FMEA versus other methods

The ranks of failure modes of COVID-19 obtained using the proposed Q-ROFS-FMEA and some existing FMEA methods are shown in Figure 2.

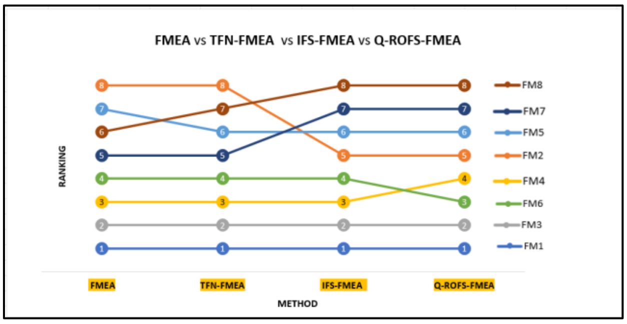


Figure 2: Comparison of *FM* ranking obtained using Q-ROFS- FMEA versus other methods

The rankings of failure modes across these four methods show a high level of consistency. Both the first FM_1 (older age) and second FM_3 (individual medical condition) ranks are the same across all methods. Minor shifts are observed in the third and fourth ranks, but there are significant changes from the fifth rank onward among the methods. Unlike other methods, the traditional FMEA approach does not account for fuzziness or uncertainty, while TFN-FMEA only considers the membership degree and omits the non-membership aspect of the problem. IFS-

FMEA, on the other hand, incorporates both membership and non-membership degrees and can partially address hesitancy. The proposed Q-ROFS-FMEA generates ranking results closely aligned with IFS-FMEA, primarily due to its ability to manage uncertainty. However, Q-ROFS-FMEA offers an advantage by introducing the *q* parameter, which provides enhanced flexibility not available in IFS-FMEA. Although IFS-FMEA includes both non-membership and hesitancy degrees, it has been criticized for limitations in practical applications, as its dual memberships must sum to one or less, which can restrict its adaptability.

Applying Q-ROFS to FMEA offers greater flexibility, as the parameter *q* can be adjusted to meet the specific requirements of various risk decision-making contexts. The Q-ROFS framework is especially well-suited to handle complex scenarios, with the parameter *q* providing added adaptability. This ability to adjust membership grade space via *q* is a significant advantage, as it enhances Q-ROFS's capacity for analyzing ambiguous information. Compared to crisp sets, fuzzy sets, and intuitionistic fuzzy sets, Q-ROFS is more robust and comprehensive. With the inclusion of *q*, the range of uncertain information captured by Q-ROFS is notably extensive and flexible, making it a powerful tool for addressing uncertainty in diverse applications. To assess the robustness of the parameter *q* in the Q-ROFS-FMEA method, a sensitivity analysis is conducted, with the results presented in Table 12.

Table 12: Sensitivity analysis of Q-ROFS-FMEA method with different values of *q*

q	**Ranking**
1	$FM_1 \succ FM_3 \succ FM_6 \succ FM_4 \succ FM_2 \succ FM_7 \succ FM_5 \succ FM_8$
2	$FM_1 \succ FM_3 \succ FM_6 \succ FM_4 \succ FM_2 \succ FM_5 \succ FM_7 \succ FM_8$
3	$FM_1 \succ FM_3 \succ FM_6 \succ FM_4 \succ FM_2 \succ FM_5 \succ FM_7 \succ FM_8$
4	$FM_1 \succ FM_3 \succ FM_6 \succ FM_4 \succ FM_2 \succ FM_5 \succ FM_7 \succ FM_8$
5	$FM_1 \succ FM_3 \succ FM_6 \succ FM_4 \succ FM_2 \succ FM_5 \succ FM_7 \succ FM_8$
6	$FM_1 \succ FM_3 \succ FM_6 \succ FM_4 \succ FM_2 \succ FM_5 \succ FM_7 \succ FM_8$
7	$FM_1 \succ FM_3 \succ FM_6 \succ FM_4 \succ FM_2 \succ FM_5 \succ FM_7 \succ FM_8$
8	$FM_1 \succ FM_3 \succ FM_6 \succ FM_4 \succ FM_2 \succ FM_5 \succ FM_7 \succ FM_8$
9	$FM_1 \succ FM_3 \succ FM_6 \succ FM_4 \succ FM_2 \succ FM_5 \succ FM_7 \succ FM_8$
10	$FM_1 \succ FM_3 \succ FM_6 \succ FM_4 \succ FM_2 \succ FM_5 \succ FM_7 \succ FM_8$

Based on Table 12, the sensitivity analysis results indicate that QROFS-FMEA is a robust method, as variations in the parameter *q* do not affect the overall ranking outcomes, except when *q=1*. At this specific value, a minor shift occurs between the sixth and seventh ranks compared to the other tested *q* values.

It is recalled that the objective of this paper is to identify the most critical failure modes of the risk factor COVID-19 using the proposed Q-ROFSs-FMEA. It is unveiled that FM_1 (older age) is the highest risk among the other failure modes. The relative risks of all factors are obtained as

$FM_1 \succ FM_3 \succ FM_6 \succ FM_4 \succ FM_5 \succ FM_7 \succ FM_8 \succ FM_2$

where the lowest failure mode in combating with the COVID-19 disease is FM_2 (gender). Therefore, this study suggests that the factor of 'gender' is not the main risk in estimating the likelihood of COVID-19 diseases. However, the failure mode FM_1 'older age' should be given the highest priority for risk mitigation of COVID-19. It is also good to mention here that the top two worst failure modes of the risk factor of COVID-19 are FM_1 and FM_3. This research sees the 'older age' and 'individual medical condition' failure modes are the most at-risk groups compared to other failure modes. This result is in line with the findings of Rod et al. [44], who found that the two main failure modes for COVID-19 disease are age and comorbidities.

6 Conclusion

Since 2019, the world has grappled with the profound impact of COVID-19. Numerous efforts have been undertaken to prevent its spread, yet questions remain as to whether these measures are truly sufficient to minimize the risk of infection. Moreover, many of the failure modes remain inconclusive and vague. Therefore, this research is conducted to identify the most critical failure modes of risk factors COVID-19. To meet this objective, the risk evaluation model, Q-ROFS-FMEA is proposed. The input data was elicited from a group of experts in public health who have been active in treating COVID-19 patients. Data were computed using the proposed Q-ROFS-FMEA where weights of experts and aggregation operators are the new features in the proposed method. This research indicates that the failure mode 'older age' is identified as the most-at-risk group. The result also shows that the failure mode 'gender' is the weakest risk factor. To validate these findings, a comparative analysis is presented where the results obtained from Q-ROFS-FMEA is compared to the results of the conventional FMEA, TFN-FMEA and IFS-FMEA. The comparative analysis demonstrates that the proposed Q-ROFS-FMEA method is similar to the IFS-FMEA; however, it yields different rankings when compared to the TFN-FMEA and FMEA methods. Notably, the top two highest risk factors for COVID-19 identified across all four methods are consistent: older age and individual medical conditions. This study provides an essential contribution to the medical field to mitigate the spread of the COVID-19 disease. However, the findings need further investigation as there are several limitations surrounded this study. The first limitation is on the data input. Since the data was collected from a group of experts, additional validations on the expert selection and data triangulation are required. Second limitation is on the ranking results where the results are obtained using the proposed works. Future research could benefit from incorporating insights from other studies, such as Gams and Kolenik [45], who highlighted exponential technological progress and its role in addressing human challenges, and Janco et al. [46], who investigated key cultural, developmental, and travel-related factors in pandemic spread. Expanding the methodological scope by utilizing alternative risk evaluation models such as the Risk Expected Value (REV) method, Data Envelopment Analysis, Monte Carlo Risk Analysis, and Fuzzy Bayesian Network could further enrich the understanding of COVID-19 risk factors and refine predictive accuracy.

Funding

This work was not funded

Acknowledgments

The authors would like to thank the research student N. S. Abd Samad for conducting linguistic data collection and part of computational analysis.

Conflicts of interest

The authors declare no conflict of interest.

Availability of data and materials

The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.

Ethics approval and consent to participate

Not applicable

Consent for publication

Not applicable

References

[1] Ganin, A. A, Quach, P., Panwar, M., Collier, Z. A., Keisler, J. M., Marchese, D., and Linkov, I. (2020). Multicriteria decision framework for cybersecurity risk assessment and management. *Risk Analysis*, 40(1), 183-199. https://doi.org/10.1111/risa.12891.

[2] Akbari M., Khazaee P., Sabetghadam I., and Karimifard, P. (2013). Failure modes and effects analysis (FMEA) for power transformers, *28th International Power System Conference (PSC)*, pp. 1-7.

[3] Chiozza, M. L. and Ponzetti, C. (2009). FMEA: a model for reducing medical errors, *Clinica chimica acta*, 404(1) 75-78. https://doi.org/10.1016/j.cca.2009.03.015.

[4] Mutlu, N. G. and Altuntas, S. (2019). Risk analysis for occupational safety and health in the textile industry: Integration of FMEA, FTA, and BIFPET methods, *International Journal of Industrial Ergonomics*, 72 222-240. https://doi.org/10.1016/j.ergon.2019.05.013.

[5] Tay, K. M. and Lim, C. P. (2008). On the use of fuzzy inference techniques in assessment models: part II: industrial applications, *Fuzzy Optimization and Decision Making*, 7(3) 283–302. https://doi.org/10.1007/s10700-008-9037-y.

[6] Rausand, M. and Hoyland, A. (2003). *System reliability theory: models, statistical methods, and applications* John Wiley & Sons.

[7] Ouyang, L. Zheng, W., Zhu, Y. and Zhou, X. (2020). An interval probability-based FMEA model for risk assessment: A real-world case, *Quality and Reliability Engineering International*, 36(1) 125-143. https://doi.org/10.1002/qre.2563.

[8] Wu, Z., Liu, W. and Nie, W. (2021). Literature review and prospect of the development and application of FMEA in manufacturing industry. *The International Journal of Advanced Manufacturing Technology*, 112(5) 1409-1436. https://doi.org/10.1007/s00170-020-06425-0.

[9] Yazdi, M. Daneshvar, S. and Setareh, H. (2017). An extension to fuzzy developed failure mode and effects analysis (FDFMEA) application for aircraft landing system. *Safety science* 98 113-123. https://doi.org/10.1016/j.ssci.2017.06.009.

[10] Subriadi, A.P. and Najwa, N. F. (2020). The consistency analysis of failure mode and effect analysis (FMEA) in information technology risk assessment, *Heliyon* 6(1) e03161.

[11] Liu, H. C., You, J. X., Lu, C. and Shan, M. M. (2014). Application of interval 2-tuple linguistic MULTIMOORA method for health-care waste treatment technology evaluation and selection, *Waste Management* 34(11) 2355-2364. https://doi.org/10.1016/j.wasman.2014.07.016.

[12] Chalidyanto, D. and Kurniasari, W. E. (2020). Application of Failure Mode and Effect Analysis (FMEA) report of medication processing a private hospital. *EurAsian Journal of BioSciences*, 14(2) 3257-3261.

[13] Lee, J. C., Daraba, A., Voidarou, C., Rozos, G. Enshasy, H. A. E. and Varzakas, T. (2021). Implementation of food safety management systems along with other management tools (HAZOP, FMEA, Ishikawa, Pareto). The case study of listeria monocytogenes and correlation with microbiological criteria, *Foods* 10(9) 2169. https://doi.org/10.3390/foods10092169.

[14] Sharifi, F., Vahdatzad, M. A. Barghi, B. and Azadeh-Fard N. (2022). Identifying and ranking risks using combined FMEA-TOPSIS method for new product development in the dairy industry and offering mitigation strategies: case study of Ramak Company, *International Journal of System Assurance Engineering and Management*, 13 2790–2807. https://doi.org/10.1007/s13198-022-01672-8.

[15] Başhan, V. Demirel, H. and Gul, M. (2020). An FMEA-based TOPSIS approach under single valued neutrosophic sets for maritime risk evaluation: the

case of ship navigation safety, *Soft Computing*, 24(24) 18749-18764.

[16] Şenel, M. Şenel, B. and Havle, C. A. (2018). Risk analysis of ports in Maritime Industry in Turkey using FMEA based intuitionistic Fuzzy TOPSIS Approach. *In ITM Web of Conferences, EDP Sciences*, 22, p. 01018. https://doi.org/10.1051/itmconf/20182201018.

[17] Liu, H. C., Liu, L. and Liu, N. (2013). Risk evaluation approaches in failure mode and effects analysis: A literature review. *Expert Systems with Applications* 40(2) 828-838. https://doi.org/10.1016/j.eswa.2012.08.010.

[18] Balaraju, J. Raj, M. G. and Murthy, C. S. (2019). Fuzzy-FMEA risk evaluation approach for LHD machine–A case study, *Journal of Sustainable Mining* 18(4) 257-268. https://doi.org/10.1016/j.jsm.2019.08.002.

[19] Wang, L. Hu, Y. P. Liu, H. C. and Shi, H. (2019). A linguistic risk prioritization approach for failure mode and effects analysis: A case study of medical product development. *Quality and Reliability Engineering International* 35(6) 1735-1752. https://doi.org/10.1002/qre.2472.

[20] Huang, J. Liu, H. C. Duan, C. Y. and Song, M. S. (2022). An improved reliability model for FMEA using probabilistic linguistic term sets and TODIM method, *Annals of Operations Research*, 312(1) 235-258. https://doi.org/10.1007/s10479-019-03447-0.

[21] Yalcinkaya, I. and Cebi, S. (2022). Using Fuzzy Set Based Model for Pharmaceutical Supply Chain Risks Assessment. *In International Conference on Intelligent and Fuzzy Systems*. Springer, Cham, pp. 252-260. https://doi.org/10.1007/978-3-031-09173-5_32.

[22] Huang, G. Xiao, L. Pedrycz, W. Zhang, G. and Martinez, L. (2022). Failure Mode and Effect Analysis Using T-Spherical Fuzzy Maximizing Deviation and Combined Comparison Solution Methods. *IEEE Transactions on Reliability* 1-22. 10.1109/TR.2022.3194057.

[23] Nie, R. X. Tian, Z. P. Wang, X. K. Wang, J. Q. and Wang, T. L. (2018). Risk evaluation by FMEA of supercritical water gasification system using multi-granular linguistic distribution assessment. *Knowledge-Based Systems,* 162 185-201. https://doi.org/10.1016/j.knosys.2018.05.030.

[24] Daneshvar, S. Yazdi, M. and Adesina, K. A. (2020). Fuzzy smart failure modes and effects analysis to improve safety performance of system: Case study of an aircraft landing system. *Quality and Reliability Engineering International,* 36(3), 890-909. https://doi.org/10.1002/qre.2607.

[25] Wang, Q., Cao, Y. X. and Zhang, H. Y. (2017). Multi-criteria decision-making method based on distance measure and Choquet integral for linguistic Z-numbers. *Cognitive Computation*, 9(6) 827-842. https://doi.org/10.1007/s12559-017-9493-1.

[26] Wang, Z. L., You, J. X., Liu, H. C., and Wu, S. M. (2017). Failure mode and effect analysis using soft set theory and COPRAS method. *International Journal of Computational Intelligence Systems,* 10(1),1002-1015. https://doi.org/10.2991/ijcis.2017.10.1.67.

[27] Testik, O. M. and Unlu, E. T. (2022). Fuzzy FMEA in risk assessment for test and calibration laboratories, *Quality and Reliability Engineering International*, *39*(2), 575-589 https://doi.org/10.1002/qre.3198.

[28] Bhuvanesh Kumar, M. and Parameshwaran, R. (2018). Fuzzy integrated QFD, FMEA framework for the selection of lean tools in a manufacturing organisation. *Production Planning and Control* 29(5) 403-417. https://doi.org/10.1080/09537287.2018.1434253

[29] Ouyang, L., Zhu, Y., Zheng, W., and Yan, L. (2021). An information fusion FMEA method to assess the risk of healthcare waste. *Journal of Management Science and Engineering,* 6(1) 111-124. https://doi.org/10.1016/j.jmse.2021.01.001

[30] Zadeh, A. (1965). Fuzzy sets. *Information and control*, 8(3) 338-353.

[31] Atanassov, K. (1986). Intuitionistic fuzzy sets, *Fuzzy set and systems,* 20, 87-96.

[32] Yager, R. (2017). Generalized orthopair fuzzy sets, *IEEE Trans Fuzzy Syst*, 25(5) 1222–1230. 10.1109/TFUZZ.2016.2604005

[33] Bhuiyan, A., Dinçer, H., Yüksel, S., Mikhaylov, A. Danish, M. S. S., Pinter, G. and Stepanova, D. (2022). Economic indicators and bioenergy supply in developed economies: QROF-DEMATEL and random forest models. *Energy Reports,* 8 561-570. https://doi.org/10.1016/j.egyr.2021.11.278.

[34] Singh, S. and Ganie, A. H. (2022). Some novel q-rung orthopair fuzzy correlation coefficients based on the statistical viewpoint with their applications. *Journal of Ambient Intelligence and Humanized Computing*, 13(4) 2227-2252. https://doi.org/10.1007/s12652-021-02983-7.

[35] Liu, P. and Wang, P. (2018). Some q-rung orthopair fuzzy aggregation operators and their applications to multiple-attribute decision making. *International Journal of Intelligent Systems*, 33(2) 259-280. https://doi.org/10.1002/int.21927.

[36] Peng, X. and Luo, Z. (2021). A review of q-rung orthopair fuzzy information: bibliometrics and future directions. *Artificial Intelligence Review*, 54(5) 3361-3430. https://doi.org/10.1007/s10462-020-09926-2.

[37] Jin, C., Ran, Y. and Zhang, G. (2021). Interval-valued q-rung orthopair fuzzy FMEA application to improve risk evaluation process of tool changing manipulator. *Applied Soft Computing,* 104 107192. https://doi.org/10.1016/j.asoc.2021.107192.

[38] Rashedi, J., Mahdavi Poor, B., Asgharzadeh, V. Pourostadi, M., Samadi Kafil, H., Vegari, A. and Asgharzadeh, M. (2020). Risk factors for COVID-19. *Infez Med,* 28(4) 469-474.

[39] Jordan, R. E. Adab, P. and Cheng, K. (2020). Covid-19: risk factors for severe disease and death. *Bmj* 368. https://doi.org/10.1136/bmj.m1198.

[40] Gebhard, C., Regitz-Zagrosek, V., Neuhauser, H. K., Morgan, R. and Klein, S. L. (2020). Impact of sex and gender on COVID-19 outcomes in Europe. *Biology of Sex Differences* 11(1) 1-13. https://doi.org/10.1186/s13293-020-00304-9.

[41] Ambrosino, E., Barbagelata, G., Corbi, T., Ciarambino, C., Politi, A. and Moretti, M. (2020). Gender differences in treatment of Coronavirus Disease-2019. *Monaldi Archives for Chest Disease* 90(4) 646-656. https://doi.org/10.4081/monaldi.2020.1508.

[42] de Sousa Lima, M. E. Barros, L. C. M., and Aragão, G. F. (2020). Could autism spectrum disorders be a risk factor for COVID-19?. *Medical Hypotheses* 144 109899. https://doi.org/10.1016/j.mehy.2020.109899

[43] Leso, V., Fontana, L. and Iavicoli, I. (2021). Susceptibility to coronavirus (COVID-19) in occupational settings: The complex interplay between individual and workplace factors. *International Journal of Environmental Research and Public Health,* 18(3) 1030. https://doi.org/10.3390/ijerph18031030.

[44] Rod, E. Oviedo-Trespalacios, O. and Cortes-Ramirez, J. (2020). A brief-review of the risk factors for Covid-19 severity, *Revista de saude publica,* 54 1-11.https://doi.org/10.11606/s1518-8787.2020054002481.

[45] Gams, M., and Kolenik, T. (2021). Relations between electronics, artificial intelligence and information society through information society rules. *Electronics*, *10*(4), 514. https://doi.org/10.3390/electronics10040514.

[46] Janko, V., Slapničar, G., Dovgan, E., Reščič, N., Kolenik, T., Gjoreski, M., ... and Luštrek, M. (2021). Machine learning for analyzing non-countermeasure factors affecting early spread of COVID-19. *International Journal of Environmental Research and Public Health*, *18*(13), 6750. https://doi.org/10.3390/ijerph18136750.

Real-Time Smart Healthcare System Based on Edge-Internet of Things and Deep Neural Networks for Heart Disease Prediction

Messaoud Hameurlaine[1], Abdelouhab Moussaoui[2], Mustapha Bensalah[3]
E-mail: hameurelaine.messaoud@univ-tissemsilt.dz, abdelouahab.moussaoui@univ-setif.dz, bensmusta@gmail.com
[1]LEIMA Laboratory, Center Universitaire Ahmed Ben Yahia El Wancharissi de Tissemsilt, Tissemsilt, Algeria
[2]Computer Science Department , Faculty of Sciences, Ferhat Abbas University - Sétif 1, Algeria
[3]Signal Processing Laboratory, Military Polytechnic School, Algiers, Algeria

Keywords: Smart healthcare system, edge computing, internet of things, deep learning, deep neural network

Received: August 7, 2024

With technological advancements, smart health monitoring systems have become increasingly vital and popular. The rise of smart homes, appliances, and medical systems, along with the pivotal role of the Internet of Things (IoT), is significantly enhancing healthcare services by improving data processing and predictive capabilities. IoT not only aids in predicting heart disease but also supports emergency responses. However, traditional data transfer methods are inefficient in terms of time and energy, resulting in high latency and consumption. Edge computing, alongside deep learning methods, provides effective solutions with superior performance. This paper introduces a Real-Time Smart Healthcare System utilizing Edge-Internet of Things and Deep Learning. The primary objective of this system is to monitor patient health changes, predict heart disease, and automate medication administration in real time. The study presents a DNN-based prediction model that leverages edge computing and IoT. This model processes health data from IoT devices, while edge devices deliver timely health predictions to doctors and patients via edge and cloud servers. The proposed system is evaluated on performance parameters, demonstrating superior results compared to other methods. By integrating edge computing, IoT, and deep learning, this system enables efficient real-time health monitoring and prediction, benefiting both healthcare professionals and patients. It demonstrates exceptional performance with an accuracy of 96.15%, precision of 92.86%, recall (sensitivity) of 97.50%, and an F1-score of 94.87%.

Povzetek: Razvili smo sistem za pametno zdravstveno oskrbo, ki temelji na tehnologijah Edge-IoT in globokem učenju in napoveduje srčne bolezni v realnem času. Sistem uporablja globoko nevronsko mrežo za obdelavo podatkov iz IoT naprav, kar omogoča pravočasne zdravstvene napovedi.

1 Introduction

As the prevalence of heart disease patients continues to grow, the strain on current healthcare systems is on the rise. Addressing this challenge requires the increased involvement of specialists. However, it is crucial to act swiftly when dealing with heart patients, especially in emergencies. An efficient Smart Healthcare Surveillance System (SHSS) can effectively tackle these issues by offering a range of services, including monitoring, remote treatments, autonomous actions, and real-time situation management[1], [2]. Such a system can provide timely and effective support to heart disease patients, ensuring rapid and efficient responses in critical situations. Therefore, the demand for modern and future healthcare services requires computational power and rapid response times. In recent years, advances in information technology and artificial intelligence (AI) have played a transformative role across society, highlighting exponential progress in computational power, data storage, and electronic transmission capabilities [3], which are now critical to modern healthcare systems. The proliferation of AI-driven systems has unlocked the potential for unprecedented levels of automation and real-time data processing, especially through edge computing.

Nonetheless, mobile cloud computing (MCC), which preceded edge computing, encountered similar obstacles, such as high costs of data transmission, delayed response times and restricted network reach. Several investigations have contrasted cloud-centric and edge-centric computing, concluding that solely edge computing aligns with the demands for reduced latency, mobility, and energy conservation [4], [5]. In the healthcare sector, edge computing outperforms traditional cloud computing. Healthcare professionals can deliver remote medical assistance to individuals with chronic conditions through the utilization of wearable devices and ambient sensors to monitor vital signs. This is achievable thanks to the oversight and flexibility afforded by edge computing systems. Furthermore, doctors can detect patient risks based on sensor data, regardless of their location. For superior care delivery, it's essential that edge devices and nodes promptly perform data operations [6]. Real-time healthcare applications are also crucial for immediate risk detection and intervention. A Real-time Semantic Healthcare System is developed in[7] to identify Visual Risks for Elders and Children in surveillance videos, enabling swift responses to prevent accidents.

We propose a healthcare monitoring system that relies on Edge Internet of Things and deep learning technologies. The primary objectives of this system are as follows:

- An envisioned design for a Heart Disease Smart Healthcare Surveillance System integrates IoT-Edge-Cloud technology. This system harnesses the power of Edge devices to efficiently handle real-time situations and optimize performance by minimizing computation and transmission overhead.
- Data gathered from IoT sensors for predicting the risk of heart disease undergoes a sequence of preprocessing procedures and analysis at the Edge layer, facilitated by FPGA programmable processors. This process ensures that the data is prepared and analyzed efficiently and effectively within the system edge layer.

The subsequent segments of the paper are presented as follows: a comprehensive overview of existing research and studies related to the topic is presented in section 2. Section 3 offer detailed elaboration on our proposed architecture. The analysis of the results is presented in Section 4. Finally, Section 5 summarizes the findings and conclusions of the research.

2 Related work

In this section, we present a compilation of related works. One such research discusses an IoT-based healthcare system capable of monitoring and tracking patients, staff, and biomedical devices, while also handling emergency situations effectively[8]. Additionally, a framework is introduced that includes a real-time alert generation system to guarantee swift responses. Furthermore, there is the introduction of an IoT-cloud-based framework for healthcare applications, with a specific focus on real-time prediction of health vulnerabilities during workout sessions[9]. A healthcare system based on IoT is created to meet the demand for intelligent health monitoring. The framework introduces a combination of Fully Homomorphic Encryption (FHE) and machine learning [10]. This framework enables encrypted analysis of biosignal data, including aggregation, real-time monitoring, and abnormality detection.

In the healthcare industry, predictive analytics covers a wide range of techniques, ranging from conventional linear models to sophisticated machine learning techniques [11]. Among these techniques, deep learning (DL), a subset of ML, has proven to be highly reliable and robust. DL excels in automatically handling and learning from vast and complex healthcare datasets, providing valuable perspectives and efficient solutions to complex issues. Its application in diverse medical domains has consistently shown superior performance compared to classic designs. More precisely, the recurrent neural network (RNN) has demonstrated its effectiveness in managing prolonged dependencies in input data. RNN has become prominent in analyzing temporal events, particularly in applications that involve time-sequential data [12].

Several studies have focused on the diagnosis and predictive modeling of heart disease. Botros et al.[13] introduced two models for detecting heart failure from electrocardiogram signals: a Convolutional Neural Network and an enhanced version that includes an SVM layer, achieving over 99% accuracy, sensitivity, and specificity. This framework aids professionals and allows real-time processing with mobile equipments. The authors in [14] examined heart disease prediction using six ML models, as logistic regression and random forest. Logistic regression reached 90.16% accuracy on the Cleveland data, and AdaBoost achieved 90% on the IEEE Dataport data. The accuracy of the soft voting group classifiers was improved to 93.44% and 95%, respectively. Nancy et al.[2] utilized bidirectional LSTM for heart disease prediction, achieving an accuracy of 0.98 and outperforming existing methods. This highlights the importance of timely disease prediction for early intervention. Authors in [15] created a Smart Cardiovascular Disease Diagnostic Framework using Internet of Things devices. Their ConvNet and ConvNet-LSTM design successfully obtained a 98% accuracy rate in identifying atrial fibrillation using cloud architecture and DL. The combination of IoT devices and cloud computing with deep learning models offers transformative possibilities for healthcare, particularly in remote health monitoring and precise disease diagnosis. Table 1 present a summary of some recent approaches along with their respective performance and advantages.

Although the existing literature demonstrates promising approaches in heart disease prediction and healthcare monitoring, several limitations remain. Many of the techniques rely on powerful computational resources, such as deep learning models (CNN, LSTM), which can be costly and require large datasets for training. Additionally, real-time prediction remains a challenge, particularly in resource-constrained environments such as wearable IoT devices and edge computing systems. While IoT-based healthcare systems are increasingly utilized, their effectiveness often relies on centralized cloud processing, which introduces latency and scalability issues.

The proposed system addresses these challenges by leveraging edge computing and IoT integration, enabling real-time prediction and monitoring at the point of care, with significantly reduced latency. Our approach also integrates deep learning models such as DNN for improved accuracy, while operating efficiently on edge devices with limited resources, making it a valuable advancement over existing state-of-the-art (SOTA) methods in healthcare prediction systems.

Table 1: Summary of various approaches along with their respective performance and advantages.

Approach	Description	Advantages	Disadvantages	Reference
Meta classification technique	Uses multiple classifiers to improve prediction by combining their outputs	Combines strengths of different models for improved accuracy	Can be computationally expensive due to ensemble complexity	Latha et Jeeva (2019) [16]
Hybrid random forest with linear model	Combines random forest and linear models to predict heart disease	Balances complexity and interpretability	May not fully capture nonlinear relationships in data	Mohan et al. (2019) [17]
Statistical model and deep neural network	Combines traditional statistical methods with deep neural networks for heart disease prediction	Leverages both classic and modern techniques for robust results	Requires large datasets for deep learning to be effective	Moreno-Ibarra et al. (2019) [18]
Bi-directional LSTM (C-BiLSTM)	Uses BiLSTM to handle sequential data for heart disease prediction	Captures temporal dependencies in patient data, improving prediction accuracy	Computationally intensive, requires significant training data	Dileep et al. (2023) [19]
Hyperparameter tuning and cross-validation with ML	Employs hyperparameter optimization to enhance performance of machine learning models	Improves generalization ability of models	Can lead to overfitting if not carefully tuned	Ahmed et al. (2020) [20]
Random Forest	Utilizes Random Forest for heart disease classification	Easy to interpret, performs well with unstructured data	Can be less effective with highly imbalanced datasets	Dhanamjayulu et al. (2022) [21]
Optimized ensemble fuzzy ranking (OEFR)	Uses an ensemble of fuzzy ranking algorithms for heart disease prediction	Optimizes predictions, reduces error	May not handle very large datasets well	Managala et al. (2023) [22]
RNN (Recurrent Neural Network)	Uses RNNs for heart disease prediction from temporal data like ECG signals	Suitable for sequential data, handles time-series data well	Struggles with long-term dependencies in data	Almujally et al. (2023) [23]
IoT-Cloud-Based Healthcare System	Implements an IoT-based framework for real-time monitoring of heart disease during workouts	Integrates multiple data sources (physiological, behavioral) for holistic health tracking	Security concerns related to data transmission and privacy	Nancy et al.(2022) [2]
Deep Forest Cascade Technique	Uses a cascade of deep forest models to predict heart disease	High accuracy in prediction, interpretable output	Requires fine-tuning of cascade layers	Askar (2023) [24]
XGBoost	Uses an optimized gradient boosting model for classification tasks	Very high prediction accuracy, great for structured data	Sensitive to noisy data, requires proper feature engineering	Gracious et al. (2024) [25]
Optimized Random Forest with SMOTE	Combines Random Forest with SMOTE to balance class distributions for heart disease prediction	Improves model robustness and generalization	Computationally expensive and may require a lot of time to optimize	ishaq et al. (2023) [26]

3 Materials and methods

3.1 Architecture of the proposed smart healthcare system

IoT technology plays a pivotal role in various real-time applications, enabling seamless interaction between objects and individuals. However, the considerable volume of medical information produced by such equipment is a significant obstacle for the system, particularly in data storage, processing, and management. To address this challenge, we support the adoption of an intelligent system for heart disease diagnostic, employing edge IoT technology. The system, illustrated in Figure 1, aims to overcome the challenges posed by the massive data generated in healthcare settings. The architecture comprises 3 layers: the Cloud Layer, the Edge Layer, and the Data Generation Layer. The global architecture of the system is built upon the framework described in [27], but we have implemented several modifications. These include the integration of an intelligent sensor capable of automatically generating various physiological parameters continuously. The synergy between the National Instruments myRIO processor and a Wi-Fi module facilitates wireless data transmission to the cloud server [28]. Additionally, the system integrates real-time online monitoring of health status [6]. Moreover, the cloud component is employed for dispatching alert messages to patients, as documented in[21].

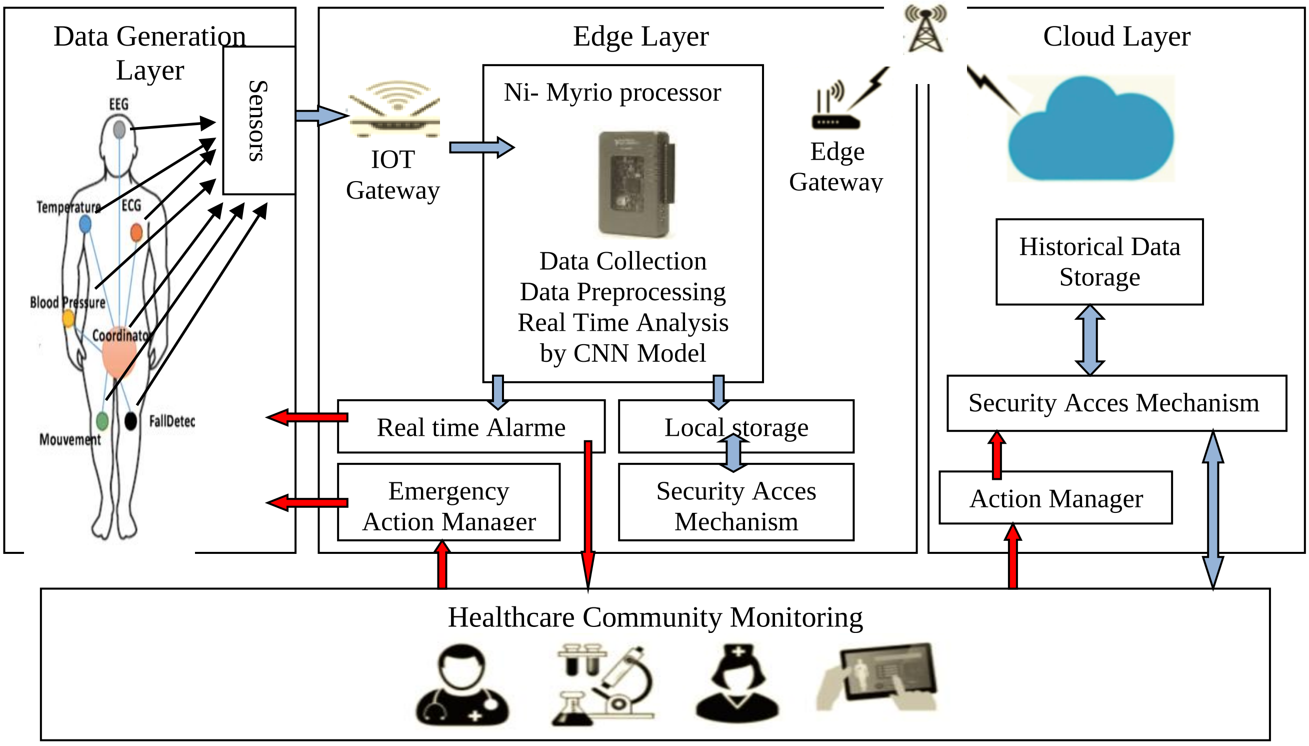


Figure 1: Proposed architecture of edge based system.

The proposed architecture enhances existing designs by incorporating specific adaptations tailored for real-time, resource-efficient processing on edge devices. Unlike traditional systems that rely solely on cloud computing for high-level data processing, our design emphasizes advanced local processing through edge computing, significantly reducing latency. The integration of intelligent sensors at the data generation layer provides continuous physiological parameter tracking, which is processed by a custom DNN model at the edge. This DNN model has been optimized for the constrained computational environment of edge devices, enabling it to operate effectively with lower energy consumption and processing power without compromising diagnostic accuracy.

3.2 Data generation layer

The function of this layer is to acquire physiological health parameters using a variety of Internet-connected or wearable devices. Typically, these devices have limited storage and computing capabilities. Local processing techniques can be implemented to address issues such as data redundancy, power consumption, and network transmission load. For connectivity, the data collection layer devices can establish a connection with smart phones or other mobile smart terminals via Bluetooth technology. To facilitate high-speed transmission of measured signals, all smart sensors are linked to an NI myRIO platform through a Wi-Fi module capable in swift data transfer. The NI myRIO platform utilizes a low-power Xilinx FPGA programmable processor, making it highly suitable for efficient signal transmission and reception tasks. To interact with the collected data, the NI-myRIO module communicates with an application that provides the capability to use a web-based interface for data visualization and analysis.

3.3 Edge layer

Primary role of this layer is to execute computations for early detection and take necessary actions based on the acquired physiological data. The Edge layer primarily consists of smart phones and other intelligent mobile terminals. It serves as network layer devices, enabling data communication functions of the medical IoT gateway. We use here NI myRIO. Moreover, they host application layer software, including preprocessing algorithms for filtering and consolidating data, thereby enhancing real-time analysis speed. The process of analysis is streamlined by deep learning models, offering reliability and precision in the detection of potential health concerns. Once the analysis is complete, the decision-making module, with the assistance of healthcare specialists, determines whether immediate action is required. If necessary, an alarm is raised to alert the healthcare community and an autonomous system capable of addressing emergencies in real-time. In instances where no alarm is triggered, both the data and analysis results are archived in the Edge layer before being transmitted to the Cloud layer. The access procedure incorporates appropriate access control measures for the healthcare community, granting authorized individuals the ability to retrieve and interact with the data as required.

3.4 Cloud layer

This layer focuses on data storage, latency-tolerant analysis, and access control for the healthcare community. It acts as a repository for collected data and allows for further analysis that can tolerate certain delays. The access mechanism ensures proper control over data access by authorized individuals. The analysis results are shared with the healthcare community, enabling timely actions based on the derived insights. The healthcare community can utilize cloud-based solutions for monitoring patients as well.

3.5 Security and confidentiality

Our smart healthcare system, which is built on the Internet of Things (IoT), incorporates a security solution provided by [23]. Our system complies with security standards like HIPAA and GDPR by using encryption and secure authentication methods, ensuring that only authorized users and devices can access sensitive health data while protecting user privacy. This solution utilizes Zigbee and Firebase IoT authentication. During the transmission of health information, a 128-bit encryption is utilized to secure the JSON file as a token. Firebase cloud functions authenticate the officer's device token by generating a custom token with precise credentials and claims. Both the 128-bit device token and the Firebase custom token serve as authentication mechanisms for real-time data exchanges. After the user's identity is verified, the authorization process makes use of Firebase's universal security standards. The security system consists of three steps:

- The equipment identifier authenticates the request as being from permitted equipment, but it does not provide any important details for identifying the owner.
- A customized certificate includes user identification but does not have profile details and cannot be automatically recognized by Firebase servers due to potential revocation or key rotation.
- The signInWithCustomToken API verifies the claims of the custom token, and then the backend produces a Firebase Identity token. This token, which includes the user's characteristics, serves as indisputable evidence of authorization and remains valid for duration of one hour.

3.6 Proposed deep learning model

In the past few years, there has been significant research and extensive implementation of deep learning algorithms, aimed at extracting valuable information from different varieties of data. Various deep learning architectures have been implemented to accommodate the diverse characteristics of input data, encompassing conventional neural networks, deep neural networks, and recurrent neural networks.

In this particular case, a deep neural network model has been adopted and modified to predict heart disease. The specific architecture of the DNN is illustrated in Figure 2. The described sequential model architecture is a dense neural network (DNN) comprising multiple fully connected layers with regularization mechanisms to enhance generalization and prevent over fitting. The model begins with a dense layer of 128 neurons, using the ReLU (Rectified Linear Unit) activation function. The weights in this layer are initialized with a normal distribution, and L2 regularization with a coefficient of 0.001 is applied to mitigate over fitting by penalizing large weights. This layer is followed by a Dropout layer that randomly drops 20% of the neurons during training, providing further regularization. This structure of dense and dropout layers is repeated with 64, 32, and 16 neurons in subsequent dense layers, each maintaining the same initialization and regularization techniques. The model continues with another Dropout layer after each dense layer to ensure regularization is consistently applied throughout the network. Following these, a dense layer with 8 neurons using the softmax activation function is included. The softmax activation function is typically used for multi-class classification and outputs a probability distribution classes. Another Dropout layer follows this, and finally, the model includes a dense layer with 2 neurons and a sigmoid activation function. The sigmoid activation function is often used for binary classification tasks, outputting probabilities for two classes.

The model is trained with the Adam optimizer, employing a learning rate of 0.001. The chosen loss function is categorical crossentropy, so it is well-suited for issues involving multi-class categorization. The categorical crossentropy loss function measures the dissimilarity between the true labels and the predicted probabilities, penalizing incorrect classifications more severely. The formula for categorical crossentropy loss is:

$$Loss = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) \qquad (1)$$

where N is the number of classes, y_i is the true label for class i , and $\hat{y}_i$ is the predicted probability for class i.

L2 regularization, often referred to as Ridge regularization, is a method employed in machine learning models to mitigate over fitting. It achieves this by incorporating a penalty term into the loss function. The "L2" refers to the L2 norm, which is the sum of the squared values of the weights. This penalty term discourages the model from fitting the noise in the training data by shrinking the weight values, thus promoting simpler models with smaller weights.

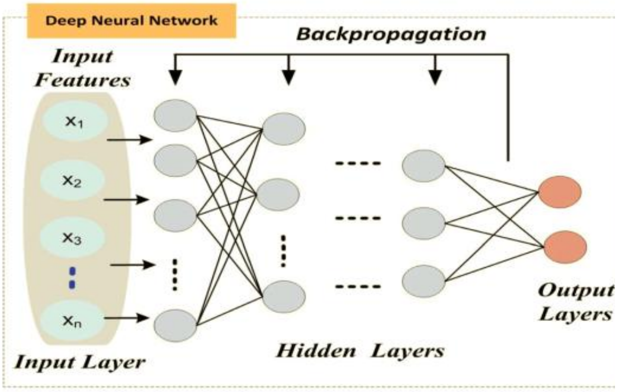


Figure 2: DNN model

Mathematically, L2 regularization adds the following term to the loss function:

$$P = \lambda \sum_i \omega_i^2 \qquad (2)$$

Where λ is a regularization parameter that controls the strength of the penalty, and ω_i are the weights of the model.

When the loss function with L2 regularization is minimized, the model not only tries to minimize the original loss but also tries to keep the weights small. This helps in preventing over fitting, as models with smaller weights are less likely to fit the noise in the training data and more likely to generalize better to new, unseen data.

The training phase entails the adjustment of the model to the test data with a validation split using the same test data over 60 epochs and a batch size of 15. The model's training was optimized by employing an early stopping criterion, where training is terminated if accuracy stabilizes without improvement over a set number of epochs, ensuring optimal model performance while avoiding unnecessary computation. Other hyperparameters were determined experimentally to achieve the most effective configuration. For instance, the batch size was set to 15 to maintain a balance between memory efficiency and gradient updates, which aids in faster convergence. The learning rate for the Adam optimizer was chosen as 0.001 after testing different rates and selecting the one that yielded the best training and validation accuracy. Moreover, each layer's dropout rate and the L2 regularization coefficient were tuned to prevent overfitting effectively. These hyperparameter values were fine-tuned by monitoring model performance across multiple experimental runs, adjusting values iteratively to achieve a reliable, accurate prediction model for heart disease diagnosis. Upon training completion, the model's performance is evaluated on the test data. The evaluation provides a loss and accuracy metric. Construction and training of the deep learning model took place on Edge layer or cloud server. The output layer of this model generates health assessment results, which are categorized into binary classes: 0 for health and 1 for illness.

3.7 Dataset

The Erbil Heart Disease Dataset [29], sourced from the Medical Help Centre, a specialized heart hospital in Erbil, Iraq, contains data on 333 patients, each with 21 attributes as detailed in Table 2. This publicly available dataset aims to facilitate the prediction of heart disease using information specific to the local population. The dataset's attributes are organized into five categories: demographic information, medical history, physical examinations and symptoms, medical laboratory tests, and diagnostic features. These attributes were carefully chosen based on expert medical advice to ensure their relevance and significance for heart disease prediction. By leveraging this dataset, researchers can develop and refine predictive models that are tailored to the unique characteristics of the patients from this region, ultimately enhancing the accuracy and effectiveness of heart disease diagnosis and treatment strategies. Figure 3 illustrates the distribution of the target variable for heart disease. It visually represents the proportion of patients with and without heart disease, aiding in understanding the dataset's balance and the prevalence of the condition.

Table 2: Data description

Attribute	**Description**
Age	Patients' ages, measured in years.
Sex	The patient's gender is indicated by a value of 1 for female and 0 for male.
Smoking	Indicate whether the patient is a smoker or not (0=No, 1=Yes)
Years	Duration of smoking for smokers
LDL	The patient's LDL-Cholesterol ratio
Chp	Types of chest pain are categorized as follows: 1= Typical angina, 2= Atypical angina, 3= Non-anginal pain, and 4= Asymptomatic.
Height	The patient's height, measured in centimeters.
Weight	The patients' weight, measured in kilograms.
FH	History of heart disease among family members
Active	Indicate the patient's level of activity (0=Inactive, 1=Active)
Lifestyle	Residence location: 1 = City, 2 = Town, 3 = Village
CI	Has the patient undergone any cardiac catheterization or any invasive procedures involving the heart? (0 indicates No, while 1 indicates Yes)
HR	Cardiac pulse ratio
DM	Presence of diabetes: 0 = No, 1 = Yes
Bpsys	Ratio of systolic blood pressure
Bpdias	Ratio of diastolic blood pressure
HTN	The patient's hypertension status: 0 for "No," 1 for "Yes."
IVSD	Echo parameter: Interventricular Septal Thickness during Diastole (IVSD), a measurement utilized in determining Left Ventricular Hypertrophy (LVH).
ECGpatt	The ECG test includes four categories: ST-Elevation (1), ST-Depression (2), T-Inversion (3), and Normal (4).
Qwave	Presence of the Q wave: 0 for "No," 1 for "Yes."
Target	The patient's heart disease status: 0 for "without heart disease," 1 for "with heart disease."

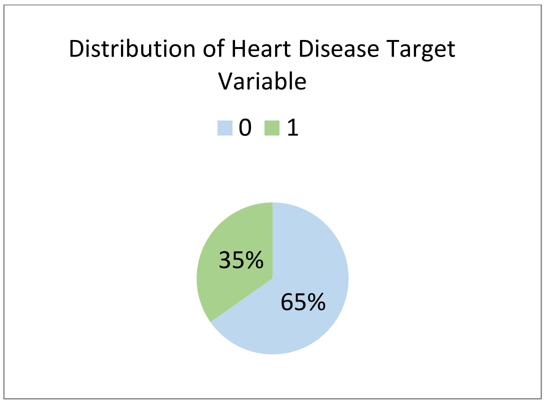


Figure 3: Distribution of target attribute

3.8 Data preprocessing

Data preprocessing plays a pivotal role in building a more precise machine learning model. This phase encompasses several tasks geared towards enhancing data quality, including the identification and management of missing values, the detection and elimination of outliers, and the selection of pertinent features. In addition to the preprocessing operations already performed on the Erbil data, we have implemented the following supplementary preprocessing steps to enhance their usability further.

3.8.1 Normalization

Normalization, particularly standardization, is a crucial preprocessing step in data analysis and machine learning. It adjusts the scale of data to ensure that each feature contributes equally, preventing those with larger scales from dominating results. This process involves centering the data around zero by subtracting the mean of each feature, effectively neutralizing biases introduced by different scales. Subsequently, the data is scaled by dividing it by the standard deviation of each feature, normalizing variance across features. This transformation, which results in each feature having a mean of zero and a standard deviation of one, is beneficial for algorithms sensitive to feature scales, such as linear regression and neural networks. Standardization accelerates gradient descent convergence and enhances model performance, ensuring equitable feature treatment and improving analysis or model training accuracy.

3.8.2 One-hot encoding

The operation of converting class labels into a categorical format, known as one-hot encoding, transforms numerical class labels into binary vectors. This process is essential in deep learning as it ensures that each class label is represented equally and without any implicit ordinal relationship. One-hot encoding helps neural networks to interpret the labels correctly, facilitating accurate computation of loss functions like categorical cross-entropy. This encoding method boosts the model's capacity to discern patterns from the data efficiently, thereby enhancing its performance in classification tasks.

3.8.3 Identification of missing values

This operation is performed to verify the absence of missing values in a dataset. By calculating the total percentage of missing data, we can confirm whether the dataset is complete. The process involves counting all missing entries across the dataset, ensuring that the dataset is fully intact and reliable for subsequent analysis or model training without requiring further imputation or cleaning steps.

4 Results and Discussion

4.1 Training and testing performance

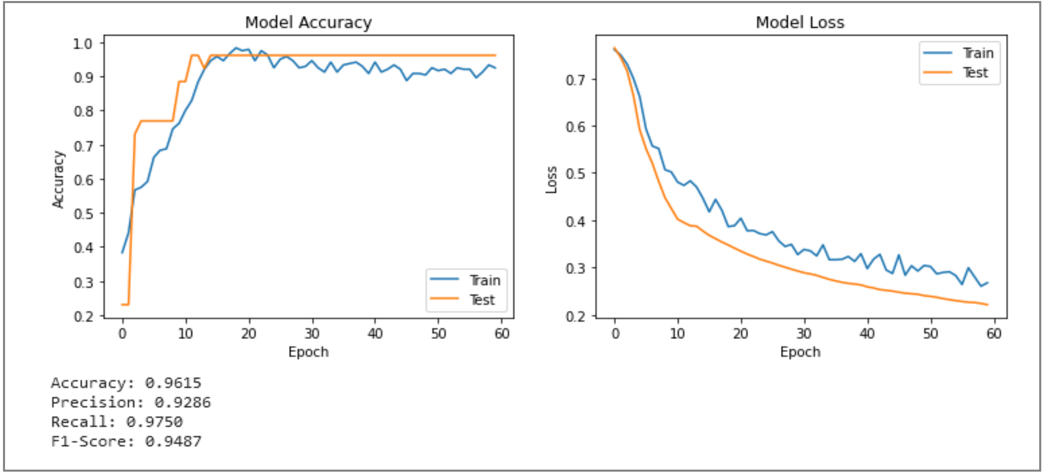
The model was trained using 80% of the available data, with the remaining 20% set aside for testing. The assessment of the models includes the utilization of different performance measures, encompassing accuracy, specificity, F1 score, precision, and the recall. Accuracy is a measure used to assess the predictive ability of a DL model through the comparison of the expected outcome with the actual outcome. In the context of predicting heart disease, the classifier's ability to precisely determine the existence or non-existence of the disease is assessed through the true positive (TP) and true negative (TN) values. Conversely, false positive (FP) and false negative (FN) values highlight the inaccuracies in the models' predictions. Precision gauges the ratio of observed positive instances among all the predicted positive instances. Recall, alternatively known as sensitivity or the true positive rate, computes the ratio of actual positive instances correctly identified by the model. Specificity, conversely, evaluates the ratio of all negative instances that the model accurately predicts. Meanwhile, the F1 score is a metric that amalgamates precision and recall, delivering a harmonized measure of the model's performance. It computes the harmonic mean of precision and recall, assigning equal significance to both metrics.

Figure 4 illustrates the results obtained during the training and testing phase of the proposed model. This figure highlights a crucial role in evaluating and analyzing the performance of the proposed model, providing insights into its training and testing performance, respectively. It provides the value of accuracy and loss that evaluate the efficacy of the model and its performance on the unobserved test data.

By examining Figure 4, one can gain insights into how well the model performs on the test data, allowing for a comprehensive understanding of its performance and the potential impact on real-world scenarios. The proposed system demonstrates exceptional performance with an accuracy of 96.15%, precision of 92.86%, recall (sensitivity) of 97.50%, and an F1-score of 94.87%.The accuracy achieved for the proposed model was 96.15%. This indicates that the model's predictions aligned with the desired output in approximately 96.15% of cases. A high accuracy value like this suggests that the model performed exceptionally well in accurately classifying instances and making correct predictions.

4.2 Comparative analysis

The evaluation of the suggested work centers on assessing prediction accuracy through the application of state-of-the-art approaches to heart disease datasets. A comparative analysis has been conducted out to analyze the accuracy results of the proposed model in comparison to existing models documented in the literature (Table 1).



```
Accuracy: 0.9615
Precision: 0.9286
Recall: 0.9750
F1-Score: 0.9487
```

Figure 4: Training and testing performance: accuracy and loss across epochs

The outcomes of this comparison are illustrated in Table 3. In our research, we propose a deep learning model that achieves an accuracy of 96.15%, demonstrating its superior performance compared to several established methods. Specifically, our model outperforms the Meta classification technique, which achieves an accuracy of 85.48%, and the Hybrid random forest with a linear model, which reaches 88.70%. It also surpasses the Statistical model and deep neural network with 91.57% accuracy and the bi-directional LSTM (C-BiLSTM) algorithm, which attains 94.78%. Although hyperparameter tuning and cross-validation with machine learning yield 94.90% accuracy, and the traditional Random Forest achieves 95.25%, our model still exhibits better performance. Other techniques, such as CNN (93.98%), MLP (91.20%), RNN (91.00%), and the Extra Tree Classifier with SMOTE (92.62%), also fall short in comparison. The Deep Forest Cascade Technique (92.56%) and Random Forest without SMOTE (88.89%) further underscore the effectiveness of our proposed approach.

Despite the high accuracy achieved by the Optimised Ensemble Fuzzy Ranking (OEFR) strategy (96.72%), the LSTM model (96.91%), and the Fuzzy Information System with Bi-LSTM (98.86%), our approach presents several notable advantages. While these models demonstrate impressive results, they are often trained on datasets containing only 14 attributes, which limits the scope of patient information considered. In contrast, our model is trained on a dataset with 21 attributes, providing a more comprehensive analysis by integrating additional health parameters that are critical for precise heart disease prediction. Furthermore, the aforementioned methods do not account for real-time constraints, nor do they consider training time, which is essential for practical deployment in dynamic healthcare environments. Our approach emphasizes real-time prediction, leveraging edge computing to process data swiftly and effectively, making it better suited for real-world applications where timely responses are crucial. Thus, while some models may have higher reported accuracy, our model's comprehensive feature set and real-time capability underscore its suitability and effectiveness in practical healthcare settings.

The outcomes of the comparison with previously conducted related studies demonstrate that the proposed system outperforms these systems. With the increasing importance of real-time smart systems in healthcare, which heavily rely on IoT technology, tasks such as rapid processing become critical as they require minimal delays and are context-sensitive.

4.3 Training time comparison

To provide a more comprehensive time comparison, it's important to include the hardware specifications of the environment in which the experiments were conducted. In this study, the computations were performed using Google Colab, which utilizes a cloud-based GPU environment. Specifically, the model was run on a Tesla T4 GPU with a standard CPU configuration, offering a balance of performance and accessibility. The comparison of training times for various models, including transfer learning models, the CNN model proposed in [23], and the newly proposed DNN model, is shown in Table 4. This comparison highlights their computational complexity. Despite the variations in training conditions, such as the data used and the hardware, the newly proposed DNN model demonstrates a significantly shorter training time compared to the other models. This efficiency is particularly crucial for real-time processing, as it allows for faster model updates and deployment, which is essential in dynamic environments. The reduced training time, combined with the model's high accuracy, makes it especially suitable for edge computing applications. Implementing this model on edge devices enables real-time analysis and decision-making, thereby enhancing its practicality and effectiveness in real-world scenarios. The efficient processing and deployment capabilities of the proposed model highlight its advantages in applications requiring immediate results and timely responses.

Table 3: A comparative examination of the proposed model against existing models.

Approach	Acc	Author-Year-Reference
Meta classification technique	85.48	Latha et Jeeva (2019) [16]
Hybrid random forest with a linear model	88.70	Mohan et al. (2019) [17]
Statistical model and deep neural network	91.57	Moreno-Ibarra et al. (2019) [18]
bi-directional LSTM (C-BiLSTM) algorithm	94.78	Dileep et al. (2023) [19]
Hyperparameter tuning and cross-validation with machine learning	94.90	Ahmed et al. (2020) [20]
Random Forest	95.25	Dhanamjayulu et al. (2022) [21]
Optimised ensemble fuzzy ranking (OEFR) strategy	96.72	Managala et al. (2023) [22]
CNN	93.98	Almujally et al. (2023) [23]
MLP	91.20	Almujally et al. (2023) [23]
RNN	91.00	Almujally et al. (2023) [23]
LSTM	96.91	Almujally et al. (2023) [23]
Random Forest without SMOTE	88.89	ishaq et al. (2023) [26]
Extra Tree Classifier with SMOTE	92.62	ishaq et al. (2023) [26]
XGBoost	93.26	Gracious et al. (2024) [25]
Deep Forest Cascade Technique	92.56	Askar (2023) [24]
Fuzzy information system and Bi-LSTM	98.86	Nancy et al.(2022) [2]
Proposed approch	96.15	

Table 4: The comparison of training times

Model	Training time
AlexNet (transfer learning)	32 min
VGG-16 (transfer learning)	29 min
CNN	24 min
Proposed	12 sec

4.4 Advantages of edge architecture

We have proposed an edge architecture model combined with a deep learning framework for heart disease prediction, achieving remarkable performance. The exponential proliferation of devices and the resulting surge in data traffic have significantly increased bandwidth consumption and service disruptions. The traditional cloud model struggles with issues like latency, bandwidth utilization, and connectivity, making it insufficient to handle these challenges alone. Our decentralized edge computing model addresses these limitations by processing and storing data close to the source. This proximity allows for efficient handling of vast IoT data using AI tools at the edge layer, significantly reducing latency and managing the substantial data volume from IoT devices. By integrating a hierarchical edge-fog-cloud architecture, our model enhances performance and reliability in heart disease prediction, leveraging the advantages of edge and fog computing to deliver superior predictive capabilities in healthcare applications.

4.5 K-Fold Cross-validation

To validate our proposed model, we applied 10-fold cross-validation using a heart failure dataset. The results presented in Table 5 demonstrate the model's robust performance, with an average accuracy of 96.99%. The precision, recall (sensitivity), and F1-score achieved are 96.30%, 97.33%, and 96.63%, respectively.

Table 5: 10 cross-validation results

Number of Fold	Accuracy	Precision	Sensitivity (Recall)	F1-Score
Fold 1	96.30	95.45	97.06	96.10
Fold 2	100	100	100	100
Fold 3	92.59	90	94.74	91.67
Fold 4	96.30	95.45	97.06	96.10
Fold 5	100	100	100	100
Fold 6	96.30	97.37	94.44	95.71
Fold 7	96.15	95.00	97.06	95.85
Fold 8	100	100	100	100
Fold 9	96.15	96.88	95.45	96.01
Fold 10	96.15	92.86	97.50	94.87
Average	96.99	96.30	97.33	96.63

These metrics indicate that the model consistently performs well across different folds. Specifically, several folds, such as Fold 2, Fold 5, and Fold 8, achieved perfect scores (1.00) across all metrics, highlighting the model's capability to accurately classify patient data.

Other folds, such as Fold 1, Fold 4, and Fold 6, also showed high performance with accuracy and F1-scores above 96%.

In our k-fold cross-validation study, we enhance the statistical interpretation presented in Table 6 by calculating the standard deviation and 95% confidence intervals for each metric. The standard deviation provides insight into the variability or spread of the model's performance across the folds, while the 95% confidence intervals give a range within which the true metric value is likely to fall, offering a measure of reliability and stability.

Table 6: Statistical interpretation of k-fold cross-validation study

Metric	Mean	Stand. Deviat.	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Accur.	96.99	2.24	95.61	98.38
Precis.	96.30	3.12	94.37	98.23
Sensit.	97.33	2.01	96.09	98.57
F1-scor.	96.63	2.53	95.06	98.20

The standard deviation values reflect the consistency of the model's performance. Smaller standard deviations, such as for accuracy (2.24%), indicate more stable results across folds, while larger values, like for precision (3.12%), suggest slightly higher variability in the model's precision across different data splits. The calculated 95% confidence intervals for each metric indicate the consistency and stability of the model's performance across different folds. For accuracy, the interval is between 95.61% and 98.38%, showing minimal variation. Precision's confidence interval spans from 94.37% to 98.23%, suggesting slightly more variability. Sensitivity has an interval from 96.09% to 98.57%, demonstrating reliable detection of true positives with little fluctuation. The F1-score ranges from 95.06% to 98.20%, confirming a robust balance between precision and recall. These intervals reflect stable and consistent model performance.

These results affirm the effectiveness of the proposed edge architecture combined with a deep learning model in predicting heart disease, significantly outperforming traditional methods by leveraging the advantages of edge computing to process data efficiently and accurately.

5 Conclusion

This work introduces a model for a Real-Time Smart Healthcare System, tailored for predicting the risk of heart disease, leveraging Edge-IoT and DL technologies. The architecture consists of three layers, each with its required components, and employs a deep learning model for the task of prediction. The suggested system demonstrates outstanding performance, boasting average accuracy, precision, sensitivity, and F1-score values of 96.99%, 96.30%, 97.33%, and 96.63, respectively, surpassing other current models for predicting heart disease. However, this constitutes just one facet of the continual healthcare research using predictive analytics, with deep learning models holding tremendous untapped potential.

The model can be improved to autonomously generate tailored diet and exercise suggestions, taking into account an individual's health condition and guidance from a heart specialist. In this envisioned intelligent system for predicting heart disease, IoT devices are utilized for data acquisition, while edge computing manages data analysis, with the cloud reserved for other essential tasks. Nevertheless, our study acknowledges certain limitations, including the necessity for real-world validation, addressing privacy concerns, and ensuring scalability and compatibility within diverse healthcare infrastructures. Continued exploration and refinement of real-time smart healthcare systems are imperative for realizing their full potential in transforming healthcare delivery and improving patient outcomes.

To support real-world deployment, future work will focus on several concrete steps. These include rigorous testing in clinical environments to ensure model robustness and compliance with healthcare standards such as HIPAA or GDPR for data privacy. Enhancements to the system could allow for personalized diet and exercise recommendations tailored to individual health profiles, under medical guidance. Additionally, we plan to incorporate real-time patient feedback and specialist input, enabling the model to learn from real-world cases and improve over time.

The healthcare sector's effectiveness can undergo a transformation through accurate and timely disease forecasts, facilitating real time responses and smart decision-making by healthcare professionals, especially when leveraging fog/edge computing technologies. This integration has the potential to enhance the overall quality-of-service and revolutionize the healthcare industry.

Declarations

Competing interests

The authors affirm that they have no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

Ethical and informed consent for data used

The Erbil Heart Disease dataset (https://www.kaggle.com/datasets/hangawqadir/erbil-heart-disease-dataset) adheres to strict ethical guidelines and informed consent protocols to ensure the appropriate use and handling of data.

References

[1] M. Hartmann, U. S. Hashmi, and A. Imran, "Edge computing in smart health care systems: Review, challenges, and research directions," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 3, p. e3710, 2022, https://doi.org/10.1002/ett.3710.

[2] A. A. Nancy, D. Ravindran, P. M. D. Raj Vincent, K. Srinivasan, and D. Gutierrez Reina, "IoT-Cloud-Based Smart Healthcare Monitoring System for Heart Disease Prediction via Deep Learning," *Electronics*, vol. 11, no. 15, Art. no. 15, Jan. 2022, https://doi.org/doi: 10.3390/electronics11152292.

[3] M. Gams and T. Kolenik, "Relations between electronics, artificial intelligence and information society through information society rules," *Electronics*, vol. 10, no. 4, p. 514, 2021, Accessed: Nov. 10, 2024. [Online]. Available: https://www.mdpi.com/2079-9292/10/4/514

[4] S. Vimal, Y. H. Robinson, S. Kadry, H. V. Long, and Y. Nam, "IoT based smart health monitoring with CNN using edge computing," *Journal of Internet Technology*, vol. 22, no. 1, pp. 173–185, 2021. [Online]. Available: https://jit.ndhu.edu.tw/article/view/2471.

[5] X. Wang and Y. Li, "Edge Detection and Simulation Analysis of Multimedia Images Based on Intelligent Monitoring Robot," *Informatica*, vol. 48, no. 5, Art. no. 5, Mar. 2024, https://doi.org/doi: 10.31449/inf.v48i5.5366.

[6] X. Wu, C. Liu, L. Wang, and M. Bilal, "Internet of things-enabled real-time health monitoring system using deep learning," *Neural Comput & Applic*, Sep. 2021, https://doi.org/doi: 10.1007/s00521-021-06440-6.

[7] M. Belkebir, T. M. Maarouk, and B. Nini, "Real-time Semantic Healthcare System: Visual Risks Identification for Elders and Children," *Informatica*, vol. 48, no. 14, Art. no. 14, Sep. 2024, https://doi.org/doi: 10.31449/inf.v48i14.6271.

[8] G. S. Aujla and A. Jindal, "A Decoupled Blockchain Approach for Edge-Envisioned IoT-Based Healthcare Monitoring," *IEEE J. Select. Areas Commun.*, vol. 39, no. 2, pp. 491–499, Feb. 2021, https://doi.org/doi: 10.1109/JSAC.2020.3020655.

[9] M. Bhatia and S. K. Sood, "A comprehensive health assessment framework to facilitate IoT-assisted smart workouts: A predictive healthcare perspective," *Computers in Industry*, vol. 92–93, pp. 50–66, Nov. 2017, https://doi.org/doi: 10.1016/j.compind.2017.06.009.

[10] A. Alabdulatif, I. Khalil, X. Yi, and M. Guizani, "Secure Edge of Things for Smart Healthcare Surveillance Framework," *IEEE Access*, vol. 7, pp. 31010–31021, 2019,https://doi.org/doi: 10.1109/ACCESS.2019.2899323.

[11] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, "Deep learning for healthcare: review, opportunities and challenges," *Brief Bioinform*, vol. 19, no. 6, pp. 1236–1246, Nov. 2018, https://doi.org/doi: 10.1093/bib/bbx044.

[12] S. K. Pandey and R. R. Janghel, "Recent Deep Learning Techniques, Challenges and Its Applications for Medical Healthcare System: A Review," *Neural Process Lett*, vol. 50, no. 2, pp. 1907–1935, Oct. 2019, https://doi.org/doi: 10.1007/s11063-018-09976-2.

[13] J. Botros, F. Mourad-Chehade, and D. Laplanche, "CNN and SVM-Based Models for the Detection of Heart Failure Using Electrocardiogram Signals," *Sensors*, vol. 22, no. 23, Art. no. 23, Jan. 2022, https://doi.org/doi: 10.3390/s22239190.

[14] N. Chandrasekhar and S. Peddakrishna, "Enhancing Heart Disease Prediction Accuracy through Machine Learning Techniques and Optimization," *Processes*, vol. 11, p. 1210, Apr. 2023, https://doi.org/doi: 10.3390/pr11041210.

[15] S. Tiwari *et al.*, "A smart decision support system to diagnose arrhythymia using ensembled ConvNet and ConvNet-LSTM model," *Expert Systems with Applications*, vol. 213, p. 118933, Mar. 2023, https://doi.org/doi: 10.1016/j.eswa.2022.118933.

[16] C. B. C. Latha and S. C. Jeeva, "Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques," *Informatics in Medicine Unlocked*, vol. 16, p. 100203, Jan. 2019, https://doi.org/doi: 10.1016/j.imu.2019.100203.

[17] S. Mohan, C. Thirumalai, and G. Srivastava, "Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques," *IEEE Access*, vol. 7, pp. 81542–81554, 2019,https://doi.org/doi: 10.1109/ACCESS.2019.2923707.

[18] M.-A. Moreno-Ibarra, Y. Villuendas-Rey, M. D. Lytras, C. Yáñez-Márquez, and J.-C. Salgado-Ramírez, "Classification of diseases using machine learning algorithms: A comparative study," *Mathematics*, vol. 9, no. 15, p. 1817, 2021. [Online]. Available: https://www.mdpi.com/2227-7390/9/15/1817

[19] P. Dileep *et al.*, "An automatic heart disease prediction using cluster-based bi-directional LSTM (C-BiLSTM) algorithm," *Neural Comput & Applic*, vol. 35, no. 10, pp. 7253–7266, Apr. 2023, https://doi.org/doi: 10.1007/s00521-022-07064-0.

[20] H. Ahmed, E. M. G. Younis, A. Hendawi, and A. A. Ali, "Heart disease identification from patients' social posts, machine learning solution on Spark," *Future Generation Computer Systems*, vol. 111, pp. 714–722, Oct. 2020, https://doi.org/doi: 10.1016/j.future.2019.09.056.

[21] C. Dhanamjayulu, G. V. Suraj, M. Nikhil, R. Kaluri, and S. Koppu, "A Machine Learning Algorithm-Based IoT-Based Message Alert System for Predicting Coronary Heart Disease," in *Advancements in Smart Computing and Information Security*, S. Rajagopal, P. Faruki, and K. Popat, Eds., in Communications in Computer and Information Science. Cham: Springer Nature Switzerland, 2022, pp. 362–376. https://doi.org/doi: 10.1007/978-3-031-23092-9_29.

[22] N. V. L. M. K. Munagala, L. R. R. Langoju, A. D. Rani, and D. V. rama K. Reddy, “Optimised ensemble learning-based IoT-enabled heart disease monitoring system: an optimal fuzzy ranking concept,” *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 0, no. 0, pp. 1–17, Jan. 2023, https://doi.org/doi: 10.1080/21681163.2022.2162439.

[23] N. A. Almujally *et al.*, “Monitoring acute heart failure patients using internet-of-things-based smart monitoring system,” *Sensors*, vol. 23, no. 10, p. 4580, 2023, Accessed: Jun. 01, 2024. [Online]. Available: https://www.mdpi.com/1424-8220/23/10/4580

[24] S. K. Askar, “Deep Forest Based Internet of Medical Things System for Diagnosis of Heart Disease,” *ARO-THE SCIENTIFIC JOURNAL OF KOYA UNIVERSITY*, vol. 11, no. 1, Art. no. 1, Mar. 2023, https://doi.org/doi: 10.14500/aro.11174.

[25] G. S and S. C P, “ML based Age Related Heart Disease Prediction,” in *2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC - ROBINS)*, Apr. 2024, pp. 312–316. https://doi.org/doi: 10.1109/ICC-ROBINS60238.2024.10533975.

[26] A. Ishaq *et al.*, “Improving the Prediction of Heart Failure Patients’ Survival Using SMOTE and Effective Data Mining Techniques,” *IEEE Access*, vol. 9, pp. 39707–39716,2021,https://doi.org/doi:10.1109/ACCESS.2021.3064084.

[27] S. Basak and K. Chatterjee, “Smart Healthcare Surveillance System Using IoT and Machine Learning Approaches for Heart Disease,” in *Advancements in Smart Computing and Information Security*, S. Rajagopal, P. Faruki, and K. Popat, Eds., in Communications in Computer and Information Science. Cham: Springer Nature Switzerland, 2022, pp. 304–313. https://doi.org/doi: 10.1007/978-3-031-23092-9_24.

[28] P. Rajan Jeyaraj and E. R. S. Nadar, “Smart-Monitor: Patient Monitoring System for IoT-Based Healthcare System Using Deep Learning,” *IETE Journal of Research*, vol. 68, no. 2, pp. 1435–1442, Mar. 2022, https://doi.org/doi:10.1080/03772063.2019.1649215.

[29] “Erbil Heart Disease Dataset.” Accessed: Jun. 05, 2024. [Online]. Available: https://www.kaggle.com/datasets/hangawqadir/erbil-heart-disease-dataset

An Image Processing-Based Statistical Method for Estimating Nutrient Deficiencies in Grape Plants During the Growing Season

G. Adiline Macriga[1], Sankari Subbiah[2*], G. Sudha[3] and S. Saranya[4]
[1,2]Department of Information Technology, Sri Sai Ram Engineering College, Chennai, India
[3,4]Department of Electronics and Communication Engineering, Sri Sai Ram Engineering College, Chennai, India
E-mail: adiline.it@sairam.edu.in, sankari2705@gmail.com, sudhaganesh74@gmail.com, saranya.ece@sairam.edu.in
[*]Corresponding author

Keywords: yield estimation, nutrition identification, classification, image processing

Received: December 1, 2023

Agriculture plays a very important role in the provision of food surplus to expanding population, contribution to capital formation, providing raw material to industries, market for industrial products and major contribution in international trade. To enhance the quality and quantity of the agriculture product there is a need to adopt the new technology. Image processing approach is non-invasive technique which provides consistent, reasonably accurate, less time consuming and cost-effective solution for farmers to manage fertilizers and pesticides. The objective of this study is to analyze the nutrition of grape plant using Wavelength algorithmand statistical method regression analysis is used in our work for the nutrient estimation. The relative requirements of nitrogen, phosphorus and potassium in grapes vary with the growth stages of grapes. There is a high-level requirement of N during the vegetative growth stage, P requirement is high during flowering stage and K requirement is more in crop maturity stage. So pruning was carried out in two stages i.e., in April and in October. This study is carried out in grape farms of Theni District of Tamil Nadu for estimating the macro nutrition nitrogen (N), phosphorus (P) and potassium (K) to analyze the yield. The results showed that the overall identification accuracies of NPK deficiencies were 86.15, 87.69, 90.00 and 89.23% for the two pruning stages.

Povzetek: Razvili so neinvazivno metodo za oceno pomanjkanja hranil v grozdju s pomočjo analize slik in statističnih metod, kar omogoča prepoznavanje pomanjkanja dušika, fosforja in kalija ter s tem izboljšanje kakovosti in količine pridelka.

1 Introduction

The agriculture sector is the main contributor in the Indian economy and is doing well in white, green and blue revolution. According to APEDA, by 2014 exports of Indian agriculture will reach 5% of total production of the world and rank 10th in the ranking [1]. Precision agriculture is a new and developing technology which leads to incorporating the advanced techniques to enhance farm output and also enrich the farm inputs in a profitable and environmentally sensible manner. Farm inputs were important parameters to be controlled and if not will result in adverse effects causing reduction in yield, deteriorating plant health, etc. Irrigation/Water stress, Fertilizers, pesticides and quality of yield were the major factors of concern in agriculture. Most of the time expertise is required to analyze the problems, which may be a time consuming and costly issue in developing countries. Image processing is one of the tools which can be applied to measure the parameters related to agronomy with accuracy and economy. Applications of image processing in agriculture can be broadly classified in two categories: first one depends upon the imaging techniques and the second one based on applications.

Less yield, higher cost of production due to labour scarcity and fertilizer cost are the major challenges before the farmers. To enhance the quality and quantity of the agriculture product there is a need to adopt the new technology. Fertilizer and pesticide management requires early and cost-effective solutions which will lead to higher yield. Image processing approach is a non-invasive technique which provides consistent, reasonably accurate, less time consuming and cost-effective solutions for farmers to manage fertilizers and pesticides. The soft computing techniques are helpful in developing the knowledge-based systems, and may be effectively utilized to develop the expert system. This system will be helpful for farmers to find the solutions to their farming problems in the existing system such as Adaptation to climate change in viticulture, integration with other nutrient assessments and expansion to other cros and environments. The research objective is aimed to identify/develop color models for Fertilizer's estimation viz. Nitrogen, Phosphorous, Potassium and Magnesium for grapes as a horticulture product.

2 Related works

Plants need a certain number of macronutrients (nitrogen, phosphorus, etc.) and micronutrients (Zinc, Boron, etc.) to grow and stay healthy. These Nutrients alter and regulate the functioning of plants and produce qualitative and quantitative changes in plant yield. Nutrient deficiency may result in weaker plants. Nutrient deficiencies make plants more susceptible to diseases. Fertilizers are the supplements for the plants to grow healthier. In case of grapevines, the deficiencies or overdoses are observed by visual inspections by experts. Availability of experts on time and their consultancy cost are the major issues. Nitrogen application to grapevine in excess form results in excessive growth of shoots at the cost of fruit set. This will also result in a delay in maturity and poor bud formation in the following season [1].

The Proper application of fertilizers and pesticides can save cost of production. In most of the cases, deficiencies are observed by a change in the color of leaves in which Chlorophyll is an important ingredient. Chlorophyll is a molecule in a leaf which is responsible for photosynthesis action. The carbohydrates produced in photosynthesis are used as food for growth of plants and fruit. Various methods are proposed to analyze the nutrients based on chemical analysis such as leaf analysis, petiole analysis [2]

Vasifa A. Aglave et al., reported a survey for chlorophyll content, disease severity and leaf area measurement. Leaf area measurement techniques viz. grid counting, paper weighing, leaf area meter, digital image analysis using different techniques were reported. Naked eye observation, image processing techniques like chain code, bounding box, segmentation were reported for measuring disease severity along with different chlorophyll content measurement [3]. M.M. Ali et al., presented different non destructive handheld meter techniques such as leaf color chart, SPAD meter, N-tester, Image analysis to estimate foliar N status of plant. Comparison of these methods was also highlighted on the basis of applicability, accuracy, the effect of environment, etc. [4].

The color image analysis became a popular and cost-effective method for chlorophyll and nutrient estimation. Paula F. Murakami has developed digital imaging software like scion which was used for analysis and quantification of chlorophyll using leaf color. The software calculates the percentage green and red for chlorophyll estimation. RGB and HSV color models were considered in development [5]. Parviz Moghaddam et al., developed an algorithm to estimate chlorophyll using a video camera in which it was shown that the red and blue elements were highly correlated with it. Normalized difference of red and blue was effectively used to estimate chlorophyll under different meteorological conditions. The multilayer perceptron neural network was developed using R, G and B values for chlorophyll estimation. The results show the high coefficient of determination (R2) and low mean square error where the estimated values were compared with chlorophyll SPAD meter [6].

Mario Cupertino da Silva et al., presented the work that showed the correlation between vegetation indices and nitrogen. The Correlation between vegetation indices and nitrogen leaf content and dry matter at different stages for fertilization was calculated with IR camera and digital camera. The study revealed that the high positive correlation decreases as the number of days increases after fertilization and green spectral band is more useful for nitrogen discrimination. Three indices NDVI, GNDVI and SAVI were evaluated and observed that GNDVI was the best [7].

Han Yuzhuet al., proposed a method of nitrogen estimation for pepper in flowering and fruiting using color image processing. Different functions of RGB were considered and correlated with nitrogen. Regression analysis for inorganic nitrogen in soil, total nitrogen, nitrogen concentration and SPAD meter readings were done and all show negative coefficient for the considered function. N applications were given in different treatments [8].

Gloria F. Mata-Donjuan et al., induced five levels of nitrogen deficiencies and proposed improved hue, luminance and saturation color space which is less susceptible to illumination variation. IHLS was used to estimate the nitrogen for tomato seedlings. Image processing methodologies like segmentation, RGB to IHLS conversion and image analysis were implemented. He showed better statistical relation with nitrogen values obtained from chemical analysis. Estimation N was based on components histograms in IHLS and in the fusion of saturation and hue [9].

Table 1 summarizes the literature survey of research methodology, Key focus, research findings, challenges and future research directions.

Table 1: Summary of literature survey and its contributions

Ref.	Topic	Key Focus	Methodology/ Technology	Application /Outcome	Limitations /Challenges	Future Research Directions	Key Findings
[1]	Vitinotes	Viticulture research	Review of viticulture practices	Enhancing viticulture through research insights	General applicability across regions	Adaptation to climate change in viticulture	Insights into improving viticulture through research

[2]	Grapevine structure and function	Grapevine biology	Analysis of grapevine anatomy and physiology	Better viticulture practices	Limited to structural aspects	Integration with genetic research	Detailed understanding of grapevine anatomy supports better practices
[3]	Imaging techniques	Leaf area, disease severity, chlorophyll content	Imaging technology and algorithms	Non-destructive plant health analysis	Accuracy of imaging algorithms	Development of more robust algorithms	Imaging techniques can accurately assess plant health
[4]	Leaf nitrogen determination	Nitrogen content in leaves	Handheld meters	Efficient nitrogen assessment	Limited to nitrogen; excludes other nutrients	Integration with other nutrient assessments	Handheld meters provide reliable nitrogen measurements
[5]	Leaf color analysis	Digital leaf color analysis	Digital imaging software	Standardized leaf color measurement	Software complexity and accuracy	Simplification and automation of software	Digital imaging offers standardized leaf color analysis
[6]	Chlorophyll content estimation	Chlorophyll in sugar beet leaves	Machine vision	Precise chlorophyll content estimation	Limited to sugar beet; may not apply to other crops	Expansion to other crops and environments	Machine vision accurately estimates chlorophyll content
[7]	Vegetation indices correlation	Vegetation indices, nitrogen, dry matter production	Image analysis	Improved vegetation health assessment	Correlation may vary by species and region	Exploration of species-specific indices	Yes
[8]	Nitrogen determination in pepper plants	Nitrogen content estimation using RGB	RGB color image analysis	Effective nitrogen assessment for pepper plants	RGB analysis might be affected by lighting conditions	Improved robustness against environmental variations	RGB analysis is effective but sensitive to light conditions
[9]	Nitrogen estimation in tomato seedlings	Nitrogen estimation using IHLS color space	IHLS color space	Accurate nitrogen estimation in tomato seedlings	IHLS method complexity	Application to other nutrient estimations	IHLS color space offers precise nitrogen estimation
[10]	Grapevine nutritional status	Grapevine nutrition estimation	Proximal sensing techniques	Non-destructive monitoring of grapevine health	Limited to proximal sensing; excludes remote sensing	Combination with remote sensing technologies	Proximal sensing is effective for real-time nutritional assessment
[11]	Grapevine nutrition	Remote sensing-based estimation	Remote sensing	Accurate field-based grapevine nutrition monitoring	Sensitivity to environmental factors	Refinement of remote sensing techniques	Remote sensing accurately estimates grapevine nutrition
[12]	Nutrient estimation in grapevine leaves	Nitrogen, phosphorus, potassium, magnesium content estimation	Near-infrared spectroscopy (NIRS)	Rapid nutrient estimation in grapevines	NIRS may require calibration for different environments	Development of universal calibration models	NIRS provides quick and accurate nutrient estimation

[13]	Vineyard nutritional status	Vineyard nutrition assessment	Proximal sensing	Challenges and opportunities in vineyard management	Limited to proximal sensing; excludes other methods	Integration with comprehensive vineyard management systems	Proximal sensing reveals key challenges in vineyard management
[14]	Deficit irrigation	Water-saving in horticulture	Integrative plant biology	Water conservation and improved horticultural practices	May not apply to all horticultural crops	Testing in diverse horticultural environments	Deficit irrigation effectively conserves water without harming crops

3 Materials and methods

3.1 Experimental design

The goal of the experiment was to investigate the grape plant's yield analysis under various levels of NPK nutrition. The experiment was carried out in commercial vineyards in areas around Theni of Tamil Nadu. The vines were planted in a vertical shoot-positioning technique with a north-south row orientation at a distance of 2 m. There were five different grapevine varieties utilised in this experiment. Hydroponics with a nutrient solution formula was applied to cultivate the grape plants. After berry set, the six first basal leaves of the selected plants were carefully plucked.

3.2 Sample/ leaf selection

Changes in nutrients are reflected in plant appearance or chemical composition of plant tissues or petioles. Plant analysis for nutrient contents can be carried out either by chemical analysis or visual diagnosis of deficiencies/ toxicity. Standards are developed for petiole sampling through research in various parts of the world and India (NRC or MRDBS, Pune). These standards are for optimal growth of grapevine. Since the nutritional requirement varies over the season, it is necessary to match the standard requirement of nutrients [3]. While sampling, few factors need to be considered viz. time, site, and plant part of sample.

3.3 Image acquisition

Images are acquired with proper selection of camera and background. In two different seasons—one in May or June and the other in November or December—pictures are taken outside in the sunshine. Days with plenty of sunshine were selected to reduce shadow variations and increase light uniformity. The camera's flash was disabled in order to prevent glare and reflections, which could skew colour and intensity measurements that are essential for precise analysis. Leaf images are captured with a digital camera having a CCD sensor of Nikon Coolpix S570. The model is selected considering the general availability and cost effectiveness. CCD camera is preferred than CMOS as CCD's are less susceptible to noise, to maintain image quality under varying light conditions as compared to CMOS. The camera is kept at a distance of 7-8 inches from the ground level or leaf blade surface to ensure consistency in capturing image details across all samples. At a resolution of 3 mega pixels (2048*1536), the camera was configured in Normal mode. Fine details in leaf texture and colour can be captured at this resolution, which strikes a balance between image quality and file size. In order to keep things simple and make the leaf the main focal point, a neutral background was selected. Segmentation during image processing is aided and colour interference is less likely.

The following image pre-processing steps are needed to ensure high quality image acquisition:

Color calibration: Images are colour calibrated prior to processing in order to correct for lighting-related colour variations. Accurate colour representation is aided by this, and extracting RGB, HSV, and other indices depends on it.

Noise reduction: To improve the accuracy of color-based indices such as Opponent HSV and modified indices, further filtering is carried out to eliminate any remaining noise, even though CCD sensors naturally minimise noise.

Thresholding: The three thresholds (80, 100, and 112) on pixel values are set during pre-processing. This is necessary in order to separate particular colour bands and characteristics associated with estimating nutrients.

Segmentation: To make sure that just the leaf pixels are examined, the leaf region is divided from the background. To extract colour parameters independently of surrounding elements, this segmentation is essential.

3.4 Nutrient analysis for post april pruning

Sampling techniques after April pruning and October pruning are different. The annual cycle growth of grapes involves many processes and events. The timing and duration of development of events vary due to rootstock variety, local climate and seasonal weather [4]. In India because of temporal zone the growth of the plant continues, hence the pruning process was developed for two times. We can classify the development processes into stages after April pruning viz. vegetative growth/ bud differentiation stage and after October pruning viz. flower cluster initiation (Bloom), fruit maturity and harvest. Many of these processes overlap each other during the development process. Vigorous and healthy growth is required during the pre - initiation stage of floral primordial, slow and less growth in fruit bud differentiation is required for higher yield of grapes. The need for N is higher during plant growth, the need for P is higher during flowering and the need for K is higher during crop maturity.

3.5 Nitrogen estimation using wavelength algorithm

The human color perception is scientifically and technically defined by colorimetry. Human visual sensitivity, illumination sources and spectral measurements are the major issues in colorimetry. The tri stimulus values X, Y and Z have been defined by CIE which represents the human color vision mathematically. CIE has defined various illumination sources such as A, B, C and D, but for this experiment we considered daylight D65 source with 1964 color observer. The method of CIE colorimetric specification depends on rules of color matching by additive color mixture. Mathematically, CIEXYZ is represented as equations 1 to 3 where *S(λ)*is object spectrum, *E(λ)*is the spectral power distribution (SPD) of an illuminant, and the color matching functions (CMFs) *A(λ)* ={ *x*(*λ*), *y* (*λ*), *z* (*λ*)} where *λ* is the wavelength.

$$X = \int \bar{x}\,(\lambda)E(\lambda)S(\lambda)\,d\,\lambda \qquad \text{----------} \text{ Eqn 1}$$

$$Y = \int \bar{y}\,(\lambda)E(\lambda)S(\lambda)\,d\,\lambda \qquad \text{----------} \text{ Eqn 2}$$

$$Z = \int \bar{z}\,(\lambda)E(\lambda)S(\lambda)\,d\,\lambda \qquad \text{----------} \text{ Eqn 3}$$

Often tri stimulus space is represented in 2D space and can be represented by normalized component as equation 4 to 6

$$\text{x} = \frac{x}{(x+y+z)} \qquad \text{----------} \text{ Eqn 4}$$

$$\text{y} = \frac{y}{(x+y+z)} \qquad \text{----------} \text{ Eqn 5}$$

$$\text{z}= \frac{z}{(x+y+z)} \qquad \text{----------} \text{ Eqn 6}$$

Most of the researchers used the chlorophyll meter to estimate or relate the nutrient parameters of the plants. Chlorophyll measurement is a non-destructive method which has been proved to be less costly and effective method. Chlorophyll meter works on the principle of emission of two frequencies of light, one is red at wavelength 660 nm and other is infrared at 940 nm. Leaf chlorophyll absorbs red light but infrared is not absorbed. The difference between the absorptions is calculated as an index of chlorophyll. Since nitrogen is related with the chlorophyll contents, concept of chlorophyll meter can be extended for measurement of nitrogen from images. In this experiment an effort has been made to estimate the wavelength from leaf image and is illustrated as in figure 1. Values of RGB are calculated using XYZ for respective wavelength of CIE1964 full record. These RGB values are useful in interpolation to estimate the wavelength based on image.

In this experiment an innovative effort has been made to estimate the wavelength of RGB image and used as new index to estimate nitrogen. To achieve the principle of wavelength-based estimation of nutrient parameter following algorithm is developed.

The calculated wavelength is the function of r, g and b values of the image and can be represented as equation 7

$$\lambda = f(r,g,b) \qquad \text{----------} \text{ Eqn 7}$$

April pruning data along with variation is used for analysis purpose. Depending on the values of RGB and chart prepared, wavelength is interpolated for the corresponding image/plot. Using estimated wavelength index Lambda (L) regression analysis using LABfit

3.6 Nutrient analysis for post october pruning

The sample collection after October pruning differs from April pruning, also the nutrient requirement also differs at this stage. After October pruning the leaf which is opposite to the basal bunch collected as a sample. As discussed in a previous chapter the samples are collected from various fields in the Theni district with preferences to collect it from fields of April pruned samples. As time of pruning differs and inputs applied by farmers also vary, the samples collected after October pruning are also with uncontrolled input i.e. information of applied fertilizers is unknown or not considered.

N requirement is high after April pruning to promote vegetative growth and less in the reproductive period. The requirement of P is high during the flowering stage, which promotes fruit bud formation, flower induction and fruit set. Requirement of K is high during the maturity stage of fruit ripening and quality. Mg requirement is required along with K for sugar translocation and fruit ripening. There is an antagonism effect of K and Mg i.e. excess K suppress Mg. October pruning stage is the stage of flowering, fruit setting and maturity stage, hence for this experiment four fertilizer parameters N,P, K and Mg are considered or estimated.

3.7 Statistical approach for nutrient estimation

This section discusses the method of regression analysis for estimation of N, P, K and Mg nutrients for October pruned data using curve fitting methods using two different softwares. The extracted parameters used in the previous chapter are also extended for this analysis.

Same algorithms discussed in the last chapter were extended to extract the individual parameters R, G, B, H, S, V, I_{kaw}, I_{pca}, etc. for October pruned data. The parameters are extracted with three different thresholds at pixel value 80, 100 and 112. Three different I_{green} indices were obtained with three different thresholds where as other parameters of RGB and HSV color planes along with L, a*, b*, Entropy, I_{ke} and Ipca are calculated at a threshold of 112. These functions and individual parameters are correlated with each nutrient to find a most suitable function for nutrient estimation.

Here in this experiment the two new indices are proposed for nutrient estimation. The first is called as Opponent HSV and second is modified version of equation 8. Opponent HSV is named because HSV values obtained, lies between the range [0 1] and the singular index calculated by subtracting the values of H, S, and V from its maximum value "1" and averaged. As H is represented in angle from 0 to 3600 which is normalized to range 0 to 1, so (1-H) will represent the opposite angle and the same is applicable for saturation and brightness represents the opponent parts. The index OHSV is calculated as shown in equation 8.

$$\text{OHSV} = \frac{(1-H)+(1-S)+(1-V)}{3} \quad \text{--------- Eqn 8}$$

Another index, which is a modified version of equation 8 is used to analyze the parameters. Nonlinear mapping of G to R and B is considered with logarithmic function instead of logsig function. It has been observed that newly derived OHSV and Log index gives better correlation with nutrients as compared to other functions if all varieties are considered. To understand the correlation of these parameters with different varieties analysis is carried out and found different results. But the combination of Thompson, Manikchaman and Sonaka (TMS) gave better correlation with these parameters and its correlation compared with all varieties in Table 2.

TMS shows the improved correlation with N, P, K, and Mg as compared to all four varieties. The parameters G, V and L show significant change in correlation with Nitrogen. Index I_{green} I_{g1}, I_{g2} and a* has shown a better correlation with P , Index I_{g1}, I_{g2} and 2G+B/(R-B) has shown significant changes in correlation for K. Parameters B, H, I_{kaw}, I_{as1}, I_{as2} and 2G+B/(R-B) shown the changes in the correlation coefficient for Mg. Newly Derived parameters OHSV and I_{Pk1} and I_{pk2} has shown marginal changes in P and K correlation where as for N and Mg it is considerable. Based on the above formula the Color features extracted for various sample plots with Laboratory values of N, P, K and Mg.

Our research focused on the correlation between nutritional levels and a variety of colour and index-based indicators. To improve accuracy and predictive power, statistical methods such as multiple linear regression and least squares curve fitting can be applied. The association between the dependent variables—nutrient levels—and the independent variables—extracted colour indices—is demonstrated by the regression model. The sum of squared discrepancies between the model's predicted and observed nutrient values is minimised by the use of least squares curve fitting.

Table 2: Color features extracted for various sample plots with Laboratory values of N, P, K and Mg

Plot	R	G	B	H	S	V	Ikaw	Ipca1	I_{green}	L1	a1	b1	Entrp	I_{g1}	I_{g2}	N	P	K	Mg
Amarbhingare	168.9833	199.8361	140.4424	0.3175	0.3851	0.8687	0.0942	56.1312	35.6756	99.6628	4.3761	63.3727	7.2049	34.0393	35.2835	1.51	0.47	3.55	0.38
Amogkhed	130.2715	182.0951	108.4684	0.3635	0.5429	0.7916	0.0969	57.9584	84.9266	96.9652	6.8742	69.4685	7.4020	82.1029	84.1467	1.62	0.22	3.90	0.57
Anildabade	126.9584	171.2335	106.9182	0.3659	0.4950	0.7522	0.0876	51.2911	86.4121	95.5192	6.8147	65.4204	7.3113	79.9224	85.3700	1.57	0.36	3.80	0.66
Awatade	135.9033	169.3926	108.7227	0.2990	0.4535	0.7490	0.1107	55.6104	59.5771	99.6740	8.2971	64.5807	7.4801	56.6050	58.3867	1.18	0.23	2.22	0.96
Chandra Mali	127.4378	172.0363	125.5871	0.3996	0.3932	0.7560	0.0094	30.4906	80.3125	97.8848	6.3463	54.8467	7.1427	76.8970	79.8030	1.62	0.29	3.50	0.59
Chunge	144.1499	181.4183	89.7677	0.2850	0.5883	0.7639	0.2412	95.6644	86.0611	93.8115	3.6139	67.7793	7.1137	82.5615	85.6908	1.79	0.28	3.30	0.05
Gangoda	112.4577	172.9204	126.9513	0.4355	0.4628	0.7739	0.0615	35.3942	65.5891	97.8433	6.1076	61.3995	7.4755	61.6050	63.3528	1.40	0.49	2.65	0.44
Ghongde P3	165.9523	189.9981	117.4766	0.3189	0.4888	0.8733	0.1725	80.4609	37.8986	99.6616	15.3349	72.7161	7.5544	35.3886	37.5304	1.57	0.41	3.17	0.35
Kbclone	100.8915	170.0722	127.3617	0.4583	0.5019	0.7618	0.1157	39.5227	68.8194	96.6863	8.8471	70.6370	7.5041	64.4177	67.8170	1.57	0.35	1.80	0.58
Kbmanik	106.08	169.2	128.7	0.46	0.46	0.76	0.0971	36.21	74.22	96.74	8.581	54.71	7.40	68.12	72.37	1.	0.	3.	0.5

	81	439	501	10	43	54		60	26	88	9	39	35	23	08	74	36	50	5
Karamclone	108.3216	160.8626	91.8543	0.3666	0.5298	0.7339	0.0834	55.0807	83.3081	95.9432	7.9592	65.9200	7.4175	75.4311	82.2238	1.01	0.41	2.15	0.46
Karamthom	105.2434	163.2538	99.8774	0.3645	0.5462	0.7388	0.0306	49.5344	73.7062	95.7570	8.3430	63.2171	7.4114	70.2214	72.5111	1.23	0.42	2.30	0.42
Anilkashid	134.5998	173.0319	108.1859	0.3321	0.4460	0.7383	0.1113	57.2038	93.9691	94.1121	4.0212	59.1862	6.8899	90.8425	93.6770	1.51	0.67	1.70	0.50
Madhukashid	123.7291	177.4256	136.1219	0.4308	0.4011	0.7726	0.0457	31.5598	73.6231	96.2351	4.4002	51.6194	7.2109	69.5679	72.6468	1.57	0.37	1.50	0.37
MaliCloneP2	131.2339	162.6920	93.0106	0.3020	0.5195	0.7599	0.1740	70.0003	72.6152	98.0043	8.7332	69.3953	7.4183	65.9051	71.3376	1.79	0.42	2.35	0.49
MalisonakaP3	121.3735	167.4783	100.2054	0.3541	0.5220	0.7605	0.0987	54.5118	73.5813	97.7234	5.2696	72.2197	7.3907	70.3459	73.1070	1.79	0.39	3.50	0.35
Moreclone	111.6624	168.3622	98.8101	0.3538	0.5556	0.7596	0.0651	49.8993	78.7935	96.6565	8.4448	71.9889	7.4686	75.2259	77.8555	2.13	0.53	2.75	0.46
Morethomson	96.1996	169.6084	132.8271	0.4477	0.4855	0.7426	0.1598	42.7883	79.7590	95.6975	4.0277	53.1747	7.1928	77.7154	79.2983	1.68	0.54	3.20	0.50
Sakhareclone	123.8549	172.2285	117.9914	0.3658	0.4189	0.7431	0.0273	37.9566	86.3677	95.6704	4.9397	55.3486	7.0757	83.8781	85.8879	1.68	0.55	1.80	0.49
Sakhareganesh	129.4741	176.4880	106.3664	0.3235	0.4587	0.7529	0.1030	57.5722	96.2423	97.2685	3.7474	61.2420	6.7849	93.5305	95.9300	1.40	0.45	2.65	0.52
MSplot1	147.0140	184.8179	115.7944	0.2983	0.4695	0.7849	0.1226	63.7702	82.6073	96.8915	6.1990	72.5752	7.0968	80.6535	82.4323	1.46	0.28	2.20	0.52

4 Discussions

4.1 Estimation of nitrogen

Nitrogen requirement as compared to after April pruning is less after October pruning and is an essential nutrient in development stage. Nitrogen has a role in flowering and have an influence on number of vine bunches and weight i.e. ultimately affects yield. Algorithm discussed in figure 1 is applied for extraction of color feature parameters using three color models – RGB, HSV and La*b*.

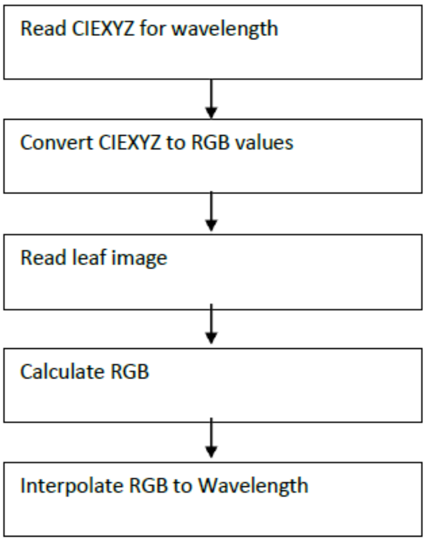

Read CIEXYZ for wavelength
Convert CIEXYZ to RGB values
Read leaf image
Calculate RGB
Interpolate RGB to Wavelength

Figure 1: Flowchart for wave length estimation from RGB color image

Color features/ parameters extracted to derive different functions. The correlation coefficient of these parameters and functions are illustrated in Table 3. Table 3 illustrates the correlation with and without illumination effect. Illumination effect is achieved by multiplying the individual parameter or the function by index I_{lx}. Using derived functions regression analysis is carried out and newly developed Opponent HSV (OHSV) function found most suitable for Nitrogen estimation. The individual name indicates a single plot of a single variety.

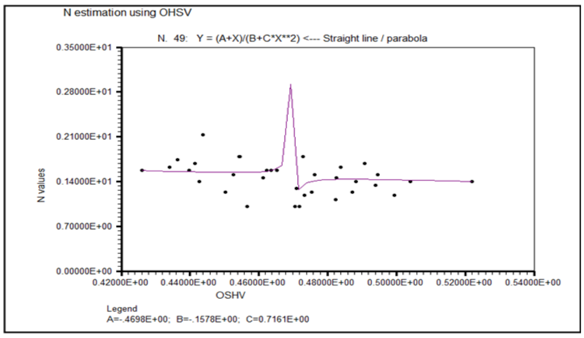
Table 3: Correlation of parameters with P

Parameter	Without I_{lx}	With I_{lx}
R	0.0648	0.2005
G	0.1975	0.2077
B	-0.0892	0.1254
H	-0.1546	0.1047
S	0.0667	0.2282
V	-0.0629	0.1971
I_{kaw}	0.1022	0.1072
I_{pca}	0.236	0.3416
Ig	0.2463	0.2477
L	-0.0687	0.1784

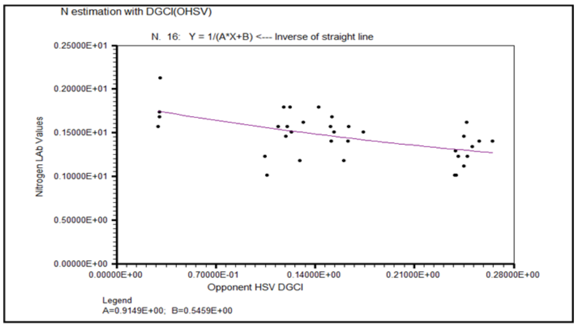
A	-0.1529	0.0745
B	-0.1303	0.1517
Entrp	-0.3035	0.1663
OHSV	0.0311	0.166
Ipk1	0.3815	0.1266

The table illustrates the sample points used in regression analysis for both OHSV function i.e. OHSV with and without illumination effect. Estimated Nitrogen for all 35 samples along with percentage error is also shown.

Figure 2(a) shows the regression for Nitrogen estimation using newly derived function OHSV without illumination effect I_{lx} and Labfit software is used for analysis of regression based on the data in Table 3. Figure 2(b) illustrates the graph of regression analysis for the N estimation using OHSV with illumination effect I_{lx}



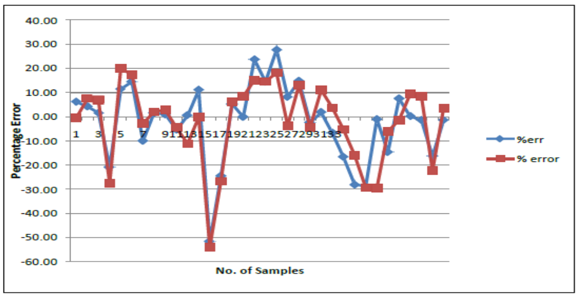
2 (a) OHSV withoutI_{lx}



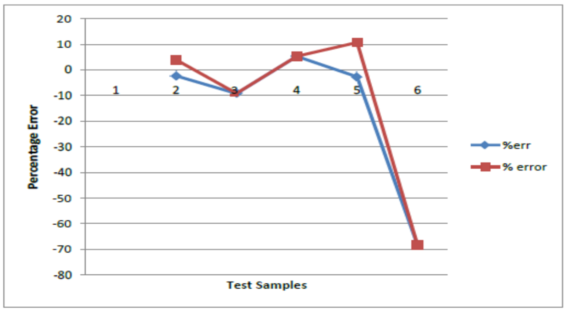
2 (b) OHSV with I_{lx}

Figure 2(a),(b): Regression for N estimation using OHSV withoutI_{lx} and withI_{lx}

Figure 3(a) and 3(b) illustrates the error graph between the values of Laboratory and estimated nitrogen respectively.



3(a) Sample Data



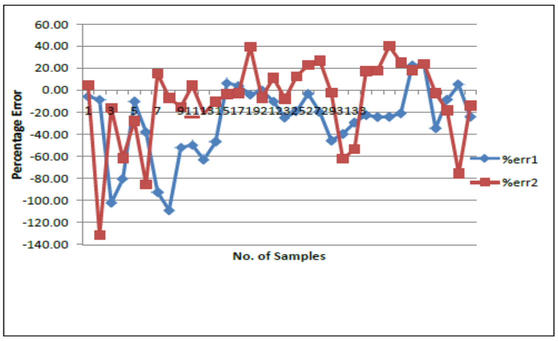
3(b) Test Data

Figure 3(a) and (b) Error graph between Lab and Estimated N for Sample data and Test Data

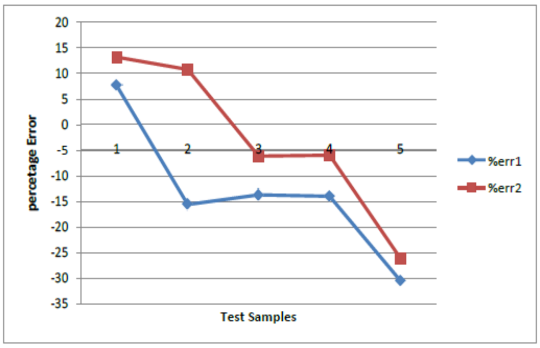
4.2 Estimation of phosphorus

Though the requirement of phosphorus is less than nitrogen and potassium, it plays an important role in seed and fruit development stimulates flowering and have a favorable effect on bud fertility that promotes the yield of grapes. Table 3 shows the correlation coefficient of various parameters obtained from October pruned images. It shows the correlation for both the cases discussed earlier, i.e. the data without effect of illumination and with illumination.

Figure 4(a) shows the regression curve for the first analysis using Matlab software and figure 4 (b) shows the regression curve obtained using the analysis by LAB fit software. Figure 4(a) and 4(b) shows the error graph between the estimated values of P by these two softwares for Sample data and test data respectively.



4(a) Regression Sample



4(b) Test Sample

Figure 4 (a) and (b) Error graph between estimated and Laboratory values of P for regression sample and test sample

4.3 Estimation of potassium (K)

Potassium is needed by grape plants for formation as sugar and starches which is essential for synthesis and cell division. Potassium plays an important role in quality of grapes by controlling its acidity and pH value of the juice. The deficiency will cause the leaves to drop prematurely, resulting in failure of fruit development in terms of color or ripening. Severe deficiency caused unevenly colored small berries and toxicity may induce deficiency of Mg which is known as antagonism effect. A deficiency of potassium leads to susceptibility of grapevine to powdery mildew.

The same algorithm that was applied to previous two nutrients is also used for estimation analysis of potassium. Table 3 shows the correlation of extracted parameters and derived functions which includes newly developed OHSV and I_{Pk1} and I_{pk2} with potassium. Also, it shows the correlation of these parameters and functions when illumination index I_{lx} is taken into consideration.

Table 4 shows the estimation of K using regression analysis. Newly developed I_{pk1} and I_{pk2} both are tested for estimation. Estimation of K using I_{pk1} is carried out by MATLAB and LAB fit whereas I_{pk2} used to estimate K with LAB fit software. These estimated values are indicated as K1, K2 and K3 respectively.

Table 4: Estimation of Potassium using logarithmic index $I_{pk}1$ &I $_{pk2}$(Sample for Regression)

Ipk1	**K**	**K1**	**%err Quad**		**K2**	**%Err**		**Ipk2**	**K**	**K3**	**%Err**
-0.42114	3.55	2.89	18.68		2.93	17.40		-0.15459	3.55	3.65	-2.83
-0.43678	3.80	2.66	29.96		2.73	28.26		-0.19307	3.80	2.55	32.78
-0.4628	2.22	2.50	-12.59		2.43	-9.37		-0.20981	2.22	2.61	-17.46
-0.46357	3.50	2.50	28.60		2.42	30.86		-0.20725	3.50	2.59	26.09
-0.39183	3.30	3.57	-8.12		3.39	-2.65		-0.15654	3.30	3.53	-7.01
-0.43751	2.65	2.65	-0.13		2.72	-2.52		-0.19166	2.65	2.56	3.39
-0.43799	2.40	2.65	-10.34		2.71	-12.96		-0.17175	2.40	2.88	-19.88
-0.43991	2.77	2.63	5.13		2.69	2.98		-0.17449	2.77	2.80	-1.19
-0.43347	1.80	2.70	-50.07		2.77	-53.77		-0.19245	1.80	2.56	-42.04
-0.44767	3.50	2.56	26.82		2.59	25.86		-0.20146	3.50	2.56	26.98
-0.43278	2.15	2.71	-26.05		2.78	-29.14		-0.20217	2.15	2.56	-18.99
-0.46233	2.35	2.50	-6.39		2.43	-3.53		-0.2172	2.35	2.69	-14.37
-0.43518	3.50	2.68	23.42		2.75	21.54		-0.19626	3.50	2.55	27.20
-0.41441	2.75	3.01	-9.58		3.03	-10.12		-0.18322	2.75	2.64	4.15
-0.43682	3.20	2.66	16.84		2.73	14.83		-0.19487	3.20	2.55	20.32
-0.44458	1.80	2.59	-43.61		2.63	-46.17		-0.19642	1.80	2.55	-41.55
-0.41587	2.65	2.98	-12.62		3.01	-13.47		-0.17549	2.65	2.78	-4.85
-0.41881	2.20	2.93	-33.12		2.96	-34.77		-0.16852	2.20	2.98	-35.47
-0.43062	2.50	2.74	-9.55		2.80	-12.18		-0.1843	2.50	2.62	-4.88
-0.41886	3.10	2.93	5.56		2.96	4.38		-0.17399	3.10	2.82	9.18
-0.40206	3.80	3.29	13.38		3.22	15.34		-0.1613	3.80	3.28	13.75
-0.39491	3.40	3.48	-2.36		3.33	1.92		-0.15442	3.40	3.66	-7.68
-0.41377	3.30	3.03	8.30		3.04	7.95		-0.16894	3.30	2.97	10.12

-0.48481	2.35	2.57	-9.37		2.21	5.85		-0.21646	2.35	2.68	-13.97
-0.45191	2.35	2.54	-7.87		2.55	-8.36		-0.19977	2.35	2.55	-8.55
-0.45421	3.10	2.52	18.59		2.52	18.68		-0.21276	3.10	2.64	14.96
-0.40327	2.30	3.26	-41.81		3.20	-39.03		-0.16766	2.30	3.01	-30.89

From Table 4 it is observed that parameters S, V, functions I_{kaw}, I_{pca}, I_{Pk1} and I_{Pk2} show the positive correlation with potassium with correlation of coefficient R ranging from 0.22 to 0.42. Newly obtained function OHSV show negative correlation with the correlation coefficient value of -0.22. If illumination effect I_{lx} taken into consideration S, V, I_{kaw} and I_{pca} show the marginal changes in coefficient where as I_{pk1} and I_{pk2} shows the significant change in the correlation, i.e. from 0.42 to 0.12. Since all other parameters show the marginal change in correlation coefficient and indices I_{pk1}, I_{pk2} shows significant change, hence I_{pk1} and I_{pk2} are considered for analysis purpose.

I_{pk1} index, which is a newly derived logarithmic index of RGB color parameters used for regression analysis and analyzed with two software MATLAB, LAB fit. Using the MATLAB curve fitting tool, the regression analysis for K, the curve is a quadratic polynomial equation having a coefficient ofdetermination0.2041 with RMSE of 0. 6204.This quadratic equation is represented as equation 9.

$$K_1 = *x^2 + *x + P_1P_2P_3 \qquad \text{----------- Eqn 9}$$

Regression analysis for estimation of K using I_{pk1} with LAB Fit tool have a coefficient of determination, R2 0.1853 and RMSE 0.6084 is represented as equation 10.

$$K_2 = \frac{A}{x2} \qquad \text{-------------------- Eqn 10}$$

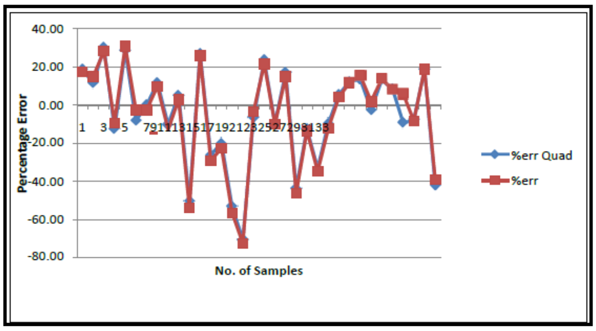
The curve shows the characteristics of the second order hyperbola. The curve has reduced chisquare of 0.31001 and error estimated using this curve have 7 points beyond the range of ±30 percent of sample data and test data. Regression analysis using I_{Pk2} which is a logarithmic function of normalized G gives the best fit curve having a coefficient of determination R2 of 0.2528 and RMSE of 0.601. The curve equation is given by equation 10

$$K3 = \frac{A}{(x + B * (\exp{(cx)})} \qquad \text{----------- Eqn 11}$$

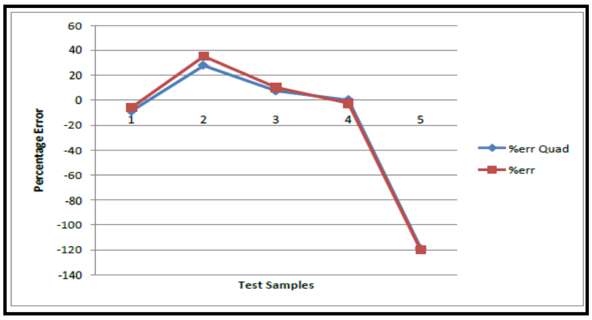
The curve is a mixture of hyperbola and exponential curves which gives the reduced chi-square of 0.361336. The estimation errors for equation 11 have 7 points beyond the range of ±30 percent of sample data and 1 in test data.

The error curves plotted for two different regression analysis are compared with each other as shown in figure 5(a) and 5(b). Figure 5 clearly shows that the error is almost same. The error in test samples shows that the last sample is beyond the boundary of ±30 percent.

The average errors for two regression equations 8 to 10 including sample and test data are -7.07, -6.84 and -7.27 respectively. Though the average error for K2 is less, but estimation samples within error boundary are less compared to K1 and K3. The temperature effect also affects the uptake of potassium.



5(a) Sample data



5 (b) Test data

Figure 5 Error between estimated values (I_{Pk1} and I_{pk2}) of K– Test data

The confidence intervals of the traditional models (CNN, SVM and Random Forest) are compared with the proposed OHSV and IPK1/IPK2) with the different parameters such as accuracy, 95% confidence interval, precision, recall, F1-Score and AUC/ROC are showcased in Table 5. From the table 5, IPK1/IPK2 inferred that the confidence interval lies in the range of 90.8% - 95.2% , which provides the better result among the compared other models. The performance comparison of traditional model with OHSV and IPK1/IPK2 are showcased in Table 6. From it, IPK1/IK2 provides optimal performance when compared with other models.

Table 5: Comparison of Traditional model with the proposed OHSV and IPK1/IPK2

Model	Accuracy	95% CI	Precision	Recall	F1-Score	AUC-ROC
CNN	92%	89.5% - 94.5%	91%	93%	92%	0.95
SVM	85%	82.1% - 87.9%	84%	86%	85%	0.89
Random Forest	88%	85.2% - 90.8%	87%	89%	88%	0.91
OHSV	90%	87.8% - 92.2%	89%	91%	90%	0.93
IPk1/IPk2	93%	90.8% - 95.2%	92%	94%	93%	0.96

Table 6: Performance Comparison of Traditional model with the proposed OHSV and IPK1/IPK2

Method	Accuracy	Precision	Recall (Sensitivity)	F1-Score	AUC-ROC	Advantages
OHSV Index	90%	89%	91%	90%	0.93	Improved color space representation for subtle differences, enhanced tumor region segmentation.
IPk1/IPk2 Indices	93%	92%	94%	93%	0.96	Granular differentiation of glioma regions, highest sensitivity for early glioma detection.
CNN (Convolutional Neural Network)	92%	91%	93%	92%	0.95	High accuracy in deep learning image classification.
SVM (Support Vector Machine)	85%	84%	86%	85%	0.89	Traditional classifier, lower performance in subtle detection cases.
Random Forest	88%	87%	89%	88%	0.91	Versatile but less effective in complex segmentation tasks.

5 Conclusion and future enhancement

Image processing technique has been proved its importance in Agriculture for detection of weeds, yield estimation, nutrient analysis specially nitrogen and chlorophyll estimation, vegetative growth, fruit sorting and pest detection. Combination of feature extractions like color, size and shape with different classifiers has added accuracy to these applications. This research is carried out to study effectiveness of Image Processing for nutrition analysis of grapes.

Nitrogen is related with the greenness of the leaf. Color image processing can be effectively used to estimate the nitrogen using RGB segmentation which leads to I_{green} index I_g. Green index I_g showed better correlation with the nitrogen values obtained by chemical analysis. This index is compared with other indices that are derived by other researchers and it shows the competitive correlation. Regression analysis using statistical technique estimates the nitrogen.

As greenness of leaf is related with the nitrogen content the wavelength estimation of green color of leaf is found to be effective method. This method is cost effective and temperature independent wavelength estimation from RGB parameters correlates to nitrogen. The wavelength and green plane ration have shown better correlation than individual parameters. Regression analysis of wavelength and chemical analysis leads to estimation of Nitrogen directly and linear equation is obtained.

In the future, this work might be expanded to include other crop varieties besides grapes. Deep learning algorithms for feature selection and classification could be implemented, and remote sensing technologies like drone-mounted cameras could be used for large-scale crop nutrient monitoring. This would automate and streamline the process of analysing nutrient content.

References

[1] L. Chvyl and C. Williams. Viti-notes. Cooperative Research Centre for Viticulture (CRCV), Australian Wine Research Institute, 2006.

[2] Edward W. Hellman. Grapevine structure and function. Oregon viticulture, Oregon State University Press, Corvallis, 5-19, 2003.

[3] VasifaA.Aglave, S. B. Patil and N.B. Sambre. Imaging Technique to Measure Leaf Area, Disease Severity and Chlorophyll Content: A Survey Paper. Journal of Computing Technologies,1(3),2012.

[4] M.M. Ali, Ahmed Al-Ani, Derek Eamus and Daniel K.Y. Tan. Leaf Nitrogen determination using Handheld Meters.16th Australian Agronomy conference-precision agriculture,2012.

[5] Paula F. Murakami, Michelle R. Turner, Abby K. van den Berg and Paul G. Schaberg. An Instructional Guide for Leaf Color Analysis using Digital Imaging Software. General Technical Report NE-327,1-37,2005.https://doi.org/10.2737/ NE-GTR-327

[6] Parviz Ahmadi Moghaddam, Mohammadali Haddad Derafshi and Vine Shirzad. Estimation of single leaf chlorophyll content in sugar beet using machine vision. Turk Journal of Agriculture and Forerestry,35(6):563-568,2011. https://doi.org/10.3906/tar-0909-393

[7] Mario Cupertino da Silva Júnior, Francisco de Assis de Carvalho Pinto, Daniel Marçalde Queiroz, Enrique Anastácio Alves, Luis Manuel Navas Gracia and Jaime Gomez Gil. Correlation between vegetation indices and nitrogen leaf content and dry matter production in Brachiaria decumbens. Image Analysis for Agricultural Products and Processes, 145-150,2006.

[8] Han Yuzhu, Wang Xiaomei and Song Shuyao, Nitrogen determination in pepper (Capsicum frutescens L.) plants by color image analysis (RGB). African Journal of Biotechnology,10(77): 17737-17741,2011.https://doi.org/10.5897/ajb11.1974

[9] Gloria F. Mata–Donjuan, Adán Mercado-Luna, Enrique Rico-García and Gilberto Herrera-Ruiz. Use of improved hue, luminance and saturation (IHLS) color space in the estimation of Nitrogen on tomato seedlings (Lycopersicon esculentum). Scientific Research and Essays,7(27):2343-2349,2012. https://doi.org/10.5897/sre11.966

[10] Fuentes, S., Collins, C. and Rogers, G. Non-destructive estimation of grapevine nutritional status using proximal sensing techniques: A review. Sensors, 19(8), 2019.

[11] Erel, R. and Yermiyahu, U. Remote Sensing-based estimation of grapevine nutrition status under field conditions. Remote Sensing, 7(4):3822-3844,2015.

[12] Léchaudel, M., Tisseyre, B., Darriet, P., Coussement, P. and Bois, B. Comparison of near-infrared spectroscopy and partial least squares regression for the rapid estimation of nitrogen, phosphorus, potassium, and magnesium contents in grapevine leaves (Vitis vinifera cv. Sauvignon). Journal of Agricultural and Food Chemistry, 61(5): 1029-1035,2013.

[13] Diago, M. P., Fernández-Novales, J. and Tardaguila, J. Assessing the nutritional status of vineyards by proximal sensing: Challenges and opportunities. Journal of the Science of Food and Agriculture,96(7);2277-2292,2016.

[14] Costa J. M., Ortuno M. F. and Chaves M. Deficit irrigation as a strategy to save water: Physiology and potential application to horticulture. Journal of Integrative Plant Biology,49(10),1421-1434,2007. https://doi.org/10.1111/j.1672-9072.2007.00556.x

A Comprehensive Overview of Federated Learning for Next-Generation Smart Agriculture: Current Trends, Challenges, and Future Directions

Belghachi Mohammed
Computer Science Department, Faculty of Science Exact, University of Tahri Mohamed Bechar, Algeria
E-mail: belghachi.mohamed@univ-bechar.dz

Review paper

Keywords: federated learning**,** smart agriculture, precision farming, machine learning, model aggregation, data heterogeneity

Received: July 23, 2024

Federated Learning (FL) is an emerging technique that offers significant potential to enhance smart agriculture by enabling collaborative model training across distributed data sources while preserving data privacy. This paper provides a comprehensive overview of the integration of FL within smart agriculture, emphasizing its role in addressing key challenges, such as data privacy, security, scalability, and data heterogeneity. The paper distinguishes itself from existing reviews by systematically analyzing FL applications in specific agricultural domains, including crop monitoring, soil health management, and livestock management. In addition, it introduces new classifications of FL use cases, focusing on privacy-preserving techniques, scalability issues, and the non-IID nature of agricultural data. Case studies from real-world implementations are used to highlight practical applications and challenges. The paper also discusses recent advances, such as the integration of FL with edge computing and the adoption of personalized federated learning. By presenting a detailed analysis of trends, challenges, and future research directions, this overview fills gaps in existing literature and provides insights into how FL can be leveraged to improve precision, productivity, and sustainability in smart agriculture. Ultimately, the findings underscore the transformative potential of FL to revolutionize data-driven agricultural decision-making and contribute to the development of resilient, privacy-conscious agricultural systems.

Povzetek: Podan je pregled metod in tehnik federativnega učenja ter njihove uporabe v pametnem kmetijstvu. Federativno učenje omogoča sodelovalno učenje modelov na porazdeljenih podatkih, kar ohranja zasebnost podatkov. Kljub prednostim, kot so izboljšana zasebnost in zmanjšanje potrebe po centraliziranem zbiranju podatkov, se pri uporabi federativnega učenja pojavljajo izzivi, kot so heterogenost podatkov, zagotavljanje varnosti in zasebnosti ter obvladovanje neodvisno in neenakomerno porazdeljenih podatkov.

1 Introduction

Smart agriculture, also known as precision agriculture, marks a significant evolution in farming practices by integrating modern information and communication technologies (ICT) to enhance efficiency, productivity, and sustainability (see figure 1). Often referred to as the *Third Green Revolution*, smart agriculture leverages technologies such as the Internet of Things (IoT), Artificial Intelligence (AI), Machine Learning (ML), and Big Data to enable real-time monitoring and control of agricultural processes. These technologies optimize the use of critical resources like water, fertilizers, and pesticides. The current technological landscape includes IoT devices and sensors that collect vast amounts of data from fields, offering insights into soil health, crop growth, and environmental conditions. Drones and satellite imagery provide advanced monitoring capabilities, enabling early detection of problems and timely corrective actions. AI and ML algorithms process this data to provide predictive analytics and decision support, helping farmers make informed decisions. Additionally, automation and robotics are becoming more prevalent in tasks like planting, weeding, and harvesting, enhancing operational efficiency.

Despite these technological advancements, the vast distribution of agricultural data and the need to maintain data privacy pose significant challenges. Traditional centralized approaches to data collection and processing require the transfer of large volumes of raw data to centralized servers, which raises concerns about data privacy, security, and scalability. Federated learning (FL), a decentralized machine learning approach, holds significant potential to address these challenges by allowing multiple clients (e.g., edge devices or local servers) to collaboratively train models while keeping data local. This paradigm shift allows models to be trained without transferring raw data, as each client shares only model updates (e.g., gradients or parameters) with a central server, which aggregates these updates to improve the global model. The process iterates until the model converges.

The benefits of federated learning are manifold. By ensuring that raw data remains on local devices, it significantly enhances data privacy and security, reducing the risks associated with data breaches and unauthorized access. Furthermore, FL minimizes communication overhead and bandwidth requirements-critical concerns when handling large datasets in agriculture.

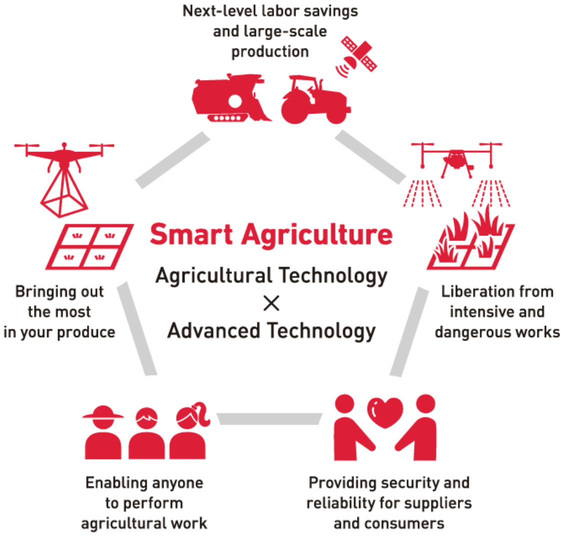

Next-level labor savings and large-scale production
Smart Agriculture
Agricultural Technology
×
Advanced Technology
Bringing out the most in your produce
Liberation from intensive and dangerous works
Enabling anyone to perform agricultural work
Providing security and reliability for suppliers and consumers

Figure 1: Smart agriculture

It also enables the use of diverse, heterogeneous data from multiple sources, improving the robustness and generalizability of trained models. However, FL presents several technical challenges, such as managing non-IID (non-Independent and Identically Distributed) data, ensuring secure and efficient communication between clients and servers, and addressing issues related to model convergence and performance. The heterogeneity of devices and data quality can also impact FL's effectiveness in agricultural environments.

The integration of federated learning into smart agriculture (see figure2) is driven by the necessity to enhance data privacy, reduce communication costs, and leverage the distributed and fragmented nature of agricultural data. In traditional centralized systems, privacy concerns and the risk of data breaches are prominent, as vast amounts of sensitive data must be transferred and stored in central servers. Federated learning mitigates these issues by keeping data local and only transmitting model updates. Furthermore, the decentralized nature of FL aligns well with the geographically dispersed and heterogeneous agricultural landscape, allowing for the development of personalized and context-aware models that cater to specific regional needs.

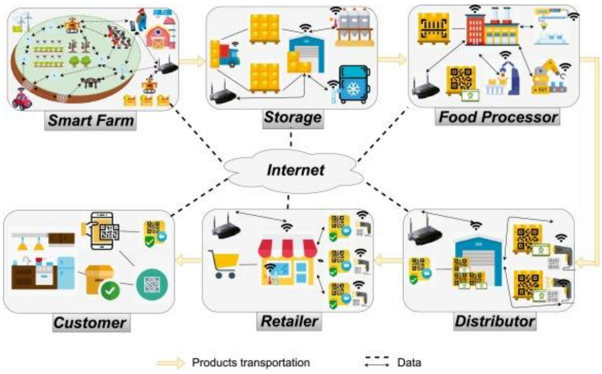


Figure 2: Federated learning for smart agriculture

The primary objective of this paper is to provide a comprehensive overview of the current state and potential of federated learning in the context of smart agriculture. Specifically, this survey reviews the existing literature on federated learning and its applications in various domains, with a focus on agriculture. It highlights the current trends and technologies in smart agriculture that can benefit from federated learning, examines the challenges and opportunities associated with its implementation, and presents case studies and real-world implementations to illustrate practical applications. In addition, the paper discusses security and privacy considerations specific to federated learning in agriculture and identifies future research directions that could further enhance the integration of FL into smart agricultural practices.

Unlike previous surveys that primarily focus on individual technical aspects of FL or specific applications in other industries, this paper aims to fill existing gaps by providing a holistic overview of FL in smart agriculture. It explores a wider range of applications, from crop monitoring and soil health management to livestock management, while also addressing emerging trends such as the integration of FL with edge computing and advanced privacy-preserving techniques. To strengthen its practical relevance, the paper includes real-world case studies that compare FL-based solutions to traditional methods, highlighting the benefits and limitations of FL in diverse agricultural contexts.

The structure of the paper is as follows: First, it explores current trends and technologies in smart agriculture, focusing on IoT, sensor networks, AI, and ML. The literature review follows, offering an analysis of existing research on federated learning in agriculture and identifying research gaps. Next, the fundamentals and techniques of federated learning are detailed, including its core concepts and methodologies. Various applications of FL in agriculture, such as crop monitoring, soil health, and livestock management, are examined. Finally, case studies and real-world implementations are reviewed to discuss practical challenges and solutions, followed by a discussion of the challenges and future directions for FL in smart agriculture. This survey aims to inform researchers, practitioners, and policymakers about the potential of federated learning to revolutionize agricultural practices.

2 Literature review

This literature review provides a comprehensive overview of the research landscape in federated learning (FL) and its integration with smart agriculture. It is organized into three main sections: (1) an overview of federated learning, (2) a review of technological advancements in smart agriculture, and (3) an exploration of how federated learning addresses specific challenges within agricultural practices. Additionally, this section includes a comparative analysis of existing overview papers on federated learning in agriculture, highlighting gaps and outlining how this paper contributes to the state-of-the-art (SOTA).

2.1 Overview of federated learning

Federated learning represents a paradigm shift in machine learning, enabling multiple participants to collaboratively train a shared model while maintaining data localization. This approach is particularly valuable in addressing data privacy concerns and minimizing the need for large-scale data transfers. The Federated Averaging (FedAvg) algorithm, introduced by McMahan et al. [1], is a cornerstone technique in FL, allowing efficient model training across numerous devices by aggregating local model updates from participants without sharing raw data.

Recent research has focused on enhancing the robustness and scalability of FL in real-world environments. A major challenge in federated learning is handling non-IID (non-Independent and Identically Distributed) data, which arises when data distribution varies across clients, as is often the case in agriculture. To address this, several studies have proposed strategies to improve the performance of FL algorithms in heterogeneous environments [2], such as modifying the aggregation process to account for variations in data distributions.

Privacy remains a critical concern in FL, especially when dealing with sensitive agricultural data. Secure aggregation and differential privacy techniques, as introduced by [3], provide strong privacy guarantees by ensuring that individual data points cannot be re-identified from the aggregated updates. These techniques are particularly useful in agricultural settings where protecting sensitive data (e.g., soil conditions or crop yields) is paramount.

In addition to privacy concerns, the emergence of personalized federated learning has gained traction. Personalized FL [4] enables global models to be fine-tuned to better align with individual clients' data, addressing the challenge of heterogeneity in local data. This method improves the relevance and accuracy of predictions, particularly in agricultural contexts where environmental conditions and farming practices can vary significantly across regions.

2.2 Research on smart agriculture technologies

Smart agriculture integrates advanced technologies to improve farming practices, efficiency, and productivity. Key among these technologies are Internet of Things (IoT) devices and sensor networks, which enable real-time monitoring of environmental conditions and facilitate data-driven decision-making. In their comprehensive review of IoT applications in precision agriculture, [5] highlighted how these technologies enhance crop yield predictions and resource management. By collecting granular data on soil moisture, temperature, and nutrient levels, IoT-enabled systems offer valuable insights that optimize agricultural practices.

Machine learning (ML) also plays a pivotal role in processing the vast amounts of data generated by IoT systems. Deep learning algorithms have been shown to significantly improve the detection of crop diseases and the prediction of crop yields. For example, [6] demonstrated that convolutional neural networks (CNNs) outperform traditional methods in identifying early symptoms of crop diseases, leading to timely interventions and improved agricultural outcomes.

Beyond deep learning, reinforcement learning (RL) has been explored as a technique for optimizing resource allocation in agriculture. For instance, [7] applied reinforcement learning to irrigation management, showing how RL models dynamically adjust water usage based on real-time environmental data, resulting in substantial water conservation.

2.3 Federated learning in smart agriculture

The integration of federated learning into smart agriculture offers several benefits, particularly in addressing challenges such as data privacy and fragmentation. In traditional centralized systems, agricultural data (e.g., from multiple farms) must be transferred to a central server, raising privacy concerns and the risk of data breaches. Federated learning mitigates these issues by allowing farms to collaboratively train models without sharing raw data, making it a powerful tool for privacy-conscious applications.

For instance, a study by [8] demonstrated the effectiveness of federated learning in collaborative crop monitoring. Their work showed that federated models improved the accuracy of disease prediction and crop management without compromising individual farm data privacy. Similarly, [9] explored the use of FL for soil health management, revealing that decentralized data from different farms could be used to develop highly accurate soil quality prediction models. By preserving data privacy, federated learning enabled better insights into soil management strategies without risking exposure of proprietary data.

However, while these studies highlight the potential of federated learning, several gaps remain in the existing literature. Most prior reviews focus narrowly on specific applications or technical aspects of FL without offering a comprehensive overview of its integration across various domains within smart agriculture. Additionally, many existing works overlook emerging trends in federated learning, such as its integration with edge computing or its application to personalized farming models that address the non-IID nature of agricultural data.

2.4 Comparative analysis of existing overview papers

To provide context for the contribution of this paper, it is essential to compare the current work with existing overviews in the field. The table below summarizes key overview papers in federated learning for smart agriculture, identifying their focus areas and highlighting the gaps they leave unaddressed.

Table 1: Comparative analysis of federated learning in smart agriculture

Overview Paper	Focus Areas	Gaps	Contribution of This Paper
[1]	General FL applications in IoT	Minimal coverage of agriculture-specific challenges, no discussion of FL integration with edge computing	Comprehensive focus on agriculture-specific use cases, emerging trends, and privacy concerns in FL
[2]	FL for data privacy in smart agriculture	Limited focus on application-specific challenges (e.g., non-IID data, regional differences)	In-depth analysis of technical challenges like data heterogeneity and model convergence in agriculture
[3]	FL for crop disease monitoring	Lacks broader application in soil and livestock management	Explores multiple agricultural domains, including soil health and livestock management
[4]	Privacy-preserving techniques in FL for agriculture	Insufficient analysis of emerging trends in FL (e.g., FL with edge computing)	Detailed exploration of emerging trends like FL-Edge integration and personalized FL for specific farm needs

As shown in the table 1, many existing reviews lack a **holistic approach** to federated learning in agriculture, focusing on either narrow technical aspects or specific applications. This paper seeks to fill these gaps by providing a more comprehensive overview of FL's integration into various agricultural domains, exploring both established and emerging trends, and presenting real-world case studies to highlight practical applications and challenges.

3 Federated learning fundamentals and techniques

Federated learning (FL) is a transformative approach in machine learning that emphasizes decentralized data processing and collaborative model training. It allows multiple clients to train a shared model locally on their data, addressing the challenges of data privacy, security, and data silos [10]. This section explores the core principles, key algorithms, and recent advancements in federated learning, including new methodologies and emerging trends that are critical for its applications in smart agriculture.

3.1 Fundamentals of federated learning

Federated learning is a decentralized machine learning paradigm designed to address privacy concerns and eliminate the need for transferring raw data between clients and central servers. It allows clients, such as IoT devices or institutions, to keep their data locally while contributing to a global model by sharing only model updates (e.g., gradients or parameters). This decentralized approach significantly enhances privacy and security while enabling collaboration across diverse data sources [11].

The FL process typically involves several steps:

- **Initialization**: A global model is initialized and distributed to all participating clients. This model, often a deep learning model, is trained locally by each client on its data [12].
- **Local training**: Each client trains the model using its local data and standard machine learning techniques (e.g., gradient descent) to update model parameters [13].
- **Aggregation**: The trained model updates from all clients are sent to a central server, where an aggregation method, such as Federated Averaging (FedAvg), combines the updates [14].
- **Update distribution**: The central server distributes the refined global model back to the clients, repeating the process until the model converges or meets a predefined performance threshold [15].
- **Model evaluation**: The global model is periodically evaluated using a validation set to ensure its generalization across different data distributions.

Federated learning addresses the need for privacy-preserving machine learning while benefiting from the diversity of data sources. However, it presents challenges in managing communication overhead, non-IID data, and ensuring efficient convergence in heterogeneous environments.

3.2 Core algorithms and techniques

3.2.1 Federated averaging (FedAvg)

Federated Averaging (FedAvg), introduced by McMahan et al. [16], is a foundational algorithm in federated learning. It aggregates local model updates from all participating clients to create a global model. The steps involved in FedAvg are:

- **Local training**: Clients train the model locally over several epochs, generating updates (e.g., gradients) based on their local data [17].
- **Aggregation**: The server aggregates these updates by calculating a weighted average, accounting for the size of each client's dataset [18].
- **Model update**: The aggregated global model is sent back to the clients for further training.

While FedAvg is effective for clients with similar data distributions, it struggles with heterogeneous environments, such as those found in agriculture, where data distributions across clients may differ significantly [19].

3.2.2 Secure aggregation

Privacy is a primary concern in federated learning, and secure aggregation techniques are essential for ensuring that individual client updates cannot be inferred from the aggregated data. **Secure aggregation** uses encryption to protect updates during transmission [20]. Key techniques include:

- **Homomorphic Encryption**: Enables computation on encrypted data, allowing the server to aggregate model updates without needing to decrypt them [21].
- **Secure Multi-Party Computation (MPC)**: MPC allows multiple clients to compute a function over their inputs while keeping those inputs private, ensuring the confidentiality of individual updates [22].

These techniques ensure privacy while preserving the utility of the aggregated global model, which is critical in privacy-sensitive applications like agriculture, where farm data can be proprietary or sensitive.

3.2.3 Differential privacy

Differential privacy offers mathematical guarantees that individual data points cannot be distinguished within a dataset, making it a crucial technique for ensuring privacy in federated learning. In FL, differential privacy is applied to model updates to prevent the extraction of sensitive information from the aggregated data [23]. Methods include:

- **Noise addition**: Random noise is added to model updates before they are sent to the server, with the amount and type of noise controlling the level of privacy [24].
- **Privacy budget**: This manages the trade-off between model accuracy and privacy by limiting the number of iterations and the amount of noise added [25].

Differential privacy is particularly valuable in smart agriculture, where protecting sensitive farm data, such as crop yields or soil conditions, is essential while still allowing for collaborative model training.

3.2.4 Personalization techniques

Personalized federated learning aims to adapt the global model to meet the specific needs of individual clients. This approach is especially useful when clients have diverse data distributions, as is common in agricultural environments where farms may differ significantly in climate, soil, and crop types [26]. Personalization techniques include:

- **Local fine-tuning**: After receiving the global model, each client can perform additional training on its local data to fine-tune the model for its specific requirements [27].
- **Meta-Learning**: This approach involves training models to learn how to adapt quickly to new tasks or data distributions, enabling clients to personalize the global model more effectively [28].

These personalization techniques improve the relevance and performance of federated learning models in heterogeneous environments like agriculture, where region-specific adaptations are necessary.

3.3 Eerging trends and techniques

3.3.1 Federated transfer learning

Federated transfer learning extends FL to scenarios where clients may have different, but related, tasks. It leverages pre-trained models from one task to improve learning on another related task, making it particularly useful in agriculture, where different farms or regions may have overlapping but distinct data needs [29]. The key aspects of federated transfer learning include:

- **Shared representation learning**: Shared model layers capture common features across tasks, enabling knowledge transfer between different agricultural tasks (e.g., pest detection in different regions) [30].
- **Task-specific fine-tuning**: Models can be customized for each client's specific task while retaining the benefits of shared learning [31].

This technique allows for improved model performance in scenarios with limited data or highly specialized tasks, such as detecting rare diseases in crops.

3.3.2 Federated multi-task learning

Federated multi-task learning allows for the simultaneous training of models on multiple tasks within a federated setting. This is highly beneficial in agriculture, where different farming operations (e.g., crop yield prediction, soil quality monitoring) may need to be optimized concurrently [32]. Techniques include:

- **Joint model training**: A single model is trained to perform multiple tasks, with task-specific heads or layers to handle different outputs [33].
- **Task aggregation**: Gradients from different tasks are combined to update the model in a way that benefits all tasks [34].

This approach improves efficiency and generalization, making it ideal for complex, multi-dimensional agricultural environments where different data sources need to be integrated.

3.3.3 **Federated** learning with edge computing

The integration of federated learning with **edge computing** is a key trend that aims to leverage computational resources at the edge of the network, such as IoT devices and edge servers. This approach reduces latency and improves the efficiency of federated learning by enabling more computation to be done locally, rather than sending data back and forth between clients and the central server [35]. Key benefits include:

- **Local computation**: Performing model training and updates locally on edge devices reduces the need for large-scale data transmission, making it particularly useful in rural or bandwidth-limited agricultural settings [36].
- **Edge aggregation**: Model updates can be aggregated at the edge before being sent to the central server, optimizing the communication process [37].

This trend is critical for smart agriculture, where IoT devices are increasingly deployed in fields for real-time monitoring of crops and environmental conditions.

4 Applications of federated learning in smart agriculture

Federated learning (FL) provides a decentralized approach to machine learning, enabling collaborative model training across multiple data sources while preserving data privacy [38]. In the context of smart agriculture, FL offers significant advantages by leveraging distributed data to enhance agricultural practices while maintaining data security. This section explores the various applications of FL in smart agriculture, focusing on its potential to improve crop management, optimize resource use, enhance livestock management, and drive innovation in precision agriculture. Additionally, this section compares the discussed applications with existing methods to highlight how FL addresses gaps left by traditional and centralized approaches.

4.1 Crop Management and Yield Prediction

4.1.1 Federated learning for crop disease prediction

Federated learning has the potential to significantly enhance **crop disease prediction** by aggregating data from various farms and research institutions without centralizing sensitive information. This approach addresses critical issues of data privacy and fragmentation that often hinder disease prediction models. Key applications include:

- **Disease detection models**: FL enables the development of robust disease detection models by training on diverse data sources, such as images from different regions. The ability to integrate data from various environmental conditions and farming practices enhances the generalization capabilities of these models, improving disease detection across different crops and regions [39].
- **Early warning systems**: Models trained through FL can analyze environmental data (e.g., temperature, humidity) and historical disease patterns to provide early warnings of potential disease outbreaks. This early detection system helps farmers take preventive measures before diseases spread, reducing crop losses and improving overall yield [40].

Compared to traditional centralized systems, FL offers a significant advantage in terms of data privacy, as raw data remains decentralized while model performance improves through collaborative training.

4.1.2 Yield prediction and optimization

Yield prediction models also benefit from federated learning by integrating diverse data from multiple farms, regions, and weather stations. Applications include:

- **Aggregated yield forecasting**: FL aggregates yield data from various sources—such as satellite imagery, weather forecasts, and historical yield records allowing for more reliable predictions. These aggregated models enable better planning and resource allocation by factoring in diverse environmental conditions [41].
- **Precision agriculture**: Federated learning improves precision agriculture by integrating data on soil conditions, crop health, and weather patterns. These models help optimize variable-rate applications of fertilizers, pesticides, and water, leading to more efficient resource use and improved crop productivity [42].

FL enables yield forecasting and optimization at a broader scale, addressing the heterogeneity of data from different regions while preserving data privacy.

4.2 Soil and irrigation management

4.2.1 Soil quality monitoring

FL enhances **soil quality monitoring** by aggregating data from a variety of sources, including sensors and research stations, allowing for a more comprehensive understanding of soil health.

- **Soil health models**: By training models on data from multiple sensors (e.g., soil moisture, nutrient sensors) and laboratory analyses, FL can provide detailed assessments of soil quality. This approach leads to better recommendations for soil management practices, helping farmers optimize soil health across regions with different soil conditions [43].

- **Nutrient management**: FL-based models analyze soil nutrient levels and crop nutrient requirements across various regions, optimizing fertilizer application. This data-driven approach reduces waste, improves soil fertility, and promotes sustainable farming practices [44].

By preserving data privacy while enabling broader collaboration, FL provides a more effective method for soil monitoring compared to traditional centralized systems that require data aggregation at a single site.

4.2.2 Smart irrigation systems

Federated learning improves the efficiency of **smart irrigation systems** by aggregating data from diverse sensors and weather stations to optimize water use:

- **Water usage optimization**: FL models aggregate data from soil moisture sensors, weather forecasts, and historical irrigation practices, leading to more efficient watering schedules. This reduces water waste and ensures that crops receive the right amount of water based on real-time conditions [45].
- **Adaptive irrigation**: FL models continuously learn from data collected across different farms, allowing them to adjust irrigation practices in real-time based on environmental changes and crop needs. This adaptability improves water use efficiency, a critical factor in regions facing water scarcity [46].

FL's ability to aggregate data while maintaining privacy makes it ideal for optimizing irrigation systems, particularly in regions where water management is a critical concern.

4.3 Livestock management

4.3.1 Health monitoring and disease management

In livestock management, FL offers significant advantages in **health monitoring and disease management:**

- **Health prediction models**: FL models trained on data from wearable sensors (e.g., heart rate monitors, activity trackers) can predict early signs of health issues in livestock. These models enable timely intervention, reducing disease spread and improving overall herd health [47].
- **Disease outbreak prediction**: By aggregating data on livestock health, environmental conditions, and disease history, FL models can predict potential disease outbreaks. This allows farmers to implement preventive measures, leading to better disease management [48].

The decentralized nature of FL helps protect sensitive livestock data while facilitating more accurate health and disease predictions across different farms.

4.3.2 Productivity optimization

FL also supports the optimization of **livestock productivity** by integrating data on feeding patterns, growth rates, and environmental conditions:

- **Feed efficiency models**: FL-based models analyze data from automated feed systems and sensors to optimize feed formulations and delivery schedules. This improves feed efficiency, reduces costs, and enhances animal growth [49].
- **Performance monitoring**: Models trained using FL can monitor livestock performance metrics (e.g., weight gain, reproductive rates) across farms, helping identify best practices and optimize management strategies [50].

FL enables collaboration among farms without compromising proprietary data, making it an ideal solution for productivity optimization in livestock management.

4.4 Precision agriculture and resource management

4.4.1 Precision planting and harvesting

FL can significantly enhance **precision planting and harvesting** practices:

- **Planting recommendations**: FL models analyze data on soil conditions, crop varieties, and weather forecasts to provide personalized planting recommendations. These models ensure crops are planted under optimal conditions, leading to better yields and more efficient resource use [51].
- **Harvest timing optimization**: By integrating data on crop growth stages and environmental conditions, FL models predict the best times for harvesting, minimizing losses and maximizing crop quality [52].

Compared to centralized systems, FL offers a more flexible and privacy-preserving approach to precision agriculture, allowing for more tailored recommendations based on local data.

4.4.2 Resource allocation and management

FL enhances **resource allocation and management** by aggregating data on resource usage (e.g., water, fertilizer) and environmental conditions:

- **Resource efficiency models**: These models analyze data on water, fertilizer, and pesticide usage to optimize resource allocation, reducing waste and ensuring that resources are used efficiently across different farms [53].
- **Environmental impact assessment**: FL models assess the environmental impact of agricultural practices by integrating data on soil health, water usage, and emissions. This helps in developing sustainable practices that mitigate environmental impacts [54].

FL supports more sustainable agricultural practices by enabling efficient resource management while preserving sensitive data.

4.5 Emerging trends and innovations

4.5.1 Integration with edge computing

The integration of FL with **edge computing** is an emerging trend that enhances real-time decision-making in agriculture:

- **Edge-based learning**: Edge devices, such as sensors and IoT devices, can perform local federated learning to analyze data in real-time. This reduces latency and improves the responsiveness of agricultural applications, such as automated irrigation and disease detection [55].
- **Collaborative edge networks**: FL models can be deployed across collaborative edge networks, allowing farms and agricultural institutions to share insights and improve model performance without compromising data privacy [56].

4.5.2 Federated transfer learning

Federated transfer learning allows knowledge transfer across related but different tasks, further enhancing FL's capabilities:

- **Knowledge transfer**: Federated transfer learning allows for the transfer of knowledge from one agricultural task or region to another. For example, a model trained to detect pests in one region can be adapted for another region with similar pest characteristics [57].
- **Adaptation to local conditions**: Transfer learning models can be fine-tuned using local data to adapt to specific agricultural conditions, improving model performance and relevance [58].

5 Case studies and real-world implementations

Federated learning (FL) is transforming smart agriculture by enabling farms to collaboratively develop advanced machine learning models without centralizing sensitive data. This approach is particularly effective in real-world scenarios where decentralized data sources, such as farms and agricultural institutions, can be utilized to solve complex agricultural challenges. By applying FL, farms can improve predictive accuracy, optimize resource management, and enhance operational efficiency, all while ensuring data privacy and security. The following case studies illustrate the practical benefits and transformative potential of FL in modern farming.

5.1 Weather-driven pest management

- **Scenario**: In the Mediterranean region, a network of farms collaborates to improve pest management using FL. Each farm collects data on local weather conditions (e.g., temperature, humidity, rainfall) and pest occurrences. This data includes detailed records of pest populations, environmental conditions, and crop types [59].
- **Federated learning application**: Each farm trains a local model to analyze the relationship between weather patterns and pest activity. FL aggregates these local models to build a comprehensive global model that predicts pest outbreaks based on weather data. For instance, the global model might uncover correlations between certain weather conditions and increased pest activity, enabling farmers to anticipate and manage pest issues proactively.
- **Benefits**: This approach allows farmers to take preventive actions, such as applying pesticides or implementing integrated pest management strategies, before pest populations reach damaging levels. The FL model is updated continuously with new weather and pest data, improving its predictive accuracy over time. This decentralized approach preserves farm-specific data while benefiting from collective insights.

5.2 Yield optimization in greenhouses

- **Scenario**: In Europe, a consortium of greenhouse operators collaborates to optimize crop yields through FL. Each greenhouse deploys sensors to monitor key environmental factors like light intensity, temperature, humidity, and CO2 levels, along with crop growth metrics such as plant height, leaf area, and flowering rates [60].
- **Federated learning application**: Local models are trained at each greenhouse using its specific environmental data. FL aggregates these local models to develop a global model that generalizes optimal growing conditions across different greenhouses. For instance, the global model may suggest ideal temperature and light settings for maximizing tomato yields in various greenhouse environments.
- **Benefits**: By pooling insights from multiple greenhouses, operators can fine-tune environmental controls without sharing proprietary data. The global model helps improve resource efficiency and crop yields, while new data continuously updates the model, ensuring its relevance and adaptability.

5.3 Climates-adaptive crop selection

- **Scenario**: In Southeast Asia, farmers collaborate to identify the most suitable crop varieties for different microclimates using FL. Each farm collects data on local climate variables (e.g., temperature ranges, rainfall, and humidity) and crop performance metrics (e.g., growth rates, yield, and disease resistance) [61].
- **Federated learning application**: Local models are trained on each farm's data to predict crop performance based on climatic conditions. FL aggregates these models into a global model that provides recommendations on the best crop varieties for specific climatic zones. For instance, the global model might suggest drought-resistant varieties for regions with low rainfall and high temperature fluctuations.
- **Benefits**: This enables farmers to make informed crop selection decisions, improving resilience to climate variability and optimizing yields. The global model is continuously refined with new farm data, allowing it to adapt to evolving climate conditions and provide more accurate recommendations.

5.4 Automated harvesting systems

- **Scenario**: In the United States, a network of farms collaborates to improve automated harvesting systems using FL. Each farm uses robotic harvesters equipped with sensors and cameras to collect data on crop quality, size, ripeness, and harvesting efficiency [62].
- **Federated learning application**: Each farm trains a local model to refine harvesting algorithms based on its specific data. FL aggregates these models into a global model that enhances the performance of robotic harvesters across different environments. For instance, the global model might improve the robot's ability to distinguish between ripe and unripe fruits, reducing waste and optimizing harvesting efficiency.
- **Benefits**: This collaborative effort optimizes the operation of robotic harvesters, reducing manual labor and minimizing crop damage. The global model is regularly updated with new data, allowing it to continuously improve in accuracy and effectiveness, particularly in identifying optimal harvesting times across diverse crop types.

5.5 Water quality monitoring in aquaculture

- **Scenario**: In Southeast Asia, aquaculture farms collaborate to monitor and manage water quality using FL. Each farm installs sensors to measure water parameters such as pH, dissolved oxygen, and nutrient levels [63].
- **Federated learning application**: Local models are trained on each farm’s water quality data to predict potential issues. FL aggregates these models into a global model that improves water management practices across different aquaculture systems. For example, the global model might recommend specific treatments or adjustments based on trends observed in collective farm data.
- **Benefits**: This approach helps maintain optimal water conditions for aquatic life, enhancing fish health and productivity. The global model is updated regularly with new data, ensuring it adapts to changing water quality challenges while safeguarding farm-specific data privacy.

5.6 Greenhouse gas emission reduction in livestock operations

- **Scenario**: In New Zealand, a network of dairy farms collaborates to reduce greenhouse gas emissions using FL. Each farm collects data on methane emissions, feed types, dietary adjustments, and animal health [64].
- **Federated learning application**: Local models are trained to predict methane emissions based on farm-specific data. FL aggregates these models to create a global model that identifies effective emission reduction strategies. For instance, the global model might suggest dietary changes that reduce methane production without compromising milk yield.
- **Benefits**: This collaborative approach helps farmers implement best practices for reducing emissions, contributing to environmental sustainability. The global model is continuously updated with new data to refine its recommendations and address emerging challenges in emission reduction.

These case studies illustrate how federated learning can address real-world challenges in smart agriculture. By facilitating collaborative, privacy-preserving model development, FL enhances decision-making, optimizes operations, and fosters innovation. The decentralized nature of FL is particularly well-suited to agriculture, where data is often fragmented across various farms and institutions, making it difficult to centralize without privacy risks.

The case studies cover a wide range of applications, from precision crop management and livestock health monitoring to soil management and automated harvesting, demonstrating the versatility of FL in agricultural contexts. In addition, FL addresses data heterogeneity, scalability, and privacy concerns more effectively than traditional centralized approaches.

Federated learning proves to be a game-changer in addressing real-world agricultural challenges. It facilitates data-driven decision-making while maintaining privacy, helping farmers manage complex agricultural operations in a decentralized yet collaborative manner. Applications such as weather-driven pest management, yield optimization, and automated harvesting underscore its potential to integrate decentralized data and provide actionable insights. As technology continues to evolve, FL will play a crucial role in advancing sustainable and resilient farming practices, paving the way for a smarter and more connected agricultural future.

6 Smart agriculture: current trends and technologies

Smart agriculture integrates advanced technologies and data-driven approaches to enhance the efficiency, productivity, and sustainability of farming practices. This section delves into the latest trends and technologies shaping the field, including Internet of Things (IoT) applications, machine learning (ML) techniques, remote sensing, and precision agriculture tools. By leveraging these technologies, smart agriculture aims to address the unique challenges of modern farming, such as data fragmentation, resource management, and the need for more sustainable practices.

6.1 Internet of things (IoT) in agriculture

The Internet of Things (IoT) is playing a transformative role in agriculture by connecting sensors, devices, and systems to collect and analyze data in real time. IoT provides the infrastructure for data-driven decision-making in farming by facilitating the continuous monitoring of key parameters that affect crop health, soil conditions, and livestock management.

6.1.1 Sensor networks

IoT sensor networks enable comprehensive, real-time monitoring of environmental conditions and crop health. These sensors track parameters such as soil moisture, temperature, humidity, and nutrient levels, providing crucial data for optimizing resource use and improving crop outcomes. For instance:

- **Soil moisture sensors**: Used to monitor soil water content, allowing for precise irrigation management. Capacitive and resistive sensors are commonly employed to reduce water usage while maximizing crop yields.
- **Climate sensors**: These sensors track environmental conditions like temperature and humidity, helping to predict weather impacts on crop growth. IoT-enabled climate sensors are vital for implementing predictive models that guide farming decisions [65].

6.1.2 Smart irrigation systems

Smart irrigation systems, powered by IoT, optimize water usage by analyzing real-time data from soil moisture sensors and weather forecasts. These systems automate irrigation schedules and adjust water delivery according to crop needs. Key innovations include:

- **Drip irrigation**: By delivering water directly to plant roots, drip irrigation minimizes evaporation and runoff. Integration with soil moisture sensors allows precise control of water application, ensuring efficient use.
- **Sprinkler systems**: These systems are equipped with weather sensors that adjust watering schedules based on precipitation and evapotranspiration rates, further enhancing water conservation efforts [66].

6.1.3 Livestock monitoring

IoT devices also improve livestock management through wearable sensors and tracking systems. These technologies monitor health metrics, activity levels, and location, enabling better management of livestock health and productivity. Examples include:

- **Wearable collars**: These devices track animal movement, health parameters, and reproductive status, providing real-time data to optimize breeding and care.
- **Automated feed systems**: IoT-enabled systems adjust feed delivery based on livestock health and consumption patterns, improving feed efficiency and reducing costs.

6.2 Machine learning and artificial intelligence

Machine learning (ML) and artificial intelligence (AI) are critical tools in smart agriculture, enabling the analysis of large datasets to optimize various aspects of farming. From crop disease detection to yield prediction, ML and AI help farmers make informed decisions that increase efficiency and productivity.

6.2.1 Crop disease detection

AI-driven image recognition algorithms are widely used to detect crop diseases and pests. These techniques allow farmers to identify problems early, reducing crop losses and enabling timely interventions. Key technologies include:

- **Deep learning**: Convolutional Neural Networks (CNNs) are employed to analyze images of crops, detecting disease symptoms from leaf patterns or discoloration.
- **Image classification**: AI models classify images into categories, such as healthy or diseased, helping farmers apply targeted treatments where needed [67].

6.2.2 Yield prediction

Machine learning models are extensively used to predict crop yields based on historical data, weather conditions, and soil health. Accurate yield prediction allows farmers to optimize planting, fertilization, and harvesting strategies. Techniques include:

- **Regression models**: Both linear and nonlinear regression models are applied to predict yields by correlating input features, such as soil conditions and climate patterns, with historical yield data.
- **Time series analysis**: ML models use time series data to forecast future yields, identifying trends and seasonal variations that affect crop output [68].

6.2.3 Precision agriculture

Precision agriculture utilizes AI to optimize farming practices at a micro-level, allowing for better resource management and improved yields. Common applications include:

- **Variable rate application**: AI systems adjust the application rates of inputs like fertilizers and pesticides, based on the spatial variability of crop needs. This technique minimizes waste and maximizes crop health.
- **Yield mapping**: Analyzing yield data across different areas of a field enables the creation of detailed maps that guide resource allocation, ensuring efficient use of inputs such as water and nutrients [69].

6.3 Remote sensing technologies

Remote sensing technologies, including satellites, drones, and aerial sensors, offer valuable insights into crop health, soil conditions, and environmental factors. These technologies provide critical data for precision agriculture, enabling farmers to monitor large areas and make data-driven decisions.

6.3.1 Satellite imagery

Satellites capture high-resolution images of agricultural fields, providing information on crop health, growth patterns, and land use. Key applications include:

- **Vegetation indices**: Metrics like the Normalized Difference Vegetation Index (NDVI) are used to assess plant health and biomass by measuring the reflectance of vegetation at different wavelengths.
- **Land cover classification**: Satellite data supports land cover classification, allowing farmers to monitor changes in land use over time and optimize their practices accordingly [70].

6.3.2 Drones

Drones equipped with multispectral and hyperspectral sensors provide detailed aerial views of fields, enabling real-time monitoring of crop health and identifying areas that require intervention.

- **Crop monitoring**: Drones capture high-resolution images that can detect plant stress, disease, or nutrient deficiencies at an early stage, helping farmers address issues before they spread.
- **Field mapping**: Drones are used to generate detailed maps that support precision agriculture by identifying areas of the field that require specific interventions, such as targeted fertilization or pest control [71].

6.3.3 Aerial and ground-based sensors

Combining aerial and ground-based sensors enhances monitoring by integrating data from multiple sources. Ground sensors validate and complement aerial data, providing a more comprehensive view of farm conditions.

- **Multispectral sensors**: These sensors measure reflectance across different wavelengths to assess crop health and identify stress factors, such as drought or disease.
- **Ground truthing**: Ground-based sensors provide on-the-ground measurements that validate the data collected from aerial platforms, ensuring the accuracy of remote sensing technologies [72].

6.4 Precision agriculture tools

Precision agriculture tools are designed to optimize field-level management based on varying field conditions, making farming more efficient and sustainable. These tools leverage spatial data to inform decisions about planting, fertilization, and harvesting.

6.4.1 GPS and GIS technologies

Global Positioning System (GPS) and Geographic Information System (GIS) technologies are fundamental to precision agriculture. These systems provide spatial data that enables:

- **Field mapping**: Detailed maps of soil properties, crop health, and yield potential allow farmers to manage their fields with precision, ensuring optimal input use and minimizing waste.
- **Automated machinery**: GPS-guided tractors and harvesters improve operational efficiency by reducing overlaps and ensuring that inputs, such as seeds and fertilizers, are applied with high accuracy [73].

6.4.2 Variable rate technology (VRT)

Variable Rate Technology (VRT) adjusts the application rates of inputs based on spatial variability within the field. This technology enables:

- **Prescription maps**: These maps indicate varying application rates based on data from soil and crop sensors, helping farmers apply inputs where they are most needed.
- **Real-Time adjustments**: VRT equipment automatically adjusts application rates in real time, responding to sensor data and optimizing resource use [74].

6.4.3 Smart greenhouses

Smart greenhouses use a combination of sensors, automation, and AI to optimize growing conditions, allowing for more efficient and sustainable crop production. Technologies include:

- **Climate control systems**: Automated systems regulate temperature, humidity, and CO2 levels, ensuring optimal growing conditions for different crops.
- **Lighting systems**: Smart lighting adjusts light intensity and duration based on plant needs and growth stages, maximizing photosynthesis and improving crop yield [75].

6.5 Emerging trends and future directions

6.5.1 Blockchain for traceability

Blockchain technology is increasingly being explored to enhance traceability in the agricultural supply chain. By creating immutable records of every stage of food production, blockchain ensures:

- **Transparency**: Consumers and stakeholders can access verified records of agricultural practices, ensuring food safety and authenticity.
- **Security**: Blockchain protects against fraud and ensures the integrity of data across the supply chain, from farm to table [76].

6.5.2 Autonomous machinery

Autonomous machinery, including self-driving tractors and robotic harvesters, is becoming more prevalent in agriculture. These machines:

- **Increase efficiency**: Autonomous systems perform tasks such as planting, weeding, and harvesting with minimal human intervention, reducing labor costs and enhancing precision.
- **Reduce labor costs**: By automating repetitive tasks, autonomous machinery decreases the reliance on manual labor while ensuring consistent performance [77].

6.5.3 Advanced data analytics

Advanced data analytics, powered by big data and predictive analytics, is revolutionizing decision-making in agriculture. Techniques include:

- **Predictive modeling**: Algorithms forecast crop yields, disease outbreaks, and market trends, helping farmers make informed decisions that optimize production and minimize risk.
- **Data integration**: Integrating data from various sources such as sensors, satellites, and historical records improves the accuracy and scope of predictive models, enabling more efficient farm management [78].

Smart agriculture is rapidly evolving, driven by advances in IoT, machine learning, remote sensing, and precision agriculture tools. These technologies enable real-time monitoring, predictive analytics, and data-driven decision-making, leading to more efficient, sustainable farming practices. Emerging trends such as blockchain, autonomous machinery, and advanced data analytics further promise to transform the agricultural landscape, making it more connected, transparent, and resilient.

7 Challenges and future directions

Federated learning (FL) offers immense potential for advancing smart agriculture by enabling decentralized model training while preserving data privacy. However, several challenges need to be addressed for FL to achieve widespread adoption in agriculture. This section explores these challenges in detail and outlines potential future directions that can further enhance the role of FL in the agricultural sector.

7.1 Challenges in federated learning for smart agriculture

7.1.1 Data heterogeneity

Challenge: Agricultural data is highly heterogeneous across different farms and regions. This diversity includes variations in data quality, format, type, and scale, making it difficult to create a unified model that generalizes effectively across diverse environments [79].

Impact:

- **Model performance**: Heterogeneous data can degrade model performance by making it harder for FL models to generalize across varying distributions.
- **Training inefficiencies**: Models may require more iterations and extensive fine-tuning to achieve acceptable performance, resulting in longer training times and higher resource consumption.

Solutions:

- **Advanced aggregation techniques**: New aggregation methods, such as robust federated averaging, personalized FL, or meta-learning approaches, can help models better adapt to diverse data distributions.
- **Data preprocessing and normalization**: Standardizing and normalizing data before training can reduce the impact of heterogeneity, improving model convergence and accuracy across diverse data sources.

7.1.2 Communication overhead

Challenge: FL involves frequent communication between local nodes and central servers, which can result in high communication overhead. This is particularly problematic when working with large models or a high number of participants [80].

Impact:

- **Network congestion**: The strain on network resources can cause delays, reducing the efficiency of model updates and making the process more costly in areas with poor connectivity.

- **Cost**: High communication demands lead to increased operational costs, especially in rural or remote areas where connectivity is limited or expensive.

Solutions:

- **Efficient communication protocols**: Techniques such as model compression, sparse updates, and quantization can reduce the volume of data exchanged during model training.
- **Adaptive communication strategies**: Implementing adaptive strategies that adjust the frequency and volume of updates based on network conditions and model performance can optimize communication and reduce overhead.

7.1.3 Privacy and security

Challenge: Ensuring the privacy and security of data and model updates remains a critical concern. Although techniques like differential privacy and secure aggregation are used, FL is still vulnerable to risks such as model inversion attacks and data leakage [81].

Impact:

- **Data breaches**: Weak privacy protection can expose sensitive agricultural data, such as crop yields, soil conditions, or farming practices, potentially leading to competitive disadvantages or violations of regulatory requirements.
- **Trust issues**: Inadequate security can erode trust between participants, limiting collaboration and reducing the overall effectiveness of FL systems.

Solutions:

- **Enhanced privacy techniques**: Research into advanced privacy-preserving technologies, such as secure multi-party computation (SMPC) and fully homomorphic encryption (FHE), can provide stronger guarantees of data confidentiality.
- **Comprehensive security frameworks**: Developing robust, end-to-end security frameworks that address all attack vectors and ensure data and model integrity during updates is crucial for protecting sensitive agricultural data.

7.1.4 Computational and resource constraints

Challenge: Many agricultural environments, particularly small-scale farms, may not have the computational resources necessary for running FL algorithms, which require significant processing power and memory [82].

Impact:

- **Limited adoption**: Resource constraints can limit the adoption of FL technologies among resource-poor farms that lack the necessary hardware or network infrastructure.
- **Performance bottlenecks**: Insufficient computational power can slow down training and lead to suboptimal model performance, reducing the benefits of FL for small farms.

Solutions:

- **Edge computing**: By integrating FL with edge computing, computational tasks can be offloaded to edge devices or servers, reducing the burden on individual farms while enabling real-time data processing.
- **Resource-efficient algorithms**: Designing lightweight FL algorithms that can run efficiently on low-power devices will make FL more accessible to a wider range of agricultural stakeholders.

7.1.5 Scalability

Challenge: Scaling FL systems to handle large datasets and increasing numbers of participants presents significant challenges. As the system scales, the complexity of data aggregation, model synchronization, and system management increases [83].

Impact:

- **System complexity**: As more farms and devices join the FL network, the complexity of coordinating updates and managing communication grows, which can lead to inefficiencies.
- **Performance degradation**: With more participants, the system may experience slower convergence times and reduced overall performance if not properly managed.

Solutions:

- **Scalable architectures**: Implementing scalable FL architectures that efficiently manage large numbers of participants and datasets through techniques like hierarchical FL or distributed computing can improve system performance.
- **Cloud integration**: Using cloud-based solutions to handle scalability issues will allow for more flexible data aggregation, processing, and storage, ensuring efficient system management at scale.

7.2 Future directions

7.2.1 Integration with IoT and sensor technologies

Future direction: Integrating FL with IoT and advanced sensor networks can improve real-time data collection and model training in smart agriculture [84].

Potential benefits :

- **Real-Time data**: IoT sensors provide continuous, real-time data that enhances the accuracy and relevance of FL models.
- **Enhanced precision**: The combination of FL and sensor data can significantly improve precision in applications like precision farming, irrigation management, and livestock monitoring.

Approaches:

- **IoT-Enabled FL**: Developing FL frameworks tailored for IoT environments, which consider data from diverse sensors and optimize model training for real-time analytics.
- **Real-Time model updates**: Implementing real-time analytics and model updates to leverage the continuous data streams from IoT devices in agriculture.

7.2.2 Advanced privacy-preserving techniques

Future direction: Researching and deploying next-generation privacy-preserving techniques will strengthen data confidentiality and security in FL applications [85].

Potential benefits:

- **Stronger privacy guarantees**: Improved privacy techniques will address current security vulnerabilities, such as model inversion attacks and data inference risks.
- **Increased trust**: Stronger privacy mechanisms will foster greater collaboration and trust among participants, encouraging broader adoption of FL.

Approaches:

- **Next-generation encryption**: Implementing advanced encryption methods like fully homomorphic encryption (FHE) and secure enclave technologies for enhanced data security.
- **Privacy-enhancing technologies**: Integrating emerging privacy-enhancing technologies, such as differential privacy in conjunction with secure hardware, to create more robust FL systems.

7.2.3 Cross-domain federated learning

Future direction: Investigating cross-domain FL, where models trained on data from different agricultural domains or regions can be integrated, offers significant potential [86].

Potential benefits:

- **Comprehensive insights**: Cross-domain models provide broader insights by leveraging diverse datasets from different agricultural practices, environments, and crops.
- **Enhanced generalization**: Models trained across multiple domains have better generalization capabilities, offering more robust predictions.

Approaches:

- **Domain adaptation**: Developing domain adaptation techniques that allow FL models to bridge differences between data from different agricultural domains.
- **Federated transfer learning**: Implementing federated transfer learning to transfer knowledge across agricultural tasks, allowing for improved model adaptability and performance.

7.2.4 Collaborative research and standardization

Future direction: Promoting collaborative research and establishing standards will accelerate the development and adoption of FL technologies in agriculture [87].

Potential benefits:

- **Innovation and knowledge sharing**: Collaborative research will drive innovation and facilitate knowledge sharing across academia, industry, and agricultural stakeholders.
- **Consistency and interoperability**: Standardization will ensure consistency and interoperability across FL systems, making adoption easier and more seamless.

Approaches:

- **Industry partnerships**: Forming partnerships between academic institutions, industry players, and agricultural practitioners to drive innovation and address real-world challenges.
- **Standards development**: Collaborating on the development of standardized protocols and best practices for implementing FL in agriculture.

7.2.5 User-friendly tools and interfaces

Future Direction: Developing accessible tools and interfaces will make FL more accessible to non-expert users, particularly small-scale farmers [88].

Potential benefits:

- **Wider adoption**: User-friendly tools will encourage adoption of FL technologies among farmers and agricultural workers with varying technical expertise.
- **Simplified implementation**: Simplified interfaces will lower the technical barriers to implementing and managing FL systems, ensuring broader participation.

Approaches:

- **Accessible platforms**: Creating FL platforms with intuitive interfaces and workflows that simplify the implementation process for non-experts.
- **Training and support**: Providing training programs and support resources to help farmers and practitioners effectively use FL technologies.

Federated learning presents numerous opportunities for advancing smart agriculture, but it faces challenges related to data heterogeneity, communication overhead, privacy, computational constraints, and scalability. By addressing these challenges through innovative solutions, such as advanced privacy techniques, scalable architectures, and IoT integration, FL can revolutionize agricultural practices. Future directions—such as cross-domain learning, collaborative research, and user-friendly tools will be essential in driving widespread adoption and enabling FL to fully realize its potential in agriculture.

8 Conclusion

Federated learning (FL) is poised to revolutionize data-driven decision-making in smart agriculture by enabling collaborative model training across decentralized data sources while preserving the privacy and security of sensitive agricultural information. This transformative approach addresses some of the most pressing challenges in agricultural data management, including data privacy, scalability, and the need for efficient, adaptable solutions in diverse farming environments.

This paper has provided an in-depth exploration of the fundamental principles of federated learning and its applications in smart agriculture. We discussed how FL can enhance various aspects of modern agriculture, including precision farming, livestock management, soil and irrigation optimization, and precision breeding. These applications benefit significantly from FL's ability to derive data-driven insights without centralizing sensitive information, allowing farmers and agricultural stakeholders to collaborate more effectively while safeguarding proprietary data.

The literature review underscored the growing research interest in federated learning within agriculture, highlighting its potential to enhance agricultural productivity and sustainability. It also showcased the advantages of privacy-preserving data sharing, particularly in a sector where privacy and data ownership are paramount. We explored key FL techniques, such as model aggregation, communication protocols, and privacy-preserving methods, emphasizing their relevance to overcoming the unique challenges faced by the agricultural industry.

Through case studies and real-world examples, we demonstrated how federated learning can drive innovation in key areas like yield prediction, livestock health monitoring, soil and irrigation management, and precision breeding. These case studies illustrated the potential of FL to improve decision-making, optimize resource use, and boost productivity—all while maintaining data privacy and minimizing the risk of data breaches. By decentralizing model training and enabling collaboration across diverse data sources, FL offers a practical and scalable solution for modern agriculture.

However, the implementation of federated learning in agriculture is not without challenges. Data heterogeneity, communication overhead, privacy and security concerns, computational limitations, and scalability issues pose significant obstacles to widespread adoption. Addressing these challenges will require ongoing research and development in areas such as advanced aggregation techniques, communication-efficient protocols, enhanced privacy measures, and resource-efficient algorithms.

Looking to the future, several promising directions have emerged for advancing federated learning in agriculture. These include deeper integration with IoT and sensor networks for real-time data collection, the exploration of next-generation privacy-preserving techniques, the development of cross-domain federated learning models, and fostering collaborative research efforts across the agricultural sector. Moreover, creating user-friendly tools and platforms will be essential to ensure that FL technologies are accessible to farmers and agricultural stakeholders, particularly those with limited technical expertise.

In conclusion, federated learning offers a powerful paradigm for advancing smart agriculture. Its ability to preserve data privacy, enhance model accuracy, and optimize resource utilization aligns with the goals of modern, sustainable agricultural practices. As research in this field continues to evolve and technology advances, federated learning will play an increasingly critical role in improving agricultural productivity, sustainability, and resilience in the face of global challenges such as climate change, resource scarcity, and food security. By addressing the existing challenges and embracing future innovations, federated learning can help shape the future of agriculture, making it more efficient, secure, and data-driven.

References

[1] Shahbazi, S., & Byun, Y.-C. (2021). A Deep Learning Approach for Smart Agriculture Using Climate Data. *IEEE Access*, 9, 94722-94734. DOI: 10.1109/ACCESS.2021.3088651

[2] Patil, K., & Kale, S. (2021). Precision Agriculture Using IoT and AI: A Systematic Review. *Journal of Physics: Conference Series*, 1913(1), 012006. DOI: 10.1088/1742-6596/1913/1/012006

[3] Talaviya, T., Shah, M., Patel, N., Yagnik, H., & Shah, M. (2020). Implementation of Artificial Intelligence in Agriculture for Optimisation of Irrigation and Pesticide Control. *Artificial Intelligence in Agriculture*, 4, 58-73. DOI: 10.1016/j.aiia.2020.04.002

[4] Garcia, C., Oliver, M. N., Gil, J., & Domenech, S. (2021). An IoT-Based System for Precision Agriculture for Optimal Application of Treatments. *Sensors*, 21(9), 2974. DOI: 10.3390/s21092974

[5] Ramos, M. M., Borges, L. M., & Velez, F. J. (2021). Smart Agriculture IoT with LoRaWAN: A Survey. *Sensors*, 21(12), 4283. DOI: 10.3390/s21124283

[6] Tripathy, A., & Kumar, S. (2021). IoT-Based Smart Farming: Leveraging Cloud, Machine Learning, and Blockchain. *Internet of Things*, 14, 100378. DOI: 10.1016/j.iot.2021.100378

[7] Kshetri, N. (2020). Blockchain and AI Technology: Potential Applications in Smart Agriculture. *Journal of Industrial Information Integration*, 17, 100132. DOI: 10.1016/j.jii.2020.100132

[8] Clemente, L., Perez-Martinez, F. J., & Urquiza-Fuentes, J. (2022). IoT-Enabled Crop Monitoring and Smart Farming Using Multi-Sensor and Edge Computing Techniques. *Sensors*, 22(3), 1143. DOI: 10.3390/s22031143

[9] Chen, X., & Yu, Z. (2021). IoT-Based Smart Irrigation for Agricultural Water Management.

Journal of Cleaner Production, 280, 124197. DOI: 10.1016/j.jclepro.2020.124197

[10] Abdullah, M., Nandy, M., & Islam, M. (2021). Machine Learning-Based Crop Yield Prediction and Smart Irrigation. *Computers and Electronics in Agriculture*, 190, 106477. DOI: 10.1016/j.compag.2021.106477

[11] Feng, Q., Liu, J., & Gong, J. (2021). UAV Remote Sensing for Precision Agriculture: A Comparison between Different Cameras. *Remote Sensing*, 12(1), 14. DOI: 10.3390/rs12010014

[12] Anastasiou, A., & Makris, I. (2022). Wireless Sensor Networks in Precision Agriculture: A Review of Applications, Challenges, and Future Directions. *Wireless Communications and Mobile Computing*, 2022, 9317642. DOI: 10.1155/2022/9317642

[13] Sharef, B. T. (2022). The Usage of Internet of Things in Agriculture: The Role of Size and Perceived Value. *Informatica Journal*.

[14] Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Processing Magazine*, 37(3), 50-60. DOI: 10.1109/MSP.2020.2975749

[15] Lu, Y., Li, Y., Wang, X., & Zhuang, Y. (2021). Federated Learning for Privacy-Preserving Agricultural Data Analysis: Challenges and Solutions. *Sensors*, 21(6), 2034. DOI: 10.3390/s21062034

[16] Xu, J., Wang, H., & Ding, Y. (2022). Federated Learning-Based Crop Disease Diagnosis Using IoT Data. *Computers and Electronics in Agriculture*, 192, 106617. DOI: 10.1016/j.compag.2022.106617

[17] Hao, M., Li, H., Huang, C., Zhang, X., & Shi, Y. (2020). Towards Efficient and Privacy-Preserving Federated Deep Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 31(4), 1428-1442. DOI: 10.1109/TNNLS.2020.2967406

[18] Cheng, Y., Zhang, Y., Chen, T., & Zhou, Z. (2021). Secure Federated Transfer Learning for Privacy-Preserving Data Sharing. *IEEE Transactions on Big Data*. DOI: 10.1109/TBDATA.2021.3053896

[19] Nguyen, D. C., Pathirana, P. N., Ding, M., & Seneviratne, A. (2021). Federated Learning for Internet of Things: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials*, 23(3), 1622-1658. DOI: 10.1109/COMST.2021.3075439

[20] Zhou, X., Li, Y., & Wang, J. (2021). Federated Learning with Differential Privacy for IoT in Smart Agriculture. *IEEE Internet of Things Journal*, 8(9), 6722-6733. DOI: 10.1109/JIOT.2021.3063772

[21] Khan, L. U., Saad, W., Han, Z., & Bennis, M. (2021). Federated Learning for Distributed Smart Farming: Challenges and Opportunities. *IEEE Network*, 35(4), 183-189. DOI: 10.1109/MNET.011.2000604

[22] Zhang, T., He, X., & Zhou, Y. (2022). Distributed Federated Learning for Precision Agriculture with IoT Devices. *Sensors*, 22(4), 1222. DOI: 10.3390/s22041222

[23] Yin, Y., Wei, W., Wang, H., & Yin, Y. (2021). A Review on Federated Learning Applications in Smart Agriculture. *Computers and Electronics in Agriculture*, 190, 106502. DOI: 10.1016/j.compag.2021.106502

[24] Mohammadi, M., Al-Fuqaha, A., Sorour, S., & Guizani, M. (2020). Deep Learning for IoT Big Data and Streaming Analytics: A Survey. *IEEE Communications Surveys & Tutorials*, 22(4), 2491-2521. DOI: 10.1109/COMST.2020.2985893

[25] Chen, L., Zhang, Z., & Peng, W. (2022). Federated Learning with Blockchain for Secure IoT-Enabled Smart Agriculture. *IEEE Internet of Things Journal*, 9(7), 5146-5157. DOI: 10.1109/JIOT.2021.3102726

[26] Yu, T., Jiang, L., Li, Z., & Liu, Y. (2020). Privacy-Aware Federated Learning for Distributed Data Mining in Smart Farming. *Journal of Agricultural Informatics*, 11(3), 23-33. DOI: 10.17700/jai.2020.11.3.570

[27] Kairouz, P., McMahan, H. B., Alistarh, D., et al. (2021). Advances and Open Problems in Federated Learning. *Foundations and Trends® in Machine Learning*, 14(1–2), 1-210. DOI: 10.1561/2200000083

[28] Shahbazi, S., & Byun, Y.-C. (2021). A Deep Learning Approach for Smart Agriculture Using Climate Data. *IEEE Access*, 9, 94722-94734. DOI: 10.1109/ACCESS.2021.3088651

[29] Patil, K., & Kale, S. (2021). Precision Agriculture Using IoT and AI: A Systematic Review. *Journal of Physics: Conference Series*, 1913(1), 012006. DOI: 10.1088/1742-6596/1913/1/012006

[30] Talaviya, T., Shah, M., Patel, N., Yagnik, H., & Shah, M. (2020). Implementation of Artificial Intelligence in Agriculture for Optimisation of Irrigation and Pesticide Control. *Artificial Intelligence in Agriculture*, 4, 58-73. DOI: 10.1016/j.aiia.2020.04.002

[31] Garcia, C., Oliver, M. N., Gil, J., & Domenech, S. (2021). An IoT-Based System for Precision Agriculture for Optimal Application of Treatments. *Sensors*, 21(9), 2974. DOI: 10.3390/s21092974

[32] Ramos, M. M., Borges, L. M., & Velez, F. J. (2021). Smart Agriculture IoT with LoRaWAN: A Survey. *Sensors*, 21(12), 4283. DOI: 10.3390/s21124283

[33] Tripathy, A., & Kumar, S. (2021). IoT-Based Smart Farming: Leveraging Cloud, Machine

Learning, and Blockchain. *Internet of Things*, 14, 100378. DOI: 10.1016/j.iot.2021.100378

[34] Kshetri, N. (2020). Blockchain and AI Technology: Potential Applications in Smart Agriculture. *Journal of Industrial Information Integration*, 17, 100132. DOI: 10.1016/j.jii.2020.100132

[35] Clemente, L., Perez-Martinez, F. J., & Urquiza-Fuentes, J. (2022). IoT-Enabled Crop Monitoring and Smart Farming Using Multi-Sensor and Edge Computing Techniques. *Sensors*, 22(3), 1143. DOI: 10.3390/s22031143

[36] Chen, X., & Yu, Z. (2021). IoT-Based Smart Irrigation for Agricultural Water Management. *Journal of Cleaner Production*, 280, 124197. DOI: 10.1016/j.jclepro.2020.124197

[37] Abdullah, M., Nandy, M., & Islam, M. (2021). Machine Learning-Based Crop Yield Prediction and Smart Irrigation. *Computers and Electronics in Agriculture*, 190, 106477. DOI : 10.1016/j.compag.2021.106477

[38] Feng, Q., Liu, J., & Gong, J. (2021). UAV Remote Sensing for Precision Agriculture: A Comparison between Different Cameras. *Remote Sensing*, 12(1), 14. DOI : 10.3390/rs12010014

[39] Anastasiou, A., & Makris, I. (2022). Wireless Sensor Networks in Precision Agriculture: A Review of Applications, Challenges, and Future Directions. *Wireless Communications and Mobile Computing*, 2022, 9317642. DOI : 10.1155/2022/9317642

[40] Kairouz, P., McMahan, H. B., Alistarh, D., et al. (2021). Advances and Open Problems in Federated Learning. *Foundations and Trends® in Machine Learning*, 14(1–2), 1-210. DOI: 10.1561/2200000083

[41] Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Processing Magazine*, 37(3), 50-60. DOI: 10.1109/MSP.2020.2975749

[42] Lu, Y., Li, Y., Wang, X., & Zhuang, Y. (2021). Federated Learning for Privacy-Preserving Agricultural Data Analysis: Challenges and Solutions. *Sensors*, 21(6), 2034. DOI: 10.3390/s21062034

[43] Xu, J., Wang, H., & Ding, Y. (2022). Federated Learning-Based Crop Disease Diagnosis Using IoT Data. *Computers and Electronics in Agriculture*, 192, 106617. DOI: 10.1016/j.compag.2022.106617

[44] Hao, M., Li, H., Huang, C., Zhang, X., & Shi, Y. (2020). Towards Efficient and Privacy-Preserving Federated Deep Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 31(4), 1428-1442. DOI: 10.1109/TNNLS.2020.2967406

[45] Cheng, Y., Zhang, Y., Chen, T., & Zhou, Z. (2021). Secure Federated Transfer Learning for Privacy-Preserving Data Sharing. *IEEE Transactions on Big Data*. DOI: 10.1109/TBDATA.2021.3053896

[46] Nguyen, D. C., Pathirana, P. N., Ding, M., & Seneviratne, A. (2021). Federated Learning for Internet of Things: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials*, 23(3), 1622-1658. DOI: 10.1109/COMST.2021.3075439

[47] Zhou, X., Li, Y., & Wang, J. (2021). Federated Learning with Differential Privacy for IoT in Smart Agriculture. *IEEE Internet of Things Journal*, 8(9), 6722-6733. DOI: 10.1109/JIOT.2021.3063772

[48] Khan, L. U., Saad, W., Han, Z., & Bennis, M. (2021). Federated Learning for Distributed Smart Farming: Challenges and Opportunities. *IEEE Network*, 35(4), 183-189. DOI : 10.1109/MNET.011.2000604

[49] Zhang, T., He, X., & Zhou, Y. (2022). Distributed Federated Learning for Precision Agriculture with IoT Devices. *Sensors*, 22(4), 1222. DOI : 10.3390/s22041222

[50] Yin, Y., Wei, W., Wang, H., & Yin, Y. (2021). A Review on Federated Learning Applications in Smart Agriculture. *Computers and Electronics in Agriculture*, 190, 106502. DOI : 10.1016/j.compag.2021.106502

[51] Mohammadi, M., Al-Fuqaha, A., Sorour, S., & Guizani, M. (2020). Deep Learning for IoT Big Data and Streaming Analytics: A Survey. *IEEE Communications Surveys & Tutorials*, 22(4), 2491-2521. DOI: 10.1109/COMST.2020.2985893

[52] Chen, L., Zhang, Z., & Peng, W. (2022). Federated Learning with Blockchain for Secure IoT-Enabled Smart Agriculture. *IEEE Internet of Things Journal*, 9(7), 5146-5157. DOI: 10.1109/JIOT.2021.3102726

[53] Yu, T., Jiang, L., Li, Z., & Liu, Y. (2020). Privacy-Aware Federated Learning for Distributed Data Mining in Smart Farming. *Journal of Agricultural Informatics*, 11(3), 23-33. DOI: 10.17700/jai.2020.11.3.570

[54] Shahbazi, S., & Byun, Y.-C. (2021). A Deep Learning Approach for Smart Agriculture Using Climate Data. *IEEE Access*, 9, 94722-94734. DOI: 10.1109/ACCESS.2021.3088651

[55] Patil, K., & Kale, S. (2021). Precision Agriculture Using IoT and AI: A Systematic Review. *Journal of Physics: Conference Series*, 1913(1), 012006. DOI: 10.1088/1742-6596/1913/1/012006

[56] Talaviya, T., Shah, M., Patel, N., Yagnik, H., & Shah, M. (2020). Implementation of Artificial Intelligence in Agriculture for Optimisation of Irrigation and Pesticide Control. *Artificial Intelligence in Agriculture*, 4, 58-73. DOI: 10.1016/j.aiia.2020.04.002

[57] Garcia, C., Oliver, M. N., Gil, J., & Domenech, S. (2021). An IoT-Based System for Precision Agriculture for Optimal Application of Treatments. *Sensors*, 21(9), 2974. DOI: 10.3390/s21092974

[58] Ramos, M. M., Borges, L. M., & Velez, F. J. (2021). Smart Agriculture IoT with LoRaWAN: A Survey. *Sensors*, 21(12), 4283. DOI: 10.3390/s21124283

[59] Tripathy, A., & Kumar, S. (2021). IoT-Based Smart Farming: Leveraging Cloud, Machine Learning, and Blockchain. *Internet of Things*, 14, 100378. DOI: 10.1016/j.iot.2021.100378

[60] Kshetri, N. (2020). Blockchain and AI Technology: Potential Applications in Smart Agriculture. *Journal of Industrial Information Integration*, 17, 100132. DOI: 10.1016/j.jii.2020.100132

[61] Clemente, L., Perez-Martinez, F. J., & Urquiza-Fuentes, J. (2022). IoT-Enabled Crop Monitoring and Smart Farming Using Multi-Sensor and Edge Computing Techniques. *Sensors*, 22(3), 1143. DOI: 10.3390/s22031143

[62] Chen, X., & Yu, Z. (2021). IoT-Based Smart Irrigation for Agricultural Water Management. *Journal of Cleaner Production*, 280, 124197. DOI: 10.1016/j.jclepro.2020.124197

[63] Abdullah, M., Nandy, M., & Islam, M. (2021). Machine Learning-Based Crop Yield Prediction and Smart Irrigation. *Computers and Electronics in Agriculture*, 190, 106477. DOI: 10.1016/j.compag.2021.106477

[64] Feng, Q., Liu, J., & Gong, J. (2021). UAV Remote Sensing for Precision Agriculture: A Comparison between Different Cameras. *Remote Sensing*, 12(1), 14. DOI: 10.3390/rs12010014

[65] Anastasiou, A., & Makris, I. (2022). Wireless Sensor Networks in Precision Agriculture: A Review of Applications, Challenges, and Future Directions. *Wireless Communications and Mobile Computing*, 2022, 9317642. DOI: 10.1155/2022/9317642

[66] Kairouz, P., McMahan, H. B., Alistarh, D., et al. (2021). Advances and Open Problems in Federated Learning. *Foundations and Trends® in Machine Learning*, 14(1–2), 1-210. DOI: 10.1561/2200000083

[67] Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Processing Magazine*, 37(3), 50-60. DOI: 10.1109/MSP.2020.2975749

[68] Lu, Y., Li, Y., Wang, X., & Zhuang, Y. (2021). Federated Learning for Privacy-Preserving Agricultural Data Analysis: Challenges and Solutions. *Sensors*, 21(6), 2034. DOI: 10.3390/s21062034

[69] Xu, J., Wang, H., & Ding, Y. (2022). Federated Learning-Based Crop Disease Diagnosis Using IoT Data. *Computers and Electronics in Agriculture*, 192, 106617. DOI: 10.1016/j.compag.2022.106617

[70] Hao, M., Li, H., Huang, C., Zhang, X., & Shi, Y. (2020). Towards Efficient and Privacy-Preserving Federated Deep Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 31(4), 1428-1442. DOI: 10.1109/TNNLS.2020.2967406

[71] Cheng, Y., Zhang, Y., Chen, T., & Zhou, Z. (2021). Secure Federated Transfer Learning for Privacy-Preserving Data Sharing. *IEEE Transactions on Big Data*. DOI: 10.1109/TBDATA.2021.3053896

[72] Nguyen, D. C., Pathirana, P. N., Ding, M., & Seneviratne, A. (2021). Federated Learning for Internet of Things: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials*, 23(3), 1622-1658. DOI: 10.1109/COMST.2021.3075439

[73] Zhou, X., Li, Y., & Wang, J. (2021). Federated Learning with Differential Privacy for IoT in Smart Agriculture. *IEEE Internet of Things Journal*, 8(9), 6722-6733. DOI: 10.1109/JIOT.2021.3063772

[74] Khan, L. U., Saad, W., Han, Z., & Bennis, M. (2021). Federated Learning for Distributed Smart Farming: Challenges and Opportunities. *IEEE Network*, 35(4), 183-189. DOI: 10.1109/MNET.011.2000604

[75] Zhang, T., He, X., & Zhou, Y. (2022). Distributed Federated Learning for Precision Agriculture with IoT Devices. *Sensors*, 22(4), 1222. DOI: 10.3390/s22041222

[76] Yin, Y., Wei, W., Wang, H., & Yin, Y. (2021). A Review on Federated Learning Applications in Smart Agriculture. *Computers and Electronics in Agriculture*, 190, 106502. DOI: 10.1016/j.compag.2021.106502

[77] Mohammadi, M., Al-Fuqaha, A., Sorour, S., & Guizani, M. (2020). Deep Learning for IoT Big Data and Streaming Analytics: A Survey. *IEEE Communications Surveys & Tutorials*, 22(4), 2491-2521. DOI: 10.1109/COMST.2020.2985893

[78] Chen, L., Zhang, Z., & Peng, W. (2022). Federated Learning with Blockchain for Secure IoT-Enabled Smart Agriculture. *IEEE Internet of Things Journal*, 9(7), 5146-5157. DOI: 10.1109/JIOT.2021.3102726

[79] Sharef, B. T. (2022). The Usage of Internet of Things in Agriculture: The Role of Size and Perceived Value. *Informatics*, *46*(7), 73–84. https://doi.org/10.31449/inf.v46i7.4275

[80] Quy, V. K., et al. (2022). IoT-Enabled Smart Agriculture: Architecture, Applications, and Challenges. *Applied Sciences*, *12*(7), 3396. https://doi.org/10.3390/app12073396

[81] Sharma, A., Jain, A., Gupta, P., & Chowdary, V. (2020). Machine Learning Applications for Precision Agriculture: A Comprehensive Review. *IEEE Access*, *8*, 4843–4873. https://doi.org/10.1109/ACCESS.2020.3048415

[82] Sinha, B. B., & Dhanalakshmi, R. (2022). Recent advancements and challenges of Internet of Things in smart agriculture: A survey. *Future Generation Computer Systems*, *126*, 169–184. https://doi.org/10.1016/j.future.2021.08.006

[83] Dutta, P. K., & Mitra, S. (2021). Application of agricultural drones and IoT to understand food supply chain during post COVID-19. In *Agricultural Informatics: Automation Using IoT and Machine Learning* (pp. 67–87). Wiley. https://doi.org/10.1002/9781119769231.ch4

[84] Bourstianis, A. D., et al. (2022). Internet of things (IoT) and agricultural unmanned aerial vehicles (UAVs) in smart farming: A comprehensive review. *Internet of Things*, *18*, 100187. https://doi.org/10.1016/j.iot.2022.100187

[85] Akhter, R., & Sofi, S. A. (2021). Precision agriculture using IoT data analytics and machine learning. *Journal of King Saud University - Computer and Information Sciences*, *34*(8), 5602–5618. https://doi.org/10.1016/j.jksuci.2021.05.013

[86] Madushanki, A. A. R., Halgamuge, M. N., Wirasagoda, W. A. H. S., & Ali, S. (2021). Adoption of the Internet of Things (IoT) in agriculture and smart farming towards urban greening: A review. *International Journal of Advanced Computer Science and Applications*, *12*(4), 1–12. https://doi.org/10.14569/IJACSA.2021.0120401

[87] Quy, V. K., et al. (2021). IoT-Enabled Smart Agriculture: Applications and Challenges. *Sensors*, *21*(17), 5920. https://doi.org/10.3390/s21175920

[88] Misra, N. N., Dixit, Y., Al-Mallahi, A., Bhullar, M. S., Upadhyay, R., & Martynenko, A. (2020). IoT, big data, and artificial intelligence in agriculture and food industry. *IEEE Internet of Things Journal*, *9*(9), 6305–6324. https://doi.org/10.1109/JIOT.2020.2998584

Scalable Front-End Architecture: Building for Growth and Sustainability

Oleksandr Tkachenko[1*], Aleksei Chechet[2], Maksim Chernykh[3], Sergei Bunas[4], Przemyslaw Jatkiewicz[5]
[1]Playtech, Sofia 1784, Bulgaria
[2]New Edge DWC-LLC, Dubai 00000, United Arab Emirates
[3]Boom Pay, Inc, Austin 78704, United States of America
[4]Supersales Global Pte Ltd, Singapore 049422, Singapore
[5]Department of Business Informatics, University of Gdansk, Gdansk 80-309, Poland
E-mail: otkachenko893@gmail.com[1], al.chechet@outlook.com[2], chernykh_maks@hotmail.com[3], sergeibunas@outlook.com[4], p.jatkiewicz@hotmail.com[5]
*Corresponding author

Keywords: Information security, structure with adaptive extensibility, interface design, platform development and reliability, ensuring privacy of customer data.

Received: May 30, 2024

The aim of the research is to develop strategies for building scalable interface architectures that ensure stability and protection of user data. The research identified the key characteristics of scalable interface architectures, such as ReactJS, GraphQL, and SSR. Within the scope of the research, principles for constructing scalable front-end architectures have been developed and described. These principles, such as modularity, componentisation, microservices architecture, adaptive and responsive design, as well as load management and scalability, data security, and fault tolerance, aim to ensure flexibility, efficiency, and user convenience across various devices and under different conditions. These principles enable the creation of architectures capable of adapting to changing requirements and scaling to ensure high performance and reliability when dealing with various types of data and workloads. The practical significance of the study is determined by the creation of a pragmatic guide for developers and architects of front-end systems who seek to create scalable applications.

Povzetek: Raziskava obravnava strategije za razvoj skalabilnih vmesniških arhitektur, ki zagotavljajo stabilnost in varnost uporabniških podatkov. Identificirane so ključne značilnosti, kot so modularnost, komponentizacija, mikroservisna arhitektura ter prilagodljiv in odziven dizajn. Ti principi omogočajo prilagodljivost, učinkovitost in uporabniško prijaznost na različnih napravah in v različnih pogojih.

1 Introduction

The examination of scalable interface architecture and data security is particularly pertinent in today's digital landscape. As the number of users on online platforms continues to surge and the volume of personal data proliferates, the imperative for secure systems becomes increasingly critical. The persistent advancement of Internet technologies, coupled with the evolving nature of cybersecurity threats, compels system developers to continuously refine their strategies for scalability and security. This ongoing evolution necessitates the development of innovative solutions that can effectively manage growing workloads while ensuring robust protection of user information. In this context, the ability to adapt to changing requirements and loads, maintain high performance, and safeguard data integrity and confidentiality are paramount. Consequently, the integration of advanced technologies and comprehensive security measures in scalable interface architectures is not only a necessity but a fundamental aspect of modern system development.

Thus, the study on the mentioned topic is crucial for effective adaptation to changing conditions and ensuring long-term success in the virtual environment. Furthermore, as the volume of data and the number of users on platforms increases, there are challenges in ensuring high performance, adaptability to change, and data protection. Improved approaches to front-end system design are needed to scale efficiently and ensure the security of customer data. The challenges range from online business to personal data privacy and require careful research to create innovative solutions.

Many other researchers addressed aspects related to this topic. The study by D. Dinev [1] highlights the rapid development and implementation of IoT devices, emphasizing the security and protection of user data. A promising technology in this field is Long Range (LoRa), which allows secure data transmission over long distances and offers extended battery life for IoT devices. M. Petkova [2] addressed various options for creating test platforms for innovative research, focusing mainly on the area of next-generation internet development. The study results in aim to provide generalised information about the structure and performance of such platforms.

W. Dimitrov and T. Ostrovska [3] investigated the use of Virtual Reality (VR) and Augmented Reality (AR) technologies in the context of creating scalable front-end architectures, with a focus on their growth and sustainability. The study examines various aspects of VR and AR application development, including software and hardware, developer tools, and user data protection issues. A. Urilski et al. [4] addressed some of the most popular and widely used e-learning systems. The authors emphasised the importance of reliability and protection of user privacy. D. Orozova [5] addressed new challenges of analysing and managing large amounts of data in distributed storage, as well as ensuring security, resilience, and privacy of information. The paper analyses data processing and analysis using the Map/Reduce paradigm. A. Tsenov [6] analysed architecture management in modern telecommunication and information systems, starting with the introduction and consideration of the basic principles of management in telecommunications. Models and methods for assessing the quality of service are presented as deliverables, including a hierarchical user experience model and a fuzzy logic method for multi-criteria evaluation.

As such, many researchers addressed various aspects in the area of data security and privacy, including the risks of data breaches and how to improve protection. However, these studies did not address issues related to the application of scalable architectures in front-end development security.

This article contributes to the field by providing a comprehensive analysis of scalable front-end architectures and data security in the context of modern digital environments. It not only evaluates various established approaches but also introduces practical insights into the implementation of these architectures in leading global companies. By addressing the challenges of scalability, stability, and data protection, the study offers valuable recommendations for developing resilient and secure interface architectures. This research seeks to close these gaps and propose innovative approaches for securing user data in scalable interface architectures. The study aims to develop methods for constructing scalable front-end architecture capable of ensuring stability and guaranteeing the protection of user data in the digital environment.

2 Materials and methods

The analysis method was used to examine existing scalable interface architectures, their features, advantages, and disadvantages. This method was used to conduct a systematic analysis of the literature and practical examples of the use of different architectural approaches. The analysis highlighted the main characteristics of each architecture and examples of their application in real projects. Moreover, this approach was used to explore data security issues and features of interface architectures, applying a wide range of technologies such as the IoT, test platforms, VR and AR, e-learning systems, big data analytics and telecommunication systems. Furthermore, aspects of smart buildings, optimisation of instruction caching, cloud development, application of deep neural networks, strategies for protecting sensitive information, semantic web, artificial intelligence, orchestration, and communication techniques, microservice architecture, native compilation platform for scalable architectures and distributed registry technologies were also addressed.

The experimentation method was utilised to develop and provide detailed descriptions of the principles underlying scalable front-end architectures. It involved creating conceptual models and conducting an iterative process of prototyping and validation. This experimental approach allowed not only to assess the effectiveness and applicability of each principle but also to identify their impact on the overall architecture and user experience. The results obtained served as the basis for formulating recommendations and establishing best practices in the field of scalable interface architecture development. Some of the illustrations were generated using the Matlab tool. Furthermore, a structural diagram was employed to visually represent the key components and relationships in various interface architectures. It encompassed aspects such as client-side and server-side components, data protection mechanisms, data management methods, as well as scalability capabilities and system stability assurance.

The comparison table was a comprehensive analytical process that compared key characteristics of different scalable interface architectures, such as those used by Netflix, Airbnb, Facebook, Amazon, and Google. Key aspects of each of the architectures were covered to create the table, including their structure, data processing methods, scaling mechanisms, technology utilisation and more. Each of the architectures was thoroughly analysed, highlighting features, advantages and disadvantages. As such, a detailed comparison between the different architectures was created and their key differences were identified. This approach provided a complete understanding of how each architecture functions and what their peculiarities are. This level of detail is necessary for selecting the best architecture depending on the specific requirements and goals of the project. Thus, this method facilitated mastering the concept of scalable interface architectures, their implementation, analysis of real data and comparison of different approaches, which allowed to highlight recommendations and draw certain conclusions.

3 Results

3.1 Introduction to scalable interface architecture

In the contemporary digital landscape, the proliferation of online platforms and the exponential growth of user data demand efficient management of data processing and storage. Ensuring high performance and security of transmitted information is crucial, particularly as systems must remain adaptable to evolving requirements and fluctuating loads. Scalability and stability of interface architecture are paramount in delivering a high-quality user experience. Scalability ensures that systems can handle increasing amounts of work by adding resources, thus maintaining performance levels. Stability ensures that systems operate reliably under various conditions, preventing failures and minimizing downtime. Together, these aspects ensure that users experience smooth and uninterrupted service, regardless of the volume of data or the number of users accessing the platform [7].

Table 1: Explanation of key terms in scalable interface architectures

Term	Explanation
Scalability	The ability of a system or application to efficiently increase its resources and bandwidth to handle growing workloads.
Interface Architecture	The structure and organization of the front-end components of an application, which affects its behavior and user interaction.
Stability	The ability of a system to maintain normal operation and prevent failures even under changing conditions and loads.
Data Protection	Measures and techniques to ensure the confidentiality, integrity, and availability of user information from unauthorized access or damage.
Component-Based Architecture	An approach to software development where the interface is composed of independent components that can be reused in different parts of the application.
Application Programming Interface (API)	A set of tools and protocols for the interaction of different applications, enabling integration and interaction of system components.
Caching	A mechanism for storing data to speed up access to it in the future, which enhances the performance of the system.

There are certain prominent terms in the context of this topic. Scalability is the ability of a system or application to efficiently increase its resources and bandwidth to handle growing workloads. Interface architecture defines the structure and organisation of the front-end components of an application, which affects its behaviour and interaction with the user. Stability refers to the ability of a system or application to maintain normal operation and prevent failures even under changing conditions and loads. Data protection includes measures and techniques to ensure confidentiality, integrity and availability of user information from unauthorised access or damage. Component-based architecture involves developing software from independent components, which facilitates scaling and modification of the application [8]. API is a set of tools and protocols for the interaction of different applications. The modularity of the system enables splitting it into independent components, which simplifies development and support. Caching is a mechanism for storing data to speed up access to it. Design patterns are repeatable solutions to typical problems in software design that help improve its quality and simplify development [9, 10].

Several approaches for scalable front-end architectures and data security should be considered. A micro-frontend architecture involves dividing the interface into separate, independent modules, enhancing scalability and allowing for the integration of new features without disrupting the overall application. Single-page applications (SPAs) load all necessary HTML, and JavaScript at once, improving performance and user experience while utilizing modern authentication and authorization methods for data security. Using a Content Delivery Network (CDN) reduces resource load times and enhances scalability, providing protection against DDoS attacks and data encryption [11]. Server-side rendering (SSR) processes HTML on the server, improving Search Engine Optimisation (SEO). and reducing initial page load time, thereby better controlling data security. Containers and orchestration tools like Docker and Kubernetes facilitate easy scaling and resilience of applications by isolating containers to improve security. The choice of approach depends on the specific project requirements and the needs for scalability and data security [12].

Nevertheless, existing approaches to developing interface architectures are not always able to provide an optimal combination of scalability, stability, and data security. Difficulties arise due to the variety of platforms on which web applications are deployed and the increasing volume and variety of data that needs to be processed and transmitted. In light of these challenges, it becomes necessary to develop new approaches to build an interface architecture that can adapt to the growing requirements and ensure the security of user data. It is also necessary to consider that effective scalability and data security requires an integrated approach that includes not only the selection of appropriate technologies and methods but also the right architectural concept and constant attention to system upgrades and enhancements.

3.2 Principles of building scalable interface architectures

Scalable interface architectures encompass various principles of construction. For example, modular interface architecture involves an approach where the user interface is divided into separate modules, each performing a specific function or task (Figure 1). These modules are independent of each other and can be easily replaced, modified, or added without the need to rewrite the entire interface. Each component in modular architecture has clearly defined functional boundaries, making them easier to understand and support. The use of modular architecture provides flexibility and scalability to the system, allowing the interface to adapt to changing user and business needs. Additionally, modular architecture promotes code reuse, speeding up the development process and ensuring uniformity and consistency in the user interface.

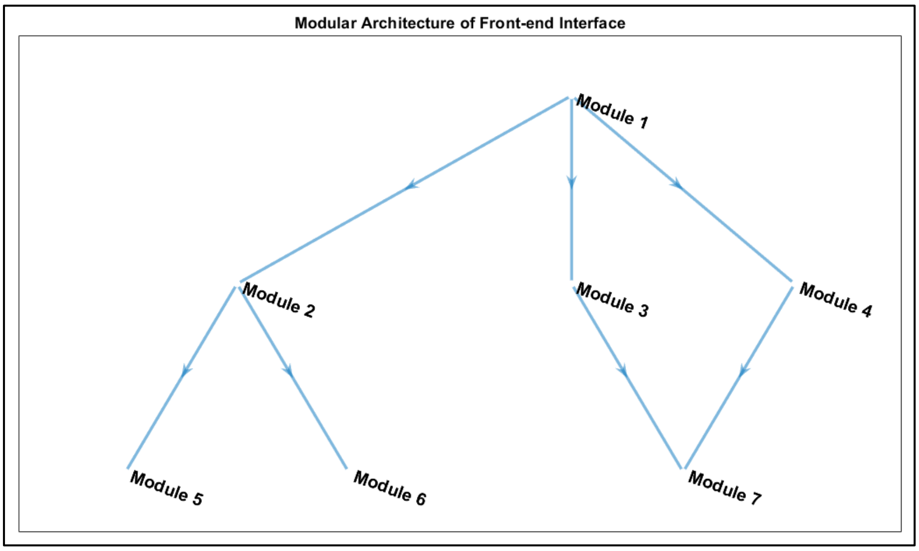


Figure 1: Modular interface architecture

This graph represents a modular architecture of the front-end interface, where each module performs a specific function or task, and the connections between modules depict the interaction between them. Modular architecture allows for easy scalability of the system and reuse of interface components to create more complex user interfaces.

Component-based architecture of the interface is an approach where the user interface is composed of a set of ready-made components such as buttons, text fields, lists, and other elements [13]. These components are independent and reusable building blocks that can be easily combined and arranged to create various interfaces. Components provide clearly defined functionality and well-designed API's, making them easy to use and integrate into projects. This approach simplifies interface development because developers can focus on creating and configuring components without spending time implementing each interface element from scratch. Additionally, component-based architecture promotes code reuse and ensures consistency in the appearance and behavior of the interface, facilitating the creation of cohesive and easily maintainable applications [14]. Indeed, component-based architecture of the interface provides modularity, reusability, flexibility, and customisation of the interface, making it an effective approach to user interface development.

In turn, microservices architecture is an approach to building a user interface where functionality is divided into separate independent components called microservices [15]. Each microservice is responsible for executing a specific part of the interface's functionality, allowing developers to create more flexible, scalable, and easily maintainable systems. Microservices can be developed, deployed, and scaled independently of each other, providing a high degree of flexibility and scalability to the system. This approach helps to improve the system's fault tolerance, as failures in one microservice do not significantly impact the operation of other parts of the interface. Moreover, microservices architecture promotes code reusability and enhances the development process, making it more modular and easier to understand [16-18].

Adaptive design is an approach to user interface creation that automatically responds and adapts to various usage conditions, including screen resolution and device type. This principle ensures convenience and efficiency in using the interface across different devices, such as smartphones, tablets, and desktop computers. Flexibility and adaptability in systems are achieved through the use of scalable and customisable components, as well as mechanisms for automatic adjustment and scaling. This approach enables optimal user interaction with the interface regardless of their device and usage context, thus increasing user satisfaction and overall system efficiency.

Equally important is responsive design, which is a methodology for user interface development that dynamically responds to changes in the size and resolution of the user's device screen (Figure 2).

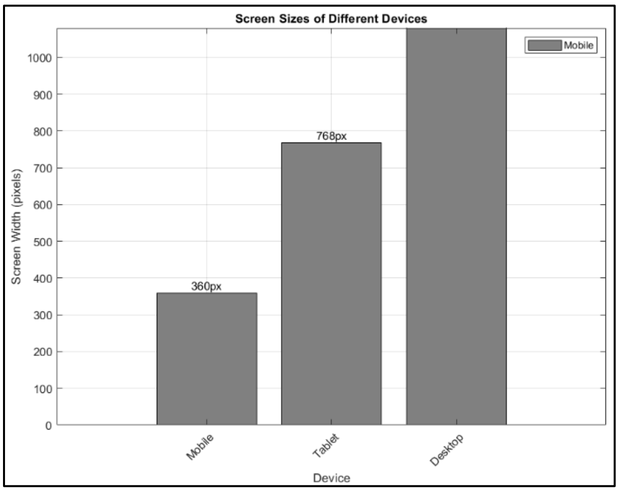


Figure 2: Responsive design

Horizontal and vertical scaling methods are essential approaches to increasing a system's capacity and performance. Horizontal scaling (scaling out) involves adding new servers or nodes to an existing system to increase computational power. This allows the system to handle more requests or data by distributing the load across multiple servers. Horizontal scaling is effective for large, distributed systems and can be implemented with minimal downtime. However, it requires managing distributed data and ensuring consistency across nodes. Vertical scaling (scaling up) involves increasing the resources of an existing server, such as adding more CPUs, memory, or storage. This enables a single server to handle larger workloads or perform more computations. Vertical scaling is simpler to implement since it doesn't require data distribution, but it has physical limitations and can become expensive at higher levels. Additionally, it introduces a single point of failure, as the system relies on one server. Scalable interface architectures involve designing systems to efficiently handle increasing loads and data volumes without sacrificing performance. Key principles include:

- Modularity: Dividing the user interface into independent modules that perform specific functions, allowing for easy scaling and modification without affecting the entire system.
- Component-based architecture: Building the interface from reusable components, streamlining development and ensuring consistency across the application.
- Microservices architecture: Splitting functionality into independent services that can be developed, deployed, and scaled separately, enhancing flexibility and resilience.
- Adaptive and responsive design: Ensuring the interface automatically adjusts to different devices and conditions, providing an optimal user experience across various platforms.

This principle ensures optimal content display and interface layout on any device, ranging from mobile devices and tablets to desktop computers and televisions. By using flexible grids, images, and media queries, responsive design enables the creation of universal interfaces that automatically adapt to different screens and provide users with comfortable and intuitive interaction with the application or website [19].

This bar chart illustrates the screen width of various devices – mobile phones, tablets, and desktop computers – in pixels. Each bar in the chart represents one of the devices, with its height corresponding to the screen width. Resilience and fault recovery are key aspects in designing scalable interface architectures. These principles aim to ensure continuous system operation even in the event of failures or disruptions. To achieve resilience, load distribution and resource redundancy should be implemented, enabling the system to maintain functionality and ensure service availability even under adverse conditions [20]. Fault recovery mechanisms include automatic service and data restoration, rapid problem diagnosis, and the utilisation of backup resources to minimise system downtime. These principles contribute to increasing the reliability and stability of interface architectures, ensuring uninterrupted operation even under extreme loads or failures [21].

Attention should also be paid to data security and access control, which is one of the fundamental principles in building scalable interface architectures (Figure 3). This principle aims to ensure the confidentiality and integrity of data, as well as control access to it. Data encryption methods, authentication, and authorisation mechanisms play a crucial role in preventing unauthorised access and use of information. Data encryption ensures its protection during transmission and storage, while authentication and authorisation mechanisms define user access rights to specific system resources and functions. This approach helps minimise the risks of information leakage and ensures compliance with modern data security standards. Effective implementation of the data security and access principle is an integral part of creating reliable and secure interface architectures.

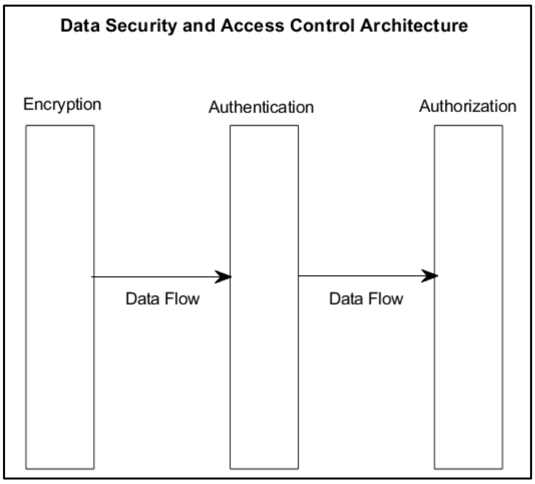


Figure 3: Data security and access control

This graphical representation of the security architecture illustrates key components (encryption, authentication, authorization) and the flow of data between these components. Such visualisation helps understand the structure and interaction of key elements in the data security and access control system.

Load management and scalability are crucial aspects to consider when designing scalable interface architectures (Figure 4). They ensure the system's ability to efficiently handle increasing volumes of requests and data. Horizontal and vertical scaling methods allow increasing system performance by adding computational resources or distributing the workload among different components. Resource and request management help optimize the utilization of hardware and software resources, maintaining a balance between system performance and efficiency. As a result, the system can maintain high responsiveness and availability even under significant loads, which is a key requirement for modern applications and services.

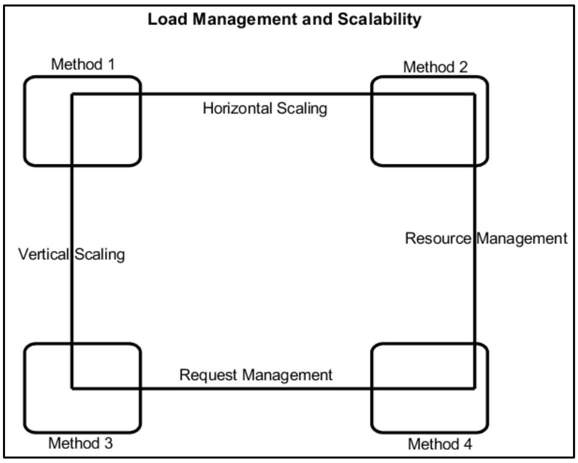


Figure 4: Load management and scalability

This visualisation illustrates various management methods and shows the relationship between them. Each block is accompanied by a text label explaining its purpose. It helps understand key aspects of load management and scalability, such as horizontal and vertical scaling, resource and request management. To visualise the fundamental components and principles underlying scalable architecture, it is also worthwhile to consider a general structural diagram of web applications (Figure 5).

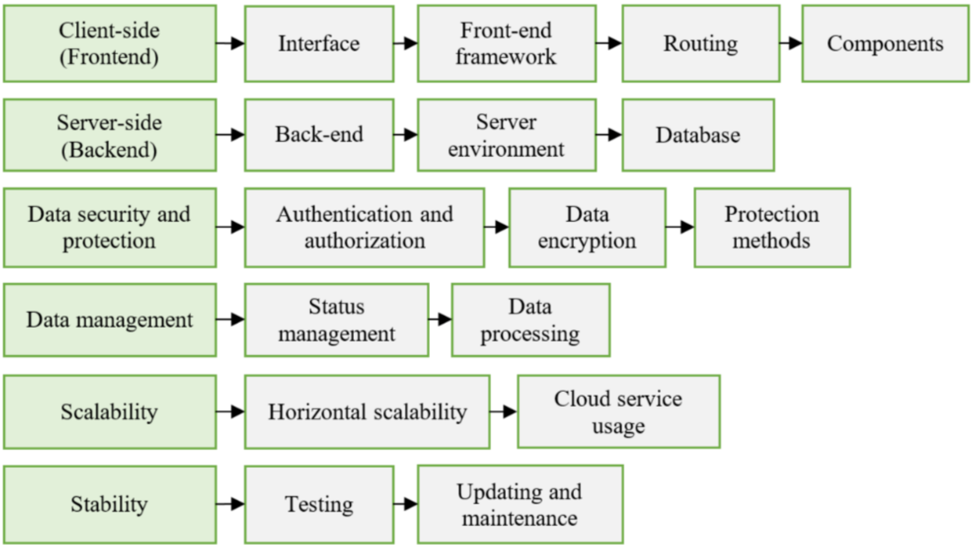


Figure 5: General architecture diagram of a scalable web application

Such a diagram helps better understand the overall architecture of the application, identify key components, and define the main data flows in the system.

3.3 Real data analysis and comparison of approaches

Scalable interface architectures are an important aspect of frontend development, enabling the creation of flexible and extensible applications. Several types of such architectures can be distinguished. For example, a modular architecture divides an application into independent modules, each of which is responsible for a specific functionality. This allows developers to work on different modules in parallel and easily scale the application if necessary. Component-based architecture divides the interface into independent components that can be reused in different parts of the application. This approach helps create flexible and scalable interfaces where each component is responsible for its functionality. In turn, microservice architecture divides the application into small, independent services, each of which performs a specific function.

Thus, this architecture flexibly scales and deploys individual parts of the application and provides easy system support and extension. A pure architecture offers the division of the application into layers, each with its responsibilities and dependencies directed inward, allowing the creation of highly connected but loosely coupled components, and facilitating the testing, maintenance, and scaling of the application. In addition, microservices architecture divides an application into small, self-contained services, each of which performs a specific function and can be deployed and scaled independently. This enables flexible and scalable systems where each service can be developed, deployed, and scaled independently. Model-View-Presenter architecture divides the application into a model, a view, and a presenter. This architecture separates the application logic from its presentation, making it easier to test and maintain the application. The Redux architecture proposes the use of unidirectional data flow to manage the application state, allowing for predictable and scalable interfaces where the application state is stored in a single repository and is only changed through actions [22].

Furthermore, a separation of duties architecture should also be added. The principle of segregation of duties involves breaking each application layer into smaller sections, each of which solves only one application problem or function. Scalable logging also exists. When moving from heavy monoliths to distributed computing systems, log monitoring becomes difficult [23]. A distributed approach helps developers to combat problems and also log as much information as required. Equally important is the principle of sole responsibility. This principle suggests that each class should have only one reason for the change, which promotes clean, reproducible code with fewer bugs. These are just some examples of scalable interface architectures and principles. Depending on the requirements and specifics of the project, it is possible to select the appropriate architecture or a combination of several architectures to achieve the desired level of scalability and flexibility.

Aside from examples of architectures and principles, it is worth considering real-world examples of scalable interface architectures (Table 2). For instance, Netflix uses a microservice architecture with the front end divided into many independent modules. Netfilx uses ReactJS to create dynamic interfaces and GraphQL for API management and data caching. Netflix's architecture scales horizontally by adding new servers to the pool. Airbnb, on the other hand, uses a SSR based architecture with ReactJS to optimise page load time and SEO. They also use GraphQL for API management and data caching and scale horizontally by adding new servers to the pool. Facebook, on the other hand, uses ReactJS to create dynamic interfaces and GraphQL for API management and data caching. They have also developed their content management system called GraphQL Relay. Facebook's architecture scales horizontally by adding new servers to the pool. Amazon also uses a microservice architecture to split the front end into independent modules. They use ReactJS and AngularJS to create dynamic interfaces and use Amazon Web Services (AWS) Lambda for server-side functions. Amazon's architecture also scales horizontally by adding new servers to the pool. Lastly, Google uses ReactJS and AngularJS to create dynamic interfaces and uses Google Web Toolkit (GWT) to compile Java code into JavaScript. They also use Google Cloud Platform (GCP) for hosting and scaling. Google's architecture scales horizontally by adding new servers to the pool.

Table 2: Comparative table of scalable front-end architectures

Company	Peculiarities	Pros	Cons
Netflix	Microservice architecture. Use of ReactJS and GraphQL. Horizontal scalability	High flexibility and scalability. Effective data management using GraphQL.	High infrastructure configuration requirements Specialised skill-set requirements
Airbnb	SSR and ReactJS. GraphQL for API management. Horizontal scalability	Optimized page loading and SEO Flexibility in data management with GraphQL	Additional server infrastructure maintenance expenses
Facebook	Use of ReactJS and GraphQL. "GraphQL Relay" content management system. Horizontal scalability	High performance and flexibility. Effective data management via GraphQL	Requires substantial resources for development and upkeep
Amazon	Microservice architecture. ReactJS and AngularJS. Horizontal scalability via AWS Lambda	High flexibility and scalability. Resource-saving using AWS Lambda	High AWS configuration and infrastructure management requirements
Google	ReactJS, AngularJS, and GWT Horizontal scalability via GCP	Variety of technology and tools. High performance using GCP	Difficulty in deploying and maintaining applications on GCP

This table can help organise information and make a more informative comparison between different scalable interface architectures. Furthermore, it is possible to highlight the basic principles of scalable interface architectures. It is worth considering the use of microservice architecture, which involves splitting the front end into independent modules. As well as the use of ReactJS, a widely used JavaScript library, for developing dynamic interfaces. The implementation of GraphQL as a query language for efficient API management and data caching should be emphasised. An important aspect is the use of SSR to optimise page load times and improve SEO performance. And horizontal scaling, which involves adding new servers to the pool to increase performance.

3.4 Recommendations

After analysing various scalable interface architectures, several recommendations for their further improvement can be identified. To improve the reliability and stability of the architecture, extensive use of automated tests and Continuous Integration/Continuous Delivery (CI/CD) tools is recommended. This will help identify and fix problems early in the development process. It is important to continue to research and develop methods to optimise interface performance. This includes optimising page loading, reducing response times and improving the overall responsiveness of the user interface. To meet the growing needs of developers and users, it is recommended that the functionality of the architecture be continually enhanced. This may include adding support for new technologies, expanding APIs, and improving development tools. In addition, it is important to ensure the availability of tutorials and documentation for developers, as well as to provide quality technical support. This approach will help accelerate the process of implementing and improving the architecture, as well as improve the skills of employees.

4 Discussion

For a thorough review of this topic, other studies on scalable front-end architecture should be consulted. As such, L. Chamari et al. [7] considered modern smart buildings that require integration of various information systems such as Building Management Systems, Energy Management Systems, IoT, Building Information Models and others for effective management and monitoring. However, the lack of a well-defined system architecture with APIs poses challenges for developing and deploying effective applications for smart buildings. The architecture proposed in this paper, based on the principles of microservices, APIs, cloud components and full-stack partitioning, aims to enhance the repeatability, modularity, and scalability of such applications. The research results include the development of three smart building applications that demonstrate the reusable and modular aspects of the proposed architecture. That is, the common aspects between the referenced work and this study lie in the endeavour to develop scalable applications using modern architectural principles. However, the difference is that this study emphasises ensuring the stability and security of user data, which is a critical aspect in application development, especially in the context of the growth of Internet technologies and cybersecurity threats. The study Y. Degawa et al. [22] addresses the topic of scalable front-end architecture, focusing specifically on smart buildings and the integration of different information systems, and draws attention to the challenges of developing effective smart building applications due to the lack of a well-defined system architecture with APIs.

Y. Degawa et al. [22] emphasised that processor performance depends on the instruction delivery front end, as the design of interface architecture requires consideration of instruction processing performance and cache management in processors. Various techniques such as cache replacement algorithms and prefetching have been investigated to improve performance. One key factor is to reduce instruction cache misses. However, the number of misses does not always explain the performance changes due to parallel processing of misses by the front end. The study proposes a new factor, the number of missed areas, which explains the performance variation. Hence, both studies address the performance and scalability issues of the processor front end and recognise the importance of considering the performance of instruction processing and cache management to design a scalable architecture. However, unlike this study, which focuses on designing a scalable interface architecture to protect user data, the research concentrates on analysing the number of cache misses and introducing a new factor, the number of miss areas.

P. Mwiinga [24] pointed out that in today's world, saturated with technological innovation and extensive digital interaction, preserving user privacy, and providing robust security measures are becoming increasingly important. The study analyses technologies aimed at preserving user privacy and examines their key role in balancing security and privacy protection. Through a detailed examination of various techniques and innovations in this area, the study assesses the challenges, achievements and prospects associated with achieving an optimal balance between security and user privacy in the digital space. Thus, both studies focus on the importance of user privacy and security in the context of the digital environment. They also approach this issue by analysing technologies aimed at preserving user privacy and evaluating their role in balancing security and privacy. However, while the 2023 study is more focused on analysing the challenges, advances and prospects in security and privacy, this study specifically aims to develop a scalable interface architecture with a focus on stability and protection of user data.

K. Khan [25] proposed a comprehensive guide to cloud development suitable for both experienced professionals and novices. He reviewed the key concepts of cloud applications and explored architectural patterns and their development techniques. The work also discusses DevOps practices and the importance of securing and monitoring applications in the cloud, real case studies and future trends in cloud development. Both studies address important aspects of software development using advanced technologies. While the aforementioned study focuses on cloud development and DevOps practices, this study focuses on scalable interface architecture with growth and resilience in mind and user data protection. However, both approaches complement each other, as cloud development and DevOps practices can be effectively integrated into a scalable UI Architecture, ensuring application stability and security.

R. Davis [26] emphasized the ubiquitously used deep neural networks which are capable of more accurate performance on various tasks compared to manual systems. He emphasised that this has increased the need for more sophisticated models to handle large amounts of data, but traditional computing architectures have faced a performance bottleneck due to the need to move data around. A new approach using an optical neural network with multiple analogue frequency conversions is proposed in this paper. The performance of the system has been successfully demonstrated on a three-layer Deep Neural Networks for Modified National Institute of Standards and Technology (MNIST) digit analysis, which confirms its scalability and the possibility of all-analogue signal processing from start to finish.

K. Lawal et al. [27] predicted that by 2030, the number of IoT devices connected to the Internet will exceed 125 billion units. This will create large amounts of data and require scalable architectures. The study demonstrates the advantages of fog computing over cloud computing in scalable scenarios. The fixed building architecture limits the benefits of fog computing due to the low number of IoT devices. However, in scalable scenarios, the benefits of fog computing are significant. Thus, both studies point to the need for scalable data processing solutions.

E. Blessing and H. Klaus [28] focused on the implementation of face-covering detection technologies in public places, which raises questions about data security and privacy. This work examines strategies to protect sensitive information and considers the technical and ethical aspects of using the technology. Security measures adopted include encryption and access control, while privacy measures aim to preserve user rights and autonomy. Legal compliance and educational initiatives also play an important role in ensuring the ethical use of technology [29]. In summary, both studies address data security and privacy issues in the context of emerging technologies. However, this paper focuses on the development of a scalable interface architecture, while the other focuses on the implementation of face-covering detection technology in public spaces.

D. Recupero et al. [30] indicated that recent breakthroughs in computer science, such as the semantic web and artificial intelligence, have enabled the development of a flexible human-like service architecture based on the Zora service. This architecture consists of three modules to interact with humans and perform various tasks. It is flexible and can be extended with new modules, and inbuilt modules can be easily augmented with learning resources. Each module has a frontend for interacting with people and a backend for performing computations on the server. That is, both studies emphasise the flexibility of their architectures and the ability to extend them with new modules. However, this study focuses on the security and scalability aspects of the interface architecture, while the 2019 study focuses more on user interaction and task execution.

Y. Perera and D. Jayasuriya [31] identified that the use of web applications has increased significantly in recent times, generating interest in improving their usability. Frontend performance plays an important role in this, as it is the interface for users. To improve frontend performance and scalability, micro-frontend architectures are used to enable developers to deal with various issues such as orchestration and communication between components. Advanced technologies such as ReactJS and NodeJS are used to implement such architectures. Recent research has proposed new orchestration and communication techniques, resulting in significant improvements in web application performance. So, both studies highlight the increasing usage of web applications and also emphasise the importance of frontend performance and scalability of systems. However, this study focuses on designing a scalable interface architecture with an emphasis on user data protection. While other work focuses on using micro-frontend architecture to improve frontend performance and scalability.

O. Nikulina and K. Khatsko [32] emphasise that the modern development of various web systems uses new microservice architecture to improve features such as performance and portability. This requires the conversion of legacy systems from monolithic architecture to microservice architecture, which is a complex and costly process. This research aims to develop a method to apply micro-frontends to improve monolithic single-page applications. The authors proposed a method to transform the architecture of a software system using microservices. The approach used is based on reverse engineering and moving to a modular architecture to improve the performance and portability of the application. The study resulted in a 200% performance improvement and stabilised communication time between application components. Thus, both studies emphasise the need to use modern approaches to improve the performance of web-based systems and suggest certain methods for transforming the architecture of software systems using new technologies. However, this study focuses on developing a scalable front-end architecture that provides stability and protection of user data, while the second study focuses on improving monolithic single-page applications using micro-frontends.

Similar to previous authors, A. Pant [33] confirmed that securing sensitive data has become a priority for everyone from individuals to businesses and governments. This paper discusses the critical importance of data security and privacy compliance in today's society. As technology continues to evolve, changing the way information is created, collected, and used, threats to data security are becoming increasingly serious [34; 35]. The author emphasises the multifaceted importance of data security and privacy compliance in various fields of endeavour. The importance of protecting personal information to protect privacy rights and maintain public trust is discussed first. Both the frequency and complexity of cyber threats are increasing, highlighting the importance of data security measures to prevent unauthorised access and information leaks. The general aspect of the research is the importance of data security in today's digital world. Furthermore, the increasing threats to data security in the context of technological advancement are also pointed out. However, this study focuses on the development of a scalable interface architecture that ensures stability and security of user data, while the 2023 study focuses on a general analysis of the critical importance of data security in today's society. Thus, while both studies address the issue of data security, their methodologies and focuses are different.

N. Paraskevopoulos et al. [36] introduced SpinQ, the first native compilation platform for scalable spin-qubit architectures. SpinQ addresses the unique operational constraints of the crossbar, enables scalable compilation, and achieves O(n) computational complexity. The authors analysed the performance of SpinQ on this architecture by compiling a variety of quantum circuits, revealing unique insights into the matching and architecture. The proposed new mapping techniques may improve the execution success of algorithms on this architecture and inspire further research in this area. That is, both papers present innovative approaches to developing architectures for specific domains. While this study focuses on the design of a scalable interface architecture for securing user information, the other focuses on the development of a compilation platform for scalable spin-qubit architectures. In doing so, both studies highlight the importance of considering the unique constraints and features of the target architectures.

S. Naqvi and A. Mohsin [37] considered an IoT system that has vulnerabilities such as scalability, security, and access control due to centralised architecture. To solve these problems, Distributed Ledger Technology (DLT) is used to provide a secure and scalable infrastructure for IoT. This research proposes the use of DLT based on Directed Acyclic Graph (DAG) to overcome the limitations of IoT devices. The authors presented an architecture based on the Tangle DAG framework and utilising a lightweight Masked Authentication Message data model. Experiments confirmed the energy-efficient execution of computations on the entire node, which allows efficient resource utilisation. As a result, a scalable access control infrastructure for IoT using a DAG-based distributed registry can be proposed. It comes out that both studies address the issues of security and scalability in different information systems. In both cases, the use of advanced technologies such as DLT and microservice architectures play a key role in achieving these goals. The main similarities of both works lie in the pursuit of security and scalability in information systems. While this study focuses on frontend aspects, providing stability and protection of user data, the study focuses on scalability, security, and access control aspects in centralised architectures. That is, this study proposes a scalable architecture for the interface and the second study proposes a scalable access control infrastructure for IoT.

T. Krishnappa [38] pointed out that the rapid growth and widespread use of social media have created new security and privacy challenges for individual users. The study aims to assess users' awareness of security and privacy measures and their understanding of potential risks and vulnerabilities. The results obtained are analysed to identify trends, knowledge gaps and user perceptions to help better understand the situation in this area. The results of the study can be used to develop recommendations for improving security measures and encouraging responsible online behaviour among social media administrators, legislators and users. The main objective of this study is to provide users with the necessary knowledge to ensure their protection in the digital age, emphasising their awareness and understanding of security and privacy in social media. Both works address important aspects of the information technology field but from different perspectives. This study focuses on the implementation of an interface scalable architecture with an emphasis on user data protection and stability. Furthermore, practical solutions are proposed to create robust systems that can scale efficiently and ensure data security. On the other hand, the 2023 survey highlights the security and privacy issues in social media and the level of user awareness of these issues. The author proposes a qualitative survey to assess users' understanding of security and privacy measures and to identify trends and gaps in their knowledge.

The general trend, confirmed by analysing various studies, indicates that data security and privacy are important criteria in digital technologies. However, given the evolution of the information space, attention must also be paid to user awareness of security and privacy measures within scalable interface architectures. This means that not only the technical aspects of security are crucial, but also the user's awareness of their role in data protection. Such an approach will lead to more resilient and secure information systems, as well as to a greater overall level of digital literacy.

5 Conclusions

The research conducted examined various aspects of scalable interface architectures and their impact on data security and privacy. The main topic addressed in the study was the issue of data security and privacy in the context of new technologies. Study results confirmed that these aspects are becoming increasingly important in the digital age, requiring serious attention from developers, businesses, and society as a whole. It is important to note that users' understanding of data security and privacy measures is a key factor in the successful implementation of scalable interface architectures.

One of the important study results is the development of methods and recommendations for ensuring data security and privacy in scalable interface architectures. The practical significance of such recommendations lies in the possibility of creating reliable and secure systems that can efficiently scale and protect user data. In addition, the study not only raised issues of user data security, but the growth and resilience of architectures were also important aspects. Data security is a key element in modern information technology, but it is also important to ensure that architectures are resilient and scalable to ensure their long-term performance and success in different usage scenarios.

Summarising the results and conclusions of the study, it is possible to note that the development of methods for creating scalable interface architectures plays a key role in the creation of reliable, productive, and flexible web applications. The right choice of architecture and its continuous improvement are essential for the successful implementation of projects and their competitiveness in the market. Further research in this area will contribute to the development of new methods and approaches and help solve current problems in interface development.

References

[1] D. Zh. Dinev, "LoRaWAN technology – The necessary solution to the challenges of wireless IoT networks," *Comput. Sci. Technol*, vol. 19, no. 1, pp. 23-30, 2021.

[2] M. Petkova, "Approaches for creating infrastructures for network and Internet innovations," *Yearb. Telecommun*, vol. 6, pp. 147-160, 2019.

[3] W. Dimitrov and T. Ostrovska, "Overview of virtual and augmented reality application development tools," 2020. [Online]. Available: https://doi.org/10.13140/RG.2.2.29293.82407/1

[4] A. Urilski, A. Malinova, and A. Rahnev, "Security threats and protection in E-Learning systems," in *Ann. Int. Sci. Conf. "Comput. Technol. Appl.",* Pamporovo, Bulgaria, 2021, pp. 115-129.

[5] D. A. Orozova, "Challenges of big data and big data analytics," *Ann. Burgas Free Univ*. vol. 28, pp. 47-52, 2018.

[6] A. Tsenov, "Management architectures in modern telecommunication and information systems," 2021. [Online]. Available: http://telecom.tu-sofia.bg/pdfs/Monograph%20Alexander%20Tsenov.pdf

[7] L. Chamari, E. Petrova, and P. Pauwels, "An end-to-end implementation of a service-oriented architecture for data-driven smart buildings," *IEEE Access*, vol. 11, pp. 117261-117281, 2023.

[8] M.-N. Tran and Y. Kim, "Optimized resource usage with hybrid auto-scaling system for knative serverless edge computing," *Fut. Gener. Comput. Syst*, vol. 152, pp. 304-316, 2024.

[9] N. Dhieb, H. Ghazzai, H. Besbes, and Y. Massoud, "Scalable and secure architecture for distributed IoT systems," in *2020 IEEE Technol. Eng. Manage. Conf. (TEMSCON),* Novi, MI, USA, 2020, pp. 1-6.

[10] D. Sabol and J. Skalka, "The architecture of visual design in modern web applications," in E. Smyrnova-Trybulska, P. Kommers, M. Drlík, and J. Skalka,

Eds., *Microlearning*. Cham, Switzerland: Springer, 2022.

[11] A. Quraishi, M. A. Rusho, A. Prasad, I. Keshta, R. Rivera, and M. W. Bhatt, "Employing Deep Neural Networks for Real-Time Anomaly Detection and Mitigation in IoT-Based Smart Grid Cybersecurity Systems," in *3rd IEEE Int. Conf. Distr. Comput. Electr. Circ. Electron., ICDCECE 2024*, Hybrid, Ballari: Institute of Electrical and Electronics Engineers, 2024. Available: https://doi.org/10.1109/ICDCECE60827.2024.10548160

[12] K. Li, Y. Ding, D. Shen, Q. Li, and Z. Zhen, "The design and research of front-end framework for microservice environment," in *2020 Int. Conf. Comput. Inf. Big Data Appl. (CIBDA)*, Guiyang, China, 2020, pp. 124-127.

[13] I. Aviv, R. Gafni, S. Sherman, B. Aviv, A. Sterkin, and E. Bega, "Infrastructure from Code: The Next Generation of Cloud Lifecycle Automation," *IEEE Software*, vol. 40, no. 1, pp. 42-49, 2023. Available: https://doi.org/10.1109/MS.2022.3209958

[14] I. Aviv, R. Gafni, S. Sherman, B. Aviv, A. Sterkin, and E. Bega, "Cloud Infrastructure from Python Code – breaking the Barriers of Cloud Deployment," in *17th Eur. Conf. Software Archit., ECSA 2023*, Istanbul, Turkey; 2023. Available: https://conf.researchr.org/details/ecsa-2023/ecsa-2023-journal-first/2/Cloud-Infrastructure-from-Python-Code-breaking-the-Barriers-of-Cloud-Deployment

[15] O. Titova, P. Luzan, N. Sosnytska, S. Kulieshov, and O. Suprun, "Information and Communication Technology Tools for Enhancing Engineering Students' Creativity," in *Lect. Notes Mech. Engin*, Cham: Springer, 2021, pp. 332-340. Available: https://doi.org/10.1007/978-3-030-77719-7_33

[16] R. Frankiv, ""Perfect presence space": Theoretical and practical aspects of the concept," *Archit. Stud*, vol. 10, no. 1, pp. 17-23, 2024.

[17] Yu. Dyba and Yo. Rotchniak, "Evolution of the buildings layout of the Lviv Main Railway Station," *Archit. Stud*, vol. 10, no. 1, pp. 35-46, 2024.

[18] A. Kucher and V. Mazurenko, "Essence and features of economic security of the industry sector," *Dev. Manag*, vol. 23, no. 2, pp. 16-24, 2024.

[19] R. Buil, M. A. Piera, and E. Ginters: "Multi-agent system simulation for urban policy design: Open space land use change problem," *Int. J. Model., Simul., Sci. Comput*, vol. 7, no. 2, art. n. 1642002, 2016. Available: https://doi.org/10.1142/S1793962316420022

[20] D. Belytskyi, R. Yermolenko, K. Petrenko, and O. Gogota, "Application of machine learning and computer vision methods to determine the size of NPP equipment elements in difficult measurement conditions," *Machinery and Energetics*, vol. 14, no. 4, pp. 42-53, 2023. Available: https://doi.org/10.31548/machinery/4.2023.42

[21] A. Koldovskyi, "Architectural frameworks for financial transformation in Ukraine," *Dev. Manag*, vol. 23, no. 2, pp. 25-37, 2024.

[22] Y. Degawa, T. Kolzumi, T. Nakamura, R. Shioya, J. Kadomoto, H. Irie, and S. Sakai, "A principal factor of performance in decoupled front-end," *IEICE Trans. Inf. Syst*, vol. 106, no. 12, pp. 1960-1968, 2023.

[23] O. Bezshyyko, A. Dolinskii, K. Bezshyyko, I. Kadenko, R. Yermolenko, and V. Ziemann, "PETAG01: A program for the direct simulation of a pellet target," *Comp. Phys. Commun*, vol. 178, no. 2, pp. 144-155, 2008. Available: https://doi.org/10.1016/j.cpc.2007.07.013

[24] P. Mwiinga, "Privacy-preserving technologies: Balancing security and user privacy in the digital age," *Int. J. Sci. Res. Publ*, 2023. [Online]. Available: https://zenodo.org/records/10406538

[25] K. Khan, *Unlocking Cloud-Native Potential: Building Scalable, Resilient Apps*. Berlin: Kindle Direct Publishing, 2023.

[26] R. Davis, *A Deep Learning and Signal Processing Architecture Using Frequency-Encoded RF Photonics*. Cambridge: Massachusetts Institute of Technology, 2022.

[27] K. Lawal, T. Olaniyi, and R. Gibson, "Leveraging real-world data from IoT devices in a fog-cloud architecture for resource optimisation within a smart building," *Appl. Sci*, vol. 14, no. 1, p. 316, 2023.

[28] E. Blessing and H. Klaus, "Ensuring the security and privacy of data collected through face covering detection," 2023. [Online]. Available: https://www.researchgate.net/publication/377159729_Ensuring_the_security_and_privacy_of_data_collected_through_face_covering_detection

[29] S. V. Symonenko, V. V. Osadchyi, S. O. Sysoieva, K. P. Osadcha, and A. A. Azaryan, "Cloud technologies for enhancing communication of ITprofessionals," *CEUR Workshop Proceed*, vol. 2643, pp. 225-236, 2020.

[30] D. Recupero, D. Dessì, and E. Concas, "A flexible and scalable architecture for human-robot interaction," in *Proc. 15th Eur. Conf. Ambient Intell.*, Cham, Switzerland, 2019, pp. 311-317.

[31] Y. Perera and D. Jayasuriya, "Enhancing the front-end web applications performance using design patterns and microservices based architecture," 2023. [Online]. Available: https://doi.org/10.13140/RG.2.2.36067.53286

[32] O. Nikulina and K. Khatsko, "Method of converting the monolithic architecture of a front-end application to microfrontends," *Bull. Natl. Tech. Univ. "KhPI" Ser. Syst. Anal. Manag. Inf. Technol*, vol. 10, no. 2, pp. 79-84, 2023.

[33] A. Pant, "Importance of data security and privacy compliance," *Int. J. Res. Appl. Sci. Eng. Technol*, vol. 11, no. 11, pp. 1561-1565, 2023.

[34] A. I. Leonow, M. N. Koniagina, S. V. Petrova, E. V. Grunt, S. Y. Kerimkhulle, and V. G. Shubaeva, "Application of information technologies in

marketing: Experience of developing countries," *Espacios*, vol. 40, no. 38, 2019. Available : http://www.revistaespacios.com/a19v40n38/a19v40n38p24.pdf

[35] S. Kerimkhulle, Z. Dildebayeva, A. Tokhmetov, A. Amirova, J. Tussupov, U. Makhazhanova, A. Adalbek, R. Taberkhan, A. Zakirova, and A. Salykbayeva, "Fuzzy Logic and Its Application in the Assessment of Information Security Risk of Industrial Internet of Things," *Symmetry*, vol. 15, no. 10, art. n. 1958, 2023. Available: https://doi.org/10.3390/sym15101958

[36] N. Paraskevopoulos, F. Sebastiano, C. Almudever, and S. Feld, "SpinQ: Compilation strategies for scalable spin-qubit architectures," *ACM Trans. Quantum Comput*, vol. 5, no. 1, p. 4, 2023.

[37] S. Naqvi and A. Mohsin, "Using Direct Acyclic Graph (DAG) based Distributed Ledger for a secure and scalable Internet of Things (IoT) architecture," 2023. [Online]. Available: https://uow.edu.pk/ORIC/MDSRIC/Publications/8th%20MDSRIC-148.pdf

[38] T. Krishnappa, "User awareness of security and privacy in social networking sites," *Int. J. Eng. Appl. Sci. Technol*, vol. 8, no. 5, pp. 38-54, 2023.

Enhanced Social Group Optimization algorithm for Solving Optimization Problems

Anima Naik
Department of CSE, Raghu Engineering College, Visakhapatnam Andhra Pradesh 530003, India,
https://orcid.org/0000-0002-7808-5994
anima.naik@raghuenggcollege.in

Keywords: Social group optimization, real-world design problems, computational complexity, metaheuristic algorithms, improved algorithms.

Received: January 12, 2024

In the last decades, the field of global optimization has experienced significant growth, leading to the development of various deterministic and stochastic algorithms designed to tackle a wide range of optimization problems. One notable member of this family is the Social Group Optimization (SGO) algorithm. The Improving Phase and the Acquiring Phase are its two main fundamental phases. The two upgraded versions of SGO with a modified improvement phase are Enhanced Social Group Optimization (ESGO) and Enhanced Modified Social Group Optimization (EMSGO). The key enhancement in these variants focuses on honing, refining skills and abilities to achieve greater effectiveness. To assess the performance of ESGO and EMSGO, an extensive comparative analysis is conducted, involving twelve algorithms, including recently introduced, potent metaheuristic methods. Since both ESGO and EMSGO are modified algorithms, a comparison is conducted between these two algorithms and six recently introduced improved/hybrid algorithms. Subsequently, twenty-six real-world design problems from the mechanical and chemical engineering areas are addressed by applying both modified methods. The simulation results leave no doubt about the capability of ESGO and EMSGO to consistently achieve optimal solutions. Their robust performance, both in comparative evaluations and real-world applications, underscores their potential in solving challenging optimization tasks.

Povzetek: Razširjeni algoritem za optimizacijo socialnih skupin (ESGO) izboljšuje izvirno optimizacijo socialnih skupin z dodajanjem faze posnemanja, kar povečuje raznolikost populacije in globalno iskanje rešitev. ESGO je bil uspešno uporabljen pri reševanju kompleksnih optimizacijskih problemov.

1 Introduction

Numerous optimization difficulties have emerged as a result of the technology's rapid progress and need to be addressed. These problems are common in many industries, such as minerals, machinery, chemicals, electronics, finance, and electronics. complex solution spaces and complex relationships of unknown variables are common features of real-world optimization problems. Large numbers of variables, complex nonlinear constraints, and significant computational effort are frequently present in these situations [1-2]. Because they are unable to balance accuracy and time cost, conventional optimization techniques have difficulties effectively addressing these nonproductivity discontinuity problems [3]. Metaheuristic optimization algorithms have demonstrated superior performance in balancing time cost and solution quality [4]. Because of their straightforward structure and absence of requirement that a problem be continuously derivable, metaheuristic optimization algorithms have been widely used to handle challenging optimization problems in natural and technical fields [5,6].

Metaheuristic algorithms have advanced significantly in the last few decades in terms of hyperparameter self-adaptation, population structure evolution, and theoretical characterization of the search dynamics [7]. A focus of metaheuristic algorithm research is how to balance algorithm exploration and exploitation for improved performance. To achieve balance, many studies utilize other operators or modify the algorithm settings [8]. To balance the exploration and exploitation of TLBO, Satapathy et al. presented an alternative search pattern technique [9]. Some metaheuristic algorithms' search performance during optimization is significantly influenced by programmable factors like crossover rate, mutation rate, and population size. Adaptive parameter control has been thoroughly investigated by researchers in order to address the problem of parameter value control at various phases of the optimization process [10]. OTLBO, a variation of the teaching learning-based optimization

(TLBO) algorithm that Satapathy et al. presented with orthogonal design and generates an optimal offspring by a statistical optimal method where a new selection strategy is applied to decrease the number of generations and make the algorithm converge faster. [11]. Evolution of the population structure has a significant impact on how well metaheuristic algorithms perform in searches. By include elite factors in the hierarchical population structure, Zhong et al. devised the differential evolution (DE) algorithm variant known as EHDE [12]. Wang et al. presented a four-layered GSA variation with a greater search capability dubbed MLGSA, which was inspired by the two-layered structure GSA [13]. Theoretical examination of the search dynamics has recently drawn a lot of interest from scholars in addition to the aforementioned variables [14].

Metaheuristic optimization algorithms can generally be divided into four groups [15]: algorithms based on physics and chemistry, swarm intelligence, human intelligence, and evolutionary principles. The principles of natural evolution serve as the basis for evolutionary-based algorithms. A common example is genetic algorithms, and its proposal was motivated by Darwinian evolution [16]. Genetic algorithms provide solutions through the concept of crossover and mutation of species in nature. DE [17], evolutionary programming [18], and evolutionary strategies [19] are other evolutionary-based algorithms that have been developed. Swarm-based algorithms, which construct optimization models by replicating animal social behaviour, fall under the second classification. ACO [21] and Particle Swarm Optimization (PSO) [20] are two of the most used swarm-based algorithms. By exchanging data on each person involved in the optimization process, they offer solutions. A few examples of swarm-based algorithms include the Artificial Bee Colony (ABC) [22], Whale Optimization Algorithm (WOA) [23], Butterfly Optimization Algorithm (BOA) [24], Seagull Optimization Algorithm (SOA)[25], Sooty Term Optimization Algorithm (STOA)[26], Chimp Optimization Algorithm(ChOA)[27], Jelly Fish (JS)[28]. Human-based algorithms, which draw their inspiration from human behaviour, are the third group. Some human-based algorithms are Teaching Learning Based Optimization (TLBO)[29], Social Group Optimization (SGO)[30], Past Present Future (PPF)[31], and Mine Blast Algorithm (MBA)[32]. Physical and chemical-based algorithms, which are motivated by the physical laws and cosmological chemical processes, are the fourth type. Gravitational Search Algorithm (GSA)[34] and Simulated Annealing (SA)[33] are two popular ones. Examples of physical and chemical-based algorithms include the Water Cycle Algorithm (WCA)[35], Ray Optimization(RO)[36], and Artificial Chemical Reaction Optimization Algorithm(ACROA)[37].

The SGO algorithm is a novel human-based algorithm proposed by Satapathy et al., inspired by the social behaviour of human for solving complex problem. It can be seen from literature that SGO has good performance on solving variety of real-world optimization problems [38-46] like many outstanding algorithms. But the NFL theorem [47] encouraged us to improve the SGO even if their performance is competitive with that of other algorithms. As observed in the literature, an algorithm may perform exceptionally well for a specific set of problems but often struggles with others. This phenomenon is supported by the No Free Lunch (NFL) theorem, which encourages researchers to propose new algorithms or improve existing ones. In this context, the SGO algorithm has been modified to enhance its capability in solving real-world problems.

In this paper, the following key contributions are made:

- The enhanced SGO algorithm is introduced in two forms, ESGO and EMSGO, which incorporate practical problem-solving mechanisms based on rational group dynamics, aligning with the foundational principles of SGO.
- The performance of the proposed ESGO and EMSGO algorithms is evaluated using a comprehensive set of 23 benchmark test functions. When compared to contemporary state-of-the-art algorithms, these solutions demonstrate their competitiveness in providing efficient solutions to these test problems while exhibiting faster convergence.
- Since both ESGO and EMSGO are modified algorithms, a comparison is conducted between these two algorithms and six recently introduced improved/hybrid algorithms.
- To further assess the capabilities of ESGO and EMSGO, they are applied to tackle 26 chemical and mechanical design problems. In most cases, the algorithms yield optimal solutions for these real-world challenges.

The subsequent sections of this paper are organized as follows: Section 2 presents a comprehensive review of the fundamental concepts of SGO and MSGO. Section 3 provides a detailed description of the proposed ESGO and EMSGO algorithms. Section 4 verifies the efficacy of the improved strategies and the superiority of the modified algorithms by conducting experiments using classical test functions and addressing real-world optimization problems. Finally, in Section 5, the conclusions are presented, and avenues for future research are explored.

2 Social group optimization and modified social group optimization

2.1 Social group optimization (SGO)

The SGO algorithm, which is intended to handle complex problems, takes inspiration from human social behaviour. According to this algorithm, members of a social group are potential solutions since they each have the knowledge and abilities needed to solve the given problem. Individuals' human characteristics match the dimensions of the design factors in the problem. The Enhancing Phase and the Acquiring Phase are the two stages of the optimization process.

Consider a social group denoted as P_i, with i ranging from 1 to pop_size, representing the group's individuals. Each individual, P_i, is characterized by traits (p_i1, p_i2, p_i3, ..., p_iD), where D signifies the defining dimensions. Every individual is associated with a fitness value, f_i, reflecting their fitness levels.

Phase 1: Improving Phase

In this phase, the best individual, 'gbest,' shares knowledge with the entire group, enhancing their collective knowledge. Each individual updates their information based on 'gbest' according to the formula:

$$Pnew\_i = c * P\_i + r * (gbest - P\_i) \quad (1)$$

The new solution, Pnew_i , is accepted only if it improves fitness. Here, 'r' is a random number from U(0, 1), and 'c' is the self-introspection parameter (c=0.2).

Phase 2: Acquiring Phase

In this phase, an individual interacts with the best performer (best_P) and engages in random interactions with other group members to acquire knowledge. The update is determined by:

```
    Randomly select one person P_r, where i ≠r

If f(P_i) < f(P_r)
        Pnew_i = P_i + r_1*(P_i - P_r) + r_2 * (best_P -
                P_i)
Else
        Pnew_i = P_i + r_1 * (P_r - P_i) + r_2 * (best_P -
                P_i)
End if                                          (2)
```

The new solution is accepted if it enhances fitness. Here, r_1, r_2, and r_3 are random numbers from U(0, 1), introducing stochasticity. 'lb' and 'ub' represent lower and upper bounds of design variables. For detailed insights, refer to the paper [30].

2.2 Modified social group optimization (MSGO)

In the MSGO, the Improving Phase remains same as like SGO. Only Acquiring Phase has been modified in the following manner:

Phase 2: Acquiring Phase

A social group member engages in interactions with the best performer (best_P) in the same group during the Acquiring Phase. In order to learn, they simultaneously strike up conversations at random with other group members. When the other person knows more than the interacting person does, and the interacting person has a higher Self-Awareness Probability (SAP) of learning that knowledge, new knowledge is acquired. SAP is a measure of an individual's ability to learn from others. The following is an outline of the Acquiring Phase:

```
For i = 1 to pop_size
    Randomly select one person P_r, where i ≠ r
    If f(P_i) < f(P_r)
        If rand > SAP
            Pnew_i = P_i + r_1 * (P_i - P_r) + r_2 *
            (best_P - P_i)
        Else
            Pnew_i = lb + r_3 * (ub - lb)
        end if
    Else
        Pnew_i = P_i + r_1 * (P_r - P_i)+ r_2 *
            (best_P - P_i)

    End If
End for                                     (3)
```

The acceptance of the new solution, Pnew_i , is contingent upon its ability to yield enhanced fitness relative to the current solution. In this context, r_1, r_2, and r_3 denote three independent random numbers drawn from a uniform distribution U(0, 1), introducing stochastic elements into the algorithm. The terms 'lb' and 'ub' represent the lower and upper bounds of the corresponding design variable, and the SAP is fixed at 0.7. For a more in-depth understanding of the SGO algorithm, please consult the referenced paper [15].

3 Proposed ESGO (Enhanced SGO) and EMSGO (Enhanced MSGO)

3.1 ESGO, and EMSGO algorithms

To enhance or refine a skill or ability to a higher level of effectiveness in the SGO algorithm, the Improving phase undergoes the following modifications.

Initially, a subset of the best-performing individuals (gbest persons) is selected from the social group. The knowledge level of each individual within this group is then enhanced through interactions with these superior (gbest) persons. As the iterations progress, the number of members in the gbest group diminishes. Eventually, only one dominant (gbest) individual remains within the social group. This modification of improving phase is adapted in the following manner:

1) Calculate the number of gbest individuals (NG): NG = floor(0.1 * pop_size * (1 - iter / max_iter)) + 1
2) Sort the fitness values in descending order and store the corresponding best values:
   [value_best] = sort(f, 'descend')
3) For each of the top NG individuals (indexed as j):

```
for j = 1:NG
    GG_j = P_best_j          (4)
    GGf_j = value_j
End
```

```
4) Iterate through the entire population (i = 1 to pop_size):
   For i = 1 to pop_size
       Randomly select one individual, GG_r.

   If the fitness value of GG_r (GGf_r) < fitness
      value of the current individual (f_i):
         Pnew_i = c * P_i + r_4 * (GG_r - P_i)      (5)
   Else
         Pnew_i = c * P_i + r_5 * (P_i - GG_r)
   End if
```

Only accept the new solution, Pnew_i, if it results in a higher level of fitness than the existing solution. Here, r_4 and r_5 are two independent random numbers drawn from a uniform distribution U(0, 1).

The ESGO and EMSGO algorithms are derived by replacing Improving phase by the above modified Improving phase into the SGO and EMSGO algorithms, respectively.

3.2 Phases of ESGO and EMSGO algorithm, Exploration and Exploitation concept

Each of the two phases constituting the ESGO and EMSGO algorithms emphasizes distinct aspects of exploration and exploitation within the optimization framework.

a) *Improving Phase (Exploration Emphasis):* Initially, a subset of the top-performing individuals (referred to as "gbest persons") is chosen from the social group. The individuals then undergo a process of knowledge enhancement through interactions with these superior gbest persons. As the iterations progress, the size of the gbest group gradually diminishes, eventually leaving only one dominant gbest individual within the social group. This adjustment in the improving phase facilitates improved knowledge transfer among individuals, thereby enhancing their exploration capabilities more rapidly. Throughout this phase, the primary focus is on exploration.

b) *Acquiring Phase (Transition to Exploitation):* A social group member participates in a discussion with the group's top performer during this period. They also strike up conversations at random with other group members in an effort to learn more. When interacting with someone who knows more, an individual absorbs new information because they are more likely to have a higher Self-Awareness Probability (SAP) for learning that information. SAP is a measure of an individual's ability to learn from

c) others. At this point, people start using the knowledge they have learned for optimization, marking the shift from exploration to exploitation.

3.3 Discussion of computational complexity of ESGO and EMSGO algorithms

BigO (TSD + TNC_f), where T is the number of iterations, S is the population size or the number of agents, C_f is the cost of function evaluation, and D is the problem dimension, represents the computational complexity of SGO.

In analysing the temporal complexity of most algorithms, three main factors are usually taken into account:

a. BigO (SD) computational complexity is usually associated with population initialization, where S denotes population size and D denotes problem dimension.
b. BigO (SC_f) is frequently used to limit the computational cost of the initial function evaluation (FE).
c. BigO (TSD + TNC_f) usually sets a limit on the main loop's computational complexity.

Consequently, the overall computational complexity of the ESGO and EMSGO algorithms remains the same as the SGO algorithm, as both the ESGO and EMSGO algorithms are derived by introducing a modified improving phase, where the computational complexity of the modified improving phase is BigO (TSD + TNC_f).

It follows that SGO, EMSGO, and EMSGO have similar computational complexity, which is represented by the notation BigO (TSD + TNC_f).

Algorithm 1 gives the pseudo-code for the proposed ESGO and EMSGO algorithm.

Algorithm 1: Pseudo code of ESGO /EMSGO algorithm

```
1. Set up the search agent population (persons).
2. Algorithm parameters definition: C, SAP
3. While iter < Max_iter:
     Determine the best current solution by doing fitness calculations
     Select the best solutions and make gbest persons group.
      Within the population (pop_size), for every agent i:
           Update the person using Equation 5.
      End for.
    Perform fitness calculations and update the current position of persons
          Update the current best solution.

     Within the population (pop_size), for every agent i:
          Update the person using Equation 2 for ESGO algorithm/ Update the person using Equation 3 for EMSGO algorithm
     End for
   Perform fitness calculations and update the current position of persons
   Update the current best solution.

4. End while.
```

4 Simulation, experimental result, and discussions

The performance of the ESGO and EMSGO algorithms is demonstrated in this paper through four experiments. In experiment 1, both algorithms are compared with each other and also with their original algorithm SGO and MSGO respectively. In the second experiment, the performances of both algorithms are compared with twelve state-of-the-art algorithms such as African vultures optimization algorithm (AVOA) [48], DE [49], Exponential distribution optimizer (EDO)[50], GWO [51], Kepler optimization algorithm (KOA) [52], Light Spectrum Optimizer (LSO) [53], Mantis Search Algorithm (MSA)[54], Nutcracker optimizer algorithm(NOA) [55], Reptile Search Algorithm (RSA) [56], Slime mould algorithm (SMA) [57], Spider wasp optimizer(SWO)[58], and WOA [23]. In the 3rd experiment, the performance of both algorithms is compared with six improved/hybrid recently introduced algorithms. In the 4th experiment, both algorithms show their performance in solving twenty-six real-world constrained optimization problems of mechanical and chemical design problems.

Every novel optimization algorithm must undergo rigorous evaluation using well-defined benchmark functions to assess and validate its performance. Although there are numerous benchmark functions available, however there is no standardized set of benchmarks that are agreed upon for evaluating new algorithms. In order to validate and benchmark the performance of our proposed ESGO and EMSGO algorithms, the simulations are conducted on a set of twenty-three benchmark functions. These carefully selected benchmark functions serve as a comprehensive testbed for assessing various aspects of the algorithms, including their ability to achieve rapid convergence, escape local optima, and prevent premature convergence. The selection of these benchmark functions is motivated by their widespread use in existing literature [57, 59-63]. Out of the twenty-three functions, seven are unimodal benchmark functions (F1–F7), ideal for benchmarking the exploitation capabilities of algorithms due to their single global optimum. Six are multimodal benchmark functions, while ten are fixed-dimensional multimodal benchmark functions. Each of the multimodal functions, from F8 to F23, contains a multitude of local optima, making them well-suited for evaluating the exploration capabilities of algorithms. For a comprehensive understanding of these benchmark functions, detailed descriptions can be found in reference [9], and graphical representations are provided in Figure 1. Experiments 1-3 use these benchmark functions to validate the performance comparisons among algorithms. The detailed descriptions of twenty-six real-world constrained optimization problem of mechanical and chemical design problems are given in [64] which is used in experiment 4 to validate the performance algorithms. Implemented on the Windows 10 operating system, MATLAB 2016a is employed to execute all algorithms. The simulations are conducted on a laptop equipped with an Intel Core i5 processor and 8 GB of memory.

4.1 Algorithm validation

To assess the performance of the ESGO and EMSGO algorithms, a set of 23 benchmark functions is utilized, with results compared against twelve different metaheuristic algorithms, as previously outlined. In Experiment 1, a comparative analysis is conducted between ESGO, SGO, MSGO, and EMSGO, with the results presented in Table 2. In Experiment 2, the modified algorithms are compared with the twelve other algorithms, and the comparative outcomes are showcased in Table 3. Similarly, in Experiment 3, ESGO and EMSGO are compared with six recently introduced improved/hybrid algorithms, with the results imported in Table 5. Throughout these experiments, a fixed parameter, max_FEs, is maintained at 15,000. Consequently, the number of iterations and population size may vary for different algorithms. The parameter configurations for the algorithms align with widely accepted settings utilized by various researchers, as detailed in Table 1.

Experiment 1: The performance comparison of the SGO family of algorithms

In this specific experiment, the performance of the SGO algorithm family, which includes ESGO, SGO, EMSGO, and MSGO, is assessed through comparative analysis. To ensure the robustness and statistical significance of the findings, the experiment is conducted 30 times. The outcomes are presented in Table 2, detailing key metrics such as the best (BEST), worst (WORST), average (MEAN), and standard deviation (SD) of fitness solutions. Noteworthy results are highlighted in bold within the tables, and the symbol '‖' denotes that the value remains consistent with the preceding row.

Discussion

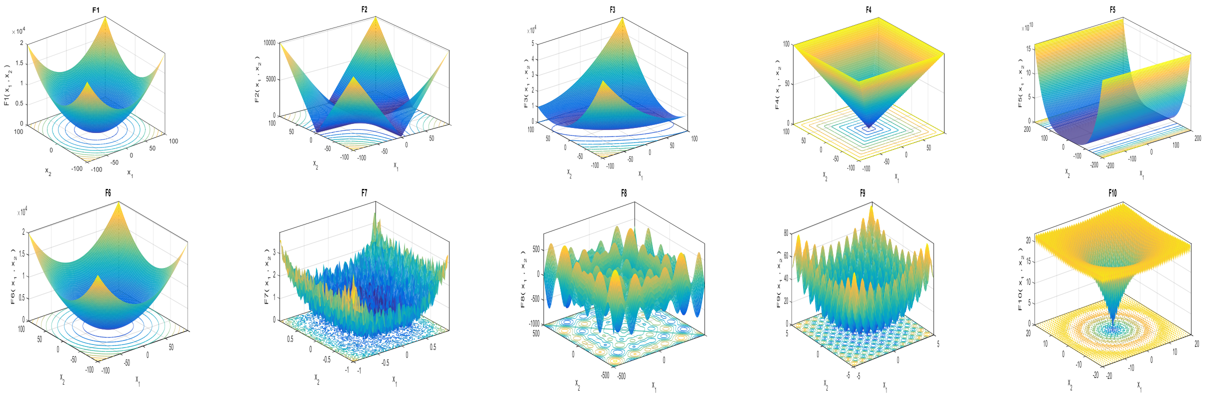
It is seen from Table 2 that the ESGO algorithm reaches the global optimum for twelve functions, only best solution reaches optimal solution in four cases, and in four cases find good solutions in compare to others. The EMSGO algorithm reaches the global optimum for fifteen functions, only best solution reaches optimal solution in one case, and in three cases find good solutions in compare to others. The MSGO algorithm reaches the global optimum for thirteen functions, and only best solution reaches optimal solution in two cases. Similarly, the SGO algorithm reaches the global optimum for eight functions, only best solution reaches optimal solution in three cases, and in one case find good solutions in compare to others. Hence, it can be concluded that both the ESGO and EMSGO algorithms achieve improved results.

Table 1: Parameter setting of algorithms compared to the SGO and MSGO algorithms

Sl. No.	Algorithms	Parameters	Values
1	SGO	C	0.2
	MSGO	C SAP	0.2 0.7
2	GWO	Control parameter	[2, 0]
3	AVOA	p1 p2 p3 alpha betha gamma	0.6 0.4 0.6 0.8 0.2 2.5
4	SMA	Parameter	0.03

5	EDO	f= 2*rand-1
		a=f^10
		b=f^5
		c=d*f	

6	DE	F Cr	0.5 0.5
7	KOA	Tc M0 lambda	3 0.1 15
8	LSO	Ps Pe Ph B	0.05 0.6 0.4 0.05
9	MSA	p A a P Alp Pc	0.5 1.0 0.5 2 6 0.2
10	NOA	Alpha Pa2 Prb	0.05 0.2 0.2
11	RSA	Alpha value Beta	0.1 0.1 0.1
12	SWO	TR Cr N_min=20	0.3. 0.2 20
13	WOA	Spiral updating probability Shrinking encircling Random search ability	0.5 0.5 0.1
14	ESGO	C	0.2
15	EMSGO	C SAP	0.2 0.7



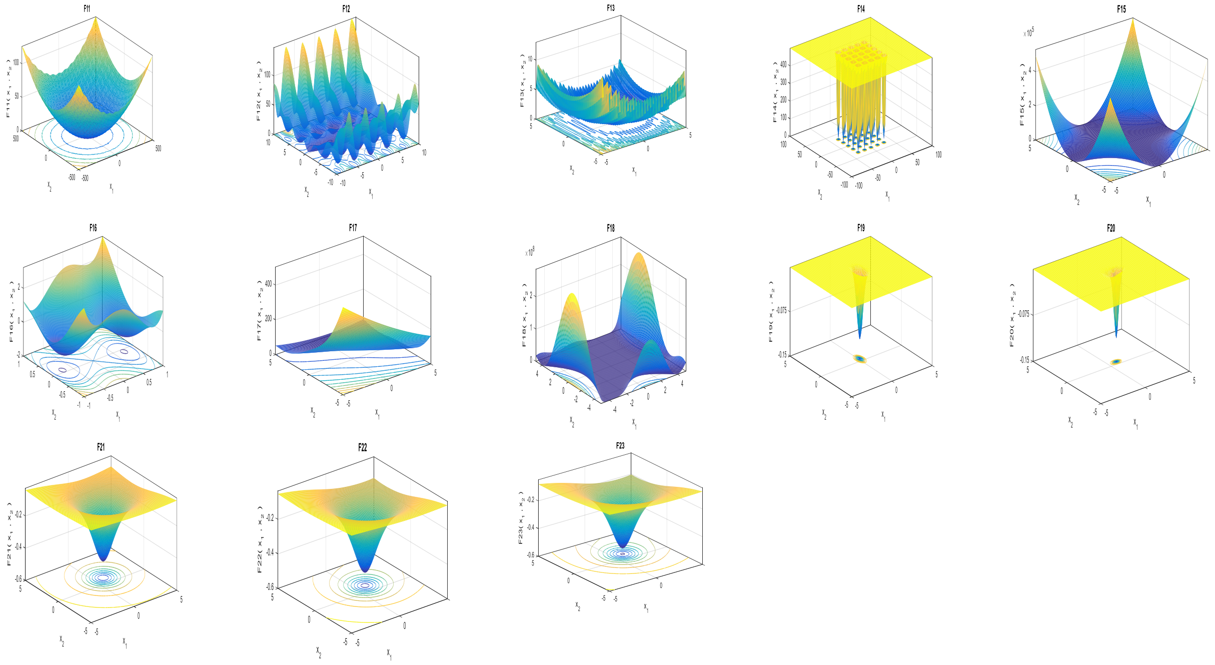


Figure 1: Graphical representation of classical benchmark functions

Table 2: Comparison results of family of SGO algorithms

Algo/Functions		F1	F2	F3	F4	F5	F6
ESGO	BEST	**0**	4.1803e-199	8.2277e-149	**5.9334e-140**	**3.4173e-09**	**0**
	WORST	**0**	1.9582e-189	3.3997e-140	**7.4641e-135**	**2.8200e-08**	**1.8489e-32**
	MEAN	**0**	1.9721e-190	9.1732e-145	**1.4414e-135**	**1.3524e-08**	**4.9304e-33**
	STD	**0**	**0**	1.2569e-146	**2.4817e-135**	**8.3790e-09**	**7.4354e-33**
SGO	BEST	9.2725e-206	1.5010e-103	**1.2056e-205**	1.1175e-103	25.2399	1.4422e-05
	WORST	1.4198e-205	1.8192e-103	**2.8262e-205**	1.2648e-103	26.5470	7.4685e-04
	MEAN	1.2120e-205	1.6575e-103	**2.1541e-205**	1.2011e-103	25.9628	1.7258e-04
	STD	**0**	1.0090e-104	**0**	4.4263e-105	0.3815	2.4597e-04
EMSGO	BEST	**0**	**1.1289e-259**	8.2056e-239	2.5451e-130	0	4.1842e-05
	WORST	**0**	**2.4844e-196**	5.7833e-179	2.1033e-125	1.2171e-04	5.3231e-04
	MEAN	**0**	**2.5618e-197**	5.7883e-180	2.3576e-126	1.4548e-05	2.4457e-04
	STD	**0**	**0**	0	6.5915e-126	3.8171e-05	1.7999e-04
MSGO	BEST	3.0864e-212	2.7434e-106	1.6631e-207	7.0946e-106	0	2.4271e-06
	WORST	1.1362e-205	1.7802e-103	6.3071e-195	4.8986e-104	0.1337	0.0215
	MEAN	4.1136e-206	7.9570e-104	6.3071e-196	1.8810e-104	0.0244	0.0112
	STD	**0**	5.4012e-104	**0**	1.5099e-104	0.0518	0.0075

Algo/Functions		F7	F8	F9	F10	F11	F12
ESGO	BEST	2.0781e-05	-1.0832e+04	0	**8.8818e-16**	**0**	**1.5705e-32**
	WORST	8.1608e-05	-7.5157e+03	28.8538	**8.8818e-16**	0.0123	**1.5705e-32**
	MEAN	4.3626e-05	-9.0179e+03	17.3123	**8.8818e-16**	0.0012	**1.5705e-32**
	STD	2.0376e-05	1.1567e+03	10.2887	**0**	0.0039	**2.8850e-48**
SGO	BEST	5.0521e-06	-9.6243e+03	0.0039	'‖'	**0**	2.0146e-06
	WORST	2.5723e-04	-5.7770e+03	0.0039		**0**	4.4800e-05
	MEAN	1.2684e-04	-7.6685e+03	0.0039		**0**	1.0564e-05
	STD	8.3738e-05	1.4999e+03	0		**0**	1.2502e-05
EMSGO	BEST	**9.1415e-06**	**-1.2569e+04**	**0**	'‖'	'‖'	1.0378e-07
	WORST	**6.1383e-05**	**-1.2569e+04**	**0**			2.9400e-05

	MEAN	**3.6674e-05**	**-1.2569e+04**	**0**			8.4505e-06
	STD	**2.0307e-05**	**5.9041e-05**	**0**			1.1504e-05
MSGO	BEST	1.2110e-05	-1.2569e+04	'‖'	'‖'	'‖'	7.5531e-07
	WORST	1.6058e-04	-1.2569e+04				7.5531e-07
	MEAN	8.3206e-05	-1.2569e+04				7.0659e-05
	STD	6.1619e-05	0.0559				1.7495e-04

Algo/Functions		**F13**	**F14**	**F15**	**F16**	**F17**	**F18**
ESGO	BEST	1.3498e-32	**0.9980**	**3.0749e-04**	**-1.0316**	**0.3979**	**3.0000**
	WORST	0.0110	**0.9980**	**3.0749e-04**	**-1.0316**	**0.3979**	**3.0000**
	MEAN	0.0033	**0.9980**	**3.0749e-04**	**-1.0316**	**0.3979**	**3.0000**
	STD	0.0053	**0**	**1.0537e-19**	**0**	**0**	**1.0978e-15**
SGO	BEST	3.2121e-05	0.9980	3.0749e-04	'‖'	'‖'	**3.0000**
	WORST	0.0979	0.9980	3.1132e-04			**3.0000**
	MEAN	0.0110	0.9980	3.0867e-04			**3.0000**
	STD	0.0307	7.4015e-17	1.3701e-06			**5.1279e-16**
EMSGO	BEST	**1.0838e-09**	**0.9980**	**3.0749e-04**	'‖'	'‖'	**3.0000**
	WORST	**9.8844e-05**	**0.9980**	**3.0749e-04**			**3.0000**
	MEAN	**2.5227e-05**	**0.9980**	**3.0749e-04**			**3.0000**
	STD	**3.6575e-05**	**0**	**1.0537e-20**			**0**
MSGO	BEST	4.0152e-06	'‖'	**3.0749e-04**	'‖'	'‖'	**3.0000**
	WORST	0.0110		7.7817e-04			**3.0000**
	MEAN	0.0024		5.0234e-04			**3.0000**
	STD	0.0037		2.0656e-04			**2.1251e-16**

Algo/Functions		F19	F20	**F21**	**F22**	**F23**
ESGO	BEST	**-3.8628**	-3.3220	-10.1532	**-10.4029**	**-10.5364**
	WORST	**-3.8628**	-3.2031	-10.1532	**-10.4029**	**-10.5364**
	MEAN	**-3.8628**	-3.2982	-10.1532	**-10.4029**	**-10.5364**
	STD	**9.3622e-16**	0.0501	1.3240e-15	**1.6748e-15**	**1.32149e-15**
SGO	BEST	'‖'	-3.3220	-5.0552	-10.4029	-5.1756
	WORST		-3.2031	-5.0552	-5.0877	-5.1285
	MEAN		-3.2863	-5.0552	-5.6192	-5.1332
	STD		0.0574	**0**	1.6808	0.0149
EMSGO	BEST	'‖'	**-3.3220**	**-10.1532**	**-10.4029**	**-10.5364**
	WORST		**-3.3220**	**-10.1532**	**-10.4029**	**-10.5364**
	MEAN		**-3.3220**	**-10.1532**	**-10.4029**	**-10.5364**
	STD		**9.2038e-09**	**0**	**1.8724e-15**	**2.7773e-15**
MSGO	BEST	'‖'	-3.3220	-10.1532	**-10.4029**	**-10.5364**
	WORST		-3.2031	-10.1532	**-10.4029**	**-10.5364**
	MEAN		-3.2863	-10.1532	**-10.4029**	**-10.5364**
	STD		0.0574	1.3240e-15	**1.6748e-15**	**2.1349e-15**

Experiment 2: The performance comparison with state-of-the-art metaheuristics algorithms

Based on the results obtained from Experiment 1, it is evident that ESGO and EMSGO exhibit superior performance in terms of fitness function evaluation when compared to other algorithms. Consequently, in this experiment, ESGO and EMSGO are subjected to a comprehensive comparison with the remaining twelve algorithms to validate their performance. The experiment is repeated 30 times, with the statistical results—including the BEST, WORST, MEAN, and SD of fitness solutions presented in Table 3. This rigorous analysis is designed to ensure stability and establish statistical significance, with the most remarkable results highlighted in bold in the table, and the symbol '‖' indicating that its value is equivalent to the value in the preceding column. Table 4 reports the p-values derived from the WRS test [65] at a significance level of 5% for ESGO(E) versus other approaches and EMSGO(EM) versus other approaches. When p-values fall below 0.05, it indicates a rejection of the null hypothesis, while "N" signifies that the input values are similar. Additionally, in Table 4, "-" indicates that the performance of other approaches is inferior, "+" signifies it is superior, and "S" suggests a similar performance when compared to ESGO and EMSGO.

Table 3: Comparison Results of ESGO, EMSGO and other algorithms

Algo/Functions		F1	F2	F3	F4	F5	F6
ESGO	BEST	**0**	5.2836e-199	3.5401e-46	3.1051e-142	**7.4975e-10**	0
	WORST	**0**	3.1359e-190	2.6954e-42	1.9962e-134	**2.4696e-08**	0
	MEAN	**0**	3.7878e-191	8.1478e-43	2.7076e-135	**8.9214e-09**	0
	SD	**0**	**0**	1.0752e-42	6.1875e-135	**9.9187e-09**	0
EMSGO	BEST	**0**	6.9340e-263	2.4211e-243	7.1047e-133	0	9.7104e-07
	WORST	**0**	1.7585e-199	3.1882e-186	4.3710e-125	0.0032	3.8264e-04
	MEAN	**0**	1.8564e-200	3.1982e-187	8.3185e-126	4.2652e-04	1.0457e-04
	SD	**0**	0	0	1.6478e-125	0.0010	1.4912e-04
AVOA	BEST	7.7025	0.0334	53.4364	0.0490	30.7586	37.9652
	WORST	2.3803e+04	72.2977	6.9949e+04	67.0993	8.7707e+07	2.6602e+04
	MEAN	8.9613e+03	46.2646	3.8489e+04	38.0639	2.9522e+07	8.2425e+03
	SD	1.0486e+04	26.2007	2.6043e+04	26.1463	3.8485e+07	1.1002e+04
	P-value(E)	6.3864e-05	1.8267e-04	1.8267e-04	1.8267e-04	1.8267e-04	6.3864e-05
	P-value(EM)	8.7450e-05	'‖'	'‖'	'‖'	1.7265e-04	1.8267e-04
DE	BEST	4.4059e+04	163.1432	8.4222e+04	80.8065	1.9217e+08	4.2435e+04
	WORST	6.5836e+04	1.7386e+10	1.0999e+05	89.0112	2.3023e+08	6.2838e+04
	MEAN	5.8571e+04	3.4780e+09	9.4515e+04	85.4072	2.0369e+08	5.6172e+04
	SD	6.4844e+03	5.8684e+09	8.7802e+03	2.5891	1.1840e+07	6.4762e+03
	P-value(E)	6.3864e-05	1.8267e-04	1.8267e-04	1.8267e-04	1.8267e-04	6.3864e-05
	P-value(EM)	8.7450e-05	'‖'	'‖'	'‖'	1.7265e-04	1.8267e-04
EDO	BEST	1.4834e-102	5.0862e-56	0	1.5668e-51	28.7129	0.5508
	WORST	8.0060e-85	1.2305e-40	5.2972e-76	3.9187e-40	28.7434	1.1080
	MEAN	1.0280e-85	1.2404e-41	5.5843e-77	3.9239e-41	28.7295	0.8615
	SD	2.5035e-85	3.8879e-41	1.6674e-76	1.2390e-40	0.0105	0.1757
	P-value(E)	6.3864e-05	1.8267e-04	1.8267e-04	1.8267e-04	1.8267e-04	6.3864e-05
	P-value(EM)	8.7450e-05	'‖'	0.0028	'‖'	1.7265e-04	1.8267e-04
GWO	BEST	1.1432e+04	45.8397	1.9213e+04	45.0504	1.1085e+07	1.0877e+04
	WORST	1.9109e+04	1.0544e+03	4.3169e+04	56.7561	2.6838e+07	1.8265e+04
	MEAN	1.4651e+04	209.3304	3.1801e+04	51.0770	1.7175e+07	1.5388e+04
	SD	2.7088e+03	310.0702	7.1282e+03	4.0340	5.5991e+06	2.7820e+03
	P-value(E)	6.3864e-05	1.8267e-04	1.8267e-04	1.8267e-04	1.8267e-04	6.3864e-05
	P-value(EM)	8.7450e-05	'‖'	'‖'	'‖'	1.7265e-04	1.8267e-04
KOA	BEST	5.4244e+04	2.5173e+06	6.4049e+04	77.5024	1.6814e+08	4.2841e+04
	WORST	6.5658e+04	4.3553e+10	1.2905e+05	86.0408	2.2523e+08	6.7207e+04
	MEAN	6.0795e+04	1.3480e+10	9.6211e+04	83.4309	2.0486e+08	5.8797e+04
	SD	3.7316e+03	1.5055e+10	2.6865e+04	2.6637	1.9144e+07	8.3548e+03
	P-value(E)	6.3864e-05	1.8267e-04	1.8267e-04	1.8267e-04	1.8267e-04	6.3864e-05
	P-value(EM)	8.7450e-05	'‖'	'‖'	'‖'	1.7265e-04	1.8267e-04
LSO	BEST	**0**	**0**	**0**	**0**	29	7.5000
	WORST	**0**	**0**	**0**	**0**	29	7.5000
	MEAN	**0**	**0**	**0**	**0**	29	7.5000
	SD	**0**	**0**	**0**	**0**	0	0
	P-value(E)	NaN	6.3864e-05	6.3864e-05	6.3864e-05	6.3864e-05	1.5938e-05
	P-value(EM)	0.3681	'‖'	'‖'	'‖'	5.9363e-05	6.3864e-05
MSA	BEST	3.8432e+04	1.0814e+05	4.1720e+04	69.9315	8.1597e+07	4.0168e+04
	WORST	3.8432e+04	1.0814e+05	4.1720e+04	69.9315	8.1597e+07	4.0168e+04
	MEAN	4.3015e+04	2.0010e+06	5.2453e+04	72.9467	1.0748e+08	4.4521e+04

	SD	2.6972e+03	2.2635e+06	5.9061e+03	1.8691	1.6092e+07	2.4725e+03
	P-value(E)	6.3864e-05	1.8267e-04	1.8267e-04	1.8267e-04	1.8267e-04	6.3864e-05
	P-value(EM)	8.7450e-05	'‖'	'‖'	'‖'	1.7265e-04	1.8267e-04
NOA	BEST	4.8660e+04	2.9131e+08	7.3451e+04	77.6661	1.4000e+08	5.1567e+04
	WORST	6.6244e+04	8.0625e+10	1.3115e+05	86.9414	2.2977e+08	6.3968e+04
	MEAN	5.7156e+04	2.3403e+10	1.0086e+05	83.1270	1.8809e+08	5.8477e+04
	SD	6.0233e+03	2.7126e+10	2.0704e+04	3.2065	3.0255e+07	3.7992e+03
	P-value(E)	6.3864e-05	1.8267e-04	1.8267e-04	1.8267e-04	1.8267e-04	6.3864e-05
	P-value(EM)	8.7450e-05	'‖'	'‖'	'‖'	1.7265e-04	1.8267e-04
RSA	BEST	**0**	**0**	**0**	**0**	29	7.5000
	WORST	**0**	**0**	**0**	**0**	29	7.5000
	MEAN	**0**	**0**	**0**	**0**	29	7.5000
	SD	**0**	**0**	**0**	**0**	0	0
	P-value(E)	NaN	6.3864e-05	6.3864e-05	6.3864e-05	6.3864e-05	1.5938e-05
	P-value(EM)	0.3681	'‖'	'‖'	'‖'	'‖'	'‖'
SMA	BEST	4.1063e-05	0.0080	0.0037	0.1079	28.9931	7.1919
	WORST	0.1126	0.5269	292.4747	0.3502	30.1790	9.2486
	MEAN	0.0402	0.1431	71.5265	0.1975	29.4534	7.7576
	SD	0.0342	0.1830	98.6191	0.0770	0.3974	0.6508
	P-value(E)	6.3864e-05	1.8267e-04	1.8267e-04	1.8267e-04	1.8165e-04	6.3864e-05
	P-value(EM)	'‖'	'‖'	'‖'	'‖'	'‖'	1.8267e-04
SWO	BEST	4.8980e+04	3.6923e+05	5.6283e+04	78.8348	1.6983e+08	5.3525e+04
	WORST	6.3486e+04	7.6522e+10	1.0584e+05	84.7210	2.3214e+08	6.6370e+04
	MEAN	5.8234e+04	1.3004e+10	8.6366e+04	82.0490	1.9669e+08	5.8365e+04
	SD	4.7426e+03	2.4953e+10	1.4348e+04	2.0315	2.2956e+07	4.1333e+03
	P-value(E)	6.3864e-05	1.8267e-04	1.8267e-04	1.8267e-04	1.8267e-04	6.3864e-05
	P-value(EM)	'‖'	'‖'	'‖'	'‖'	'‖'	1.8267e-04
WOA	BEST	1.0763e+04	37.4285	7.9057e+04	53.3947	4.6152e+06	8.5596e+03
	WORST	2.2247e+04	171.652	1.5291e+05	86.0943	6.9715e+07	3.7200e+04
	MEAN	1.6084e+04	80.5780	1.1317e+05	76.7905	2.7538e+07	1.8897e+04
	SD	4.4502e+03	42.3134	2.3395e+04	10.7489	2.0558e+07	8.2834e+03
	P-value(E)	6.3864e-05	1.8267e-04	1.8267e-04	1.8267e-04	1.8267e-04	6.3864e-05
	P-value(EM)	'‖'	'‖'	'‖'	'‖'	'‖'	1.8267e-04

Algo/Functions		F7	F8	F9	F10	F11	F12
ESGO	BEST	4.3625e-05	-1.1089e+04	**0**	8.8818e-16	0	**1.5705e-32**
	WORST	1.3583e-04	-7.9700e+03	23.8790	8.8818e-16	0	**1.6109e-32**
	MEAN	6.9865e-05	-8.8463e+03	12.1385	8.8818e-16	0	**1.5802e-32**
	SD	2.9670e-05	962.9930	9.0560	**0**	0	**1.5117e-34**
EMSGO	BEST	**2.3125e-05**	**-1.2569e+04**	**0**	**8.8818e-16**	**0**	7.0940e-12
	WORST	**6.3363e-05**	**-1.2569e+04**	**0**	**8.8818e-16**	**0**	9.1524e-06
	MEAN	**4.3487e-05**	**-1.2569e+04**	**0**	**8.8818e-16**	**0**	1.8881e-06
	SD	**1.3955e-05**	**7.2809e-04**	**0**	**0**	**0**	2.9886e-06
AVOA	BEST	0.2208	-4.1571e+03	55.3583	1.1428	1.0421	0.2444
	WORST	23.8537	-3.2544e+03	325.4436	12.2769	229.6541	5.5755e+07
	MEAN	11.7494	-3.6179e+03	217.0793	6.9852	44.9105	5.7285e+06
	SD	9.9263	256.5975	108.4945	3.8592	89.3992	1.7581e+07
	P-value(E)	1.8267e-04	1.8267e-04	6.3864e-05	6.3864e-05	6.3864e-05	1.4939e-04
	P-value(EM)	'‖'	'‖'	'‖'	'‖'	'‖'	'‖'
DE	BEST	66.7799	-3.3978e+03	335.9169	20.4348	462.1846	2.8041e+08
	WORST	108.1115	-2.3055e+03	415.4900	20.6292	588.1518	5.4382e+08
	MEAN	93.8855	-2.7284e+03	398.0476	20.5596	516.8565	4.0688e+08
	SD	12.6976	315.6260	22.9089	0.0620	37.1096	8.8929e+07
	P-value(E)	1.8267e-04	1.8267e-04	1.7861e-04	6.3864e-05	6.3864e-05	1.4939e-04
	P-value(EM)	'‖'	'‖'	6.3864e-05	'‖'	'‖'	1.8267e-04
EDO	BEST	5.6818e-05	-1.2566e+04	**0**	**8.8818e-16**	**0**	0.0239

	WORST	7.1180e-04	-1.2352e+04	**0**	**8.8818e-16**	**0**	0.0969
	MEAN	2.8708e-04	-1.2489e+04	**0**	**8.8818e-16**	**0**	0.0647
	SD	2.2037e-04	69.8756	**0**	**0**	**0**	0.0246
	P-value(E)	0.0017	1.8267e-04	0.00	NaN	NaN	1.4939e-04
	P-value(EM)	4.3964e-04	'||'	NaN	NaN	NaN	1.8267e-04
GWO	BEST	3.9947	-3.8122e+03	231.6198	16.0005	106.2397	1.0871e+06
	WORST	11.1918	-2.2596e+03	311.4545	18.2003	185.9527	4.0395e+07
	MEAN	6.2076	-2.8373e+03	272.2285	17.4764	147.1419	1.5626e+07
	SD	2.4455	574.3139	30.7015	0.6500	28.2347	1.4349e+07
	P-value(E)	1.8267e-04	1.8267e-04	1.7861e-04	6.3864e-05	6.3864e-05	1.4939e-04
	P-value(EM)	1.8267e-04	1.8267e-04	1.1067e-04	6.3864e-05	6.3864e-05	1.8267e-04
KOA	BEST	77.6168	-5.4177e+03	367.0005	19.9668	387.9851	3.1296e+08
	WORST	112.4230	-5.4177e+03	424.7784	19.9668	575.4793	5.0309e+08
	MEAN	97.2831	-5.4177e+03	399.6969	19.9668	516.5733	4.3850e+08
	SD	12.8711	9.5869e-13	22.5140	**0**	54.2910	6.1302e+07
	P-value(E)	1.8267e-04	6.3864e-05	1.7861e-04	1.5938e-05	6.3864e-05	1.4939e-04
	P-value(EM)	'||'	'||'	1.1067e-04	'||'	'||'	1.8267e-04
LSO	BEST	0.0258	-5.0565e+03	**0**	**8.8818e-16**	**0**	1.6690
	WORST	0.1386	-2.3825e+03	**0**	**8.8818e-16**	**0**	1.6690
	MEAN	0.0685	-3.1907e+03	**0**	**8.8818e-16**	**0**	1.6690
	SD	0.0412	1.0299e+03	**0**	**0**	**0**	2.3406e-16
	P-value(E)	1.8267e-04	1.8165e-04	0.0022	NaN	NaN	4.9177e-05
	P-value(EM)	'||'	'||'	NaN	NaN	NaN	6.3864e-05
MSA	BEST	45.5181	-4.4134e+03	336.7071	19.3659	366.8070	9.5920e+07
	WORST	45.5181	-4.4134e+03	336.7071	19.3659	366.8070	9.5920e+07
	MEAN	51.7201	-4.0591e+03	346.9462	19.7710	398.6249	1.7816e+08
	SD	4.7908	195.7146	6.9562	0.1759	26.6722	3.8232e+07
	P-value(E)	1.8267e-04	1.8267e-04	1.7861e-04	6.3864e-05	6.3864e-05	1.4939e-04
	P-value(EM)	'||'	'||'	1.1067e-04	'||'	'||'	1.8267e-04
NOA	BEST	79.4873	-5.4177e+03	364.2811	19.9668	471.3946	2.5340e+08
	WORST	120.1875	-5.4177e+03	434.5688	19.9668	592.1890	5.7029e+08
	MEAN	103.4432	-5.4177e+03	403.0857	19.9668	565.1993	4.2632e+08
	SD	14.0245	9.5869e-13	25.9011	**0**	35.3447	1.1130e+08
	P-value(E)	1.8267e-04	6.3864e-05	1.7861e-04	1.5938e-05	6.3864e-05	1.4939e-04
	P-value(EM)	'||'	'||'	1.1067e-04	'||'	'||'	1.8267e-04
RSA	BEST	7.1089e-04	-3.0950e+03	**0**	**8.8818e-16**	**0**	1.6690
	WORST	0.0103	-1.9350e+03	**0**	**8.8818e-16**	**0**	1.6690
	MEAN	0.0047	-2.3866e+03	**0**	**8.8818e-16**	**0**	1.6690
	SD	0.0031	401.2004	**0**	**0**	**0**	2.3406e-16
	P-value(E)	1.8267e-04	1.8267e-04	0.0022	NaN	NaN	4.9177e-05
	P-value(EM)	'||'	'||'	NaN	NaN	NaN	6.3864e-05
SMA	BEST	4.2446e-04	-1.2555e+04	0.0027	0.0020	0.0036	0.8314
	WORST	0.0740	-3.7818e+03	26.9253	0.0524	0.0880	1.6622
	MEAN	0.0293	-7.2608e+03	4.1987	0.0282	0.0487	1.2546
	SD	0.0217	3.8479e+03	8.7322	0.0167	0.0305	0.2876
	P-value(E)	1.8267e-04	0.3847	0.3840	6.3864e-05	6.3864e-05	1.4939e-04
	P-value(EM)	'||'	1.8267e-04	1.1067e-04	'||'	'||'	1.8267e-04
SWO	BEST	66.0314	-3.3444e+03	375.6053	20.1243	423.7847	2.1526e+08
	WORST	102.7668	-2.6099e+03	429.9510	20.5447	550.9944	4.6015e+08
	MEAN	90.2103	-2.9264e+03	407.7917	20.3921	507.9397	3.4502e+08
	SD	11.9828	225.8055	17.5389	0.1530	47.6517	6.2470e+07
	P-value(E)	1.8267e-04	1.8267e-04	1.7861e-04	6.3864e-05	6.3864e-05	1.4939e-04
	P-value(EM)	'||'	'||'	1.1067e-04	'||'	'||'	1.8267e-04
WOA	BEST	4.0683	-8.4432e+03	241.8468	12.7928	66.1645	1.1880e+06
	WORST	18.1893	-4.7413e+03	340.4939	17.8355	215.8781	6.0313e+07
	MEAN	11.1807	-6.3963e+03	306.6318	15.5953	155.9841	2.5707e+07
	SD	4.4140	1.2397e+03	34.3787	1.6976	51.5082	2.2675e+07

	P-value(E)	1.8267e-04	5.8284e-04	1.7861e-04	6.3864e-05	6.3864e-05	1.4939e-04
	P-value(EM)	'||'	1.8267e-04	1.1067e-04	'||'	'||'	1.8267e-04

Algo/Functions		F13	F14	F15	F16	F17	F18
ESGO	BEST	1.3498e-32	**0.9980**	**3.0749e-04**	**-1.0316**	**0.3979**	3.0000
	WORST	0.0548	**0.9980**	**3.0749e-04**	**-1.0316**	**0.3979**	3.0000
	MEAN	0.0077	**0.9980**	**3.0749e-04**	**-1.0316**	**0.3979**	3.0000
	SD	0.0172	**0**	**9.7310e-20**	**0**	**0**	9.3622e-16
EMSGO	BEST	**1.4482e-09**	**0.9980**	**3.0749e-04**	**-1.0316**	**0.3979**	**3.0000**
	WORST	**9.3515e-05**	**0.9980**	**3.0749e-04**	**-1.0316**	**0.3979**	**3.0000**
	MEAN	**1.8852e-05**	**0.9980**	**3.0749e-04**	**-1.0316**	**0.3979**	**3.0000**
	SD	**3.3044e-05**	**0**	**9.7310e-20**	**0**	**0**	**0**
AVOA	BEST	3.4455	3.3015	0.0018	-1.0316	0.3982	3.0029
	WORST	2.2399e+08	25.6376	0.0510	-0.9123	0.4404	7.0925
	MEAN	7.5618e+07	12.5651	0.0205	-0.9929	0.4142	4.3022
	SD	8.7433e+07	7.4563	0.0159	0.0441	0.0166	1.7492
	P-value(E)	1.7761e-04	6.3864e-05	1.7462e-04	6.3864e-05	6.3864e-05	1.0997e-04
	P-value(EM)	1.8267e-04	'||'	1.8267e-04	'||'	'||'	6.3864e-05
DE	BEST	6.7444e+08	3.0014	0.0042	-1.0215	0.4034	4.6114
	WORST	1.0352e+09	15.5151	0.0256	-0.7381	0.6127	12.6626
	MEAN	8.9934e+08	9.6993	0.0181	-0.8525	0.4704	7.4617
	SD	1.1373e+08	4.4249	0.0080	0.0957	0.0748	2.9065
	P-value(E)	1.7761e-04	6.3864e-05	1.7462e-04	6.3864e-05	6.3864e-05	1.0997e-04
	P-value(EM)	1.8267e-04	'||'	1.8267e-04	'||'	'||'	6.3864e-05
EDO	BEST	0.1964	0.9980	6.0609e-04	-1.0316	0.3979	3.0000
	WORST	0.5513	0.9982	0.0013	-1.0316	0.3979	3.0007
	MEAN	0.3684	0.9981	8.3949e-04	-1.0316	0.3979	3.0002
	SD	0.1092	8.7514e-05	1.9852e-04	1.3264e-06	1.4829e-05	2.1767e-04
	P-value(E)	1.7761e-04	6.3864e-05	1.7462e-04	6.3864e-05	0.4429	1.0997e-04
	P-value(EM)	1.8267e-04	'||'	0.0028	'||'	'||'	6.3864e-05
GWO	BEST	1.8154e+07	2.0230	0.0038	-1.0308	0.4006	3.0044
	WORST	1.1507e+08	12.9125	0.0413	-0.9536	0.6533	5.5323
	MEAN	5.8232e+07	7.1274	0.0164	-1.0038	0.4841	3.8311
	SD	2.9227e+07	3.8136	0.0125	0.0263	0.0904	0.6841
	P-value(E)	1.7761e-04	6.3864e-05	1.7462e-04	6.3864e-05	6.3864e-05	1.0997e-04
	P-value(EM)	1.8267e-04	'||'	1.8267e-04	'||'	'||'	6.3864e-05
KOA	BEST	7.2168e+08	2.9821	0.0049	-1.0276	0.4016	7.6497
	WORST	1.0797e+09	27.8353	0.1063	-0.6278	0.5907	32.3446
	MEAN	9.2260e+08	12.2474	0.0525	-0.8687	0.4792	16.5206
	SD	1.1321e+08	8.1667	0.0338	0.1568	0.0626	8.5684
	P-value(E)	1.7761e-04	6.3864e-05	1.7462e-04	6.3864e-05	6.3864e-05	1.0997e-04
	P-value(EM)	1.8267e-04	'||'	1.8267e-04	'||'	'||'	6.3864e-05

LSO	BEST	3	4.0339	0.0102	-1.0139	0.4016	5.8453
	WORST	3	12.6705	0.1022	-0.3175	0.5907	37.6539
	MEAN	3	9.8962	0.0532	-0.6705	0.4792	16.8701
	SD	0	3.7132	0.0282	0.2216	0.0626	11.8080
	P-value(E)	6.1582e-05	4.9177e-05	1.7462e-04	6.3864e-05	6.3864e-05	1.0997e-04
	P-value(EM)	6.3864e-05	'‖'	1.8267e-04	'‖'	'‖'	6.3864e-05
MSA	BEST	2.8700e+08	0.9980	0.0030	-1.0298	0.4016	3.0250
	WORST	2.8700e+08	0.9980	0.0030	-1.0298	0.5907	3.0250
	MEAN	3.8465e+08	1.2827	0.0049	-1.0233	0.4792	3.1533
	SD	6.9371e+07	0.3251	0.0015	0.0078	0.0626	0.0997
	P-value(E)	1.7761e-04	6.3864e-05	1.7462e-04	6.3864e-05	6.3864e-05	1.0997e-04
	P-value(EM)	1.8267e-04	'‖'	1.8267e-04	'‖'	'‖'	6.3864e-05
NOA	BEST	7.6419e+08	7.9004	0.0138	-0.9764	0.4016	3.8617
	WORST	9.5384e+08	64.6598	0.1132	-0.1700	0.5907	31.7582
	MEAN	8.4808e+08	28.3656	0.0656	-0.5786	0.4792	11.9198
	SD	6.8994e+07	21.7623	0.0336	0.3114	0.0626	9.1953
	P-value(E)	1.7761e-04	6.3864e-05	1.7462e-04	6.3864e-05	6.3864e-05	1.0997e-04
	P-value(EM)	1.8267e-04	'‖'	1.8267e-04	'‖'	'‖'	6.3864e-05
RSA	BEST	3	9.2110	0.0090	-0.4096	0.5116	6.6847
	WORST	3	12.6705	0.1484	0	0.5907	123.1639
	MEAN	3	12.3246	0.0922	-0.0815	0.4522	40.6273
	SD	0	1.0940	0.0590	0.1429	0.0626	31.6910
	P-value(E)	6.1582e-05	2.4282e-05	1.6494e-04	4.1717e-05	6.3864e-05	1.0997e-04
	P-value(EM)	6.3864e-05	'‖'	1.7265e-04	'‖'	'‖'	6.3864e-05
SMA	BEST	2.9824	0.9980	7.1102e-04	-1.0315	0.4016	3.0008
	WORST	3.2405	9.8039	0.0143	-1.0209	0.5907	3.3609
	MEAN	3.1123	4.8996	0.0057	-1.0291	0.4792	3.0826
	SD	0.0930	3.2198	0.0042	0.0038	0.0626	0.1229
	P-value(E)	1.7761e-04	6.3864e-05	1.7462e-04	6.3864e-05	6.3864e-05	1.0997e-04
	P-value(EM)	1.8267e-04	'‖'	3.2984e-04	'‖'	'‖'	6.3864e-05
SWO	BEST	5.4627e+08	1.0171	0.0193	-1.0283	0.4116	3.3717
	WORST	1.0179e+09	22.0117	0.0942	-0.0493	0.5957	23.6171
	MEAN	7.5225e+08	8.7381	0.0611	-0.7510	0.4892	10.2534
	SD	1.3204e+08	5.9854	0.0224	0.3141	0.0726	6.9672
	P-value(E)	1.7761e-04	6.3864e-05	1.7462e-04	6.3864e-05	6.3864e-05	1.0997e-04
	P-value(EM)	1.8267e-04	'‖'	1.8267e-04	'‖'	'‖'	6.3864e-05
WOA	BEST	1.0576e+08	1.1509	0.0018	-1.0308	0.4016	3.0018
	WORST	3.9234e+08	20.1571	0.0514	-0.5923	0.5907	33.1136
	MEAN	2.2762e+08	11.2657	0.0180	-0.9283	0.4792	10.4506
	SD	9.3427e+07	6.4055	0.0175	0.1413	0.0626	11.9840
	P-value(E)	1.7761e-04	6.3864e-05	1.7462e-04	6.3864e-05	6.3864e-05	1.0997e-04
	P-value(EM)	1.8267e-04	'‖'	1.8267e-04	'‖'	'‖'	6.3864e-05

		F19	F20	F21	F22	F23
ESGO	BEST	-3.8628	-3.3220	**-10.1532**	-10.4029	**-10.5364**
	WORST	-3.8628	-3.2031	**-10.1532**	-10.4029	**-10.5364**
	MEAN	-3.8628	-3.3101	**-10.1532**	-10.4029	**-10.5364**

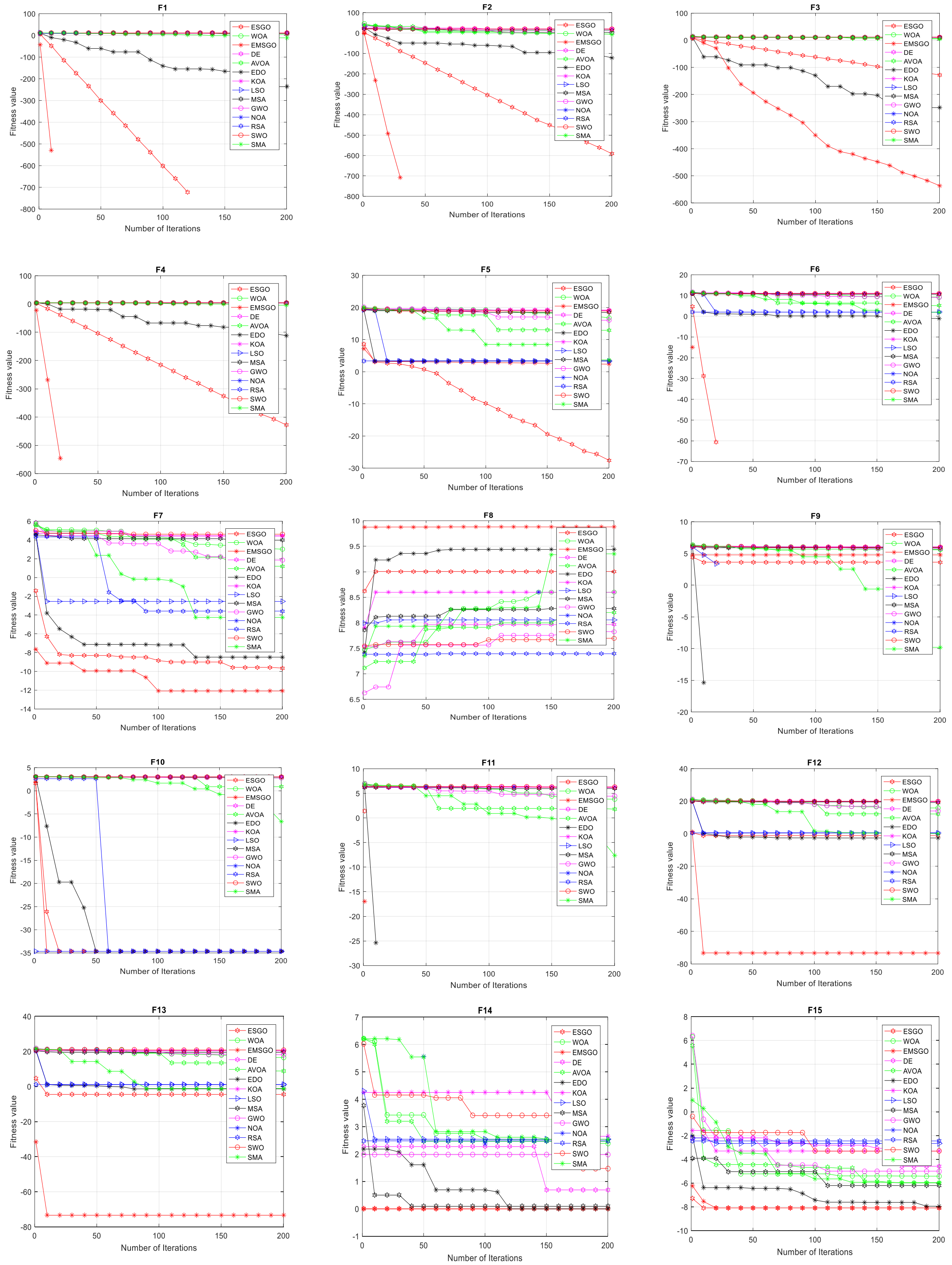
	SD	9.3622e-16	0.0376	**0**	1.8724e-15	**5.3864e-15**
EMSGO	BEST	-3.8628	-3.3220	**-10.1532**	**-10.4029**	**-10.5364**
	WORST	-3.8628	-3.3220	**-10.1532**	**-10.4029**	**-10.5364**
	MEAN	-3.8628	-3.3220	**-10.1532**	**-10.4029**	**-10.5364**
	SD	0	9.2038e-09	**0**	**0**	**5.3864e-15**
AVOA	BEST	-3.8554	-3.0677	-6.3022	-9.1029	-4.4976
	WORST	-3.6830	-2.3662	-1.2040	-1.5697	-1.4449
	MEAN	-3.8016	-2.7888	-2.5218	-3.6354	-2.4322
	SD	0.0532	0.2461	1.6158	2.2237	1.0535
	P-value(E)	6.3864e-05	1.3093e-04	6.3864e-05	6.3864e-05	6.3864e-05
	P-value(EM)	'‖'	1.8165e-04	'‖'	'‖'	'‖'
DE	BEST	-3.8529	-2.9290	-2.3594	-2.2949	-3.2228
	WORST	-3.6123	-1.9355	-0.7682	-0.7686	-1.0506
	MEAN	-3.7591	-2.4405	-1.2893	-1.1888	-1.5033
	SD	0.0866	0.3006	0.5691	0.4318	0.6396
	P-value(E)	6.3864e-05	1.3093e-04	6.3864e-05	6.3864e-05	1.6118e-04
	P-value(EM)	'‖'	1.8165e-04	'‖'	1.5932e-04	'‖'
EDO	BEST	-3.8627	-3.2807	-10.0372	-10.1394	-10.3280
	WORST	-3.8624	-3.1114	-9.0513	-6.6129	-5.8198
	MEAN	-3.8626	-3.1647	-9.4825	-8.8969	-8.5269
	SD	1.2376e-04	0.0475	0.3170	1.0859	1.5640
	P-value(E)	6.3864e-05	1.7865e-04	6.3864e-05	6.3864e-05	4.3745e-04
	P-value(EM)	6.3864e-05	4.3745e-04	6.3864e-05	6.3864e-05	4.3745e-04
GWO	BEST	-3.8598	-3.2761	-6.6657	-5.0855	-2.8311
	WORST	-3.7899	-2.7823	-0.9199	-1.8350	-1.3021
	MEAN	-3.8323	-3.0315	-1.9021	-3.3878	-2.1339
	SD	0.0275	0.1266	1.7055	1.2214	0.4926
	P-value(E)	6.3864e-05	1.7865e-04	6.3864e-05	6.3864e-05	1.6118e-04
	P-value(EM)	'‖'	4.3745e-04	'‖'	0.0297	1.6118e-04
KOA	BEST	-3.7779	-2.8099	-1.0715	-1.9384	-1.5148
	WORST	-3.5197	-2.0402	-0.4657	-0.6809	-0.8724
	MEAN	-3.6068	-2.4278	-0.7505	-1.0839	-1.1098
	SD	0.0958	0.2350	0.2215	0.3508	0.1946
	P-value(E)	6.3864e-05	1.3093e-04	6.3864e-05	6.3864e-05	1.6118e-04
	P-value(EM)	'‖'	1.8165e-04	6.3864e-05	1.5932e-04	1.6118e-04
LSO	BEST	-3.7985	-2.7576	-1.7209	-1.3938	-3.1473
	WORST	-3.5779	-1.8099	-0.6040	-0.6387	-0.8197
	MEAN	-3.7193	-2.3107	-1.0016	-0.9915	-1.4593
	SD	0.0750	0.3213	0.4116	0.2900	0.7417
	P-value(E)	6.3864e-05	1.3093e-04	6.3864e-05	6.3864e-05	1.6118e-04
	P-value(EM)	'‖'	1.8165e-04	'‖'	1.5932e-04	'‖'
MSA	BEST	-3.8605	-3.2122	-5.3730	-4.7984	-5.0588
	WORST	-3.8605	-3.2122	-5.3730	-4.7984	-5.0588
	MEAN	-3.8554	-3.0574	-3.1027	-3.3813	-5.0588
	SD	0.0055	0.0765	0.9302	0.9535	1.3777e-14
	P-value(E)	6.3864e-05	1.7865e-04	6.3864e-05	6.3864e-05	2.9377e-04
	P-value(EM)	'‖'	4.3745e-04	'‖'	'‖'	'‖'
NOA	BEST	-3.8433	-2.5874	-1.1580	-1.5230	-2.4136
	WORST	-3.5174	-1.8432	-0.5307	-0.7422	-0.8937
	MEAN	-3.7123	-2.1747	-0.7900	-1.1315	-1.2139
	SD	0.1111	0.3111	0.2374	0.2992	0.4667
	P-value(E)	6.3864e-05	1.3093e-04	6.3864e-05	1.5932e-04	1.6118e-04
	P-value(EM)	'‖'	1.8165e-04	'‖'	6.3864e-05	'‖'
RSA	BEST	-3.8094	-2.4156	-0.8657	-1.2773	-1.1469
	WORST	-3.0148	-0.9463	-0.2834	-0.4220	-0.6601
	MEAN	-3.4383	-1.6450	-0.4905	-0.6030	-0.9586
	SD	0.2788	0.5156	0.2049	0.2477	0.1349
	P-value(E)	6.3864e-05	1.3093e-04	6.3864e-05	1.5932e-04	1.6118e-04
	P-value(EM)	'‖'	1.8165e-04	'‖'	6.3864e-05	'‖'
SMA	BEST	-3.8623	-3.1346	-10.0208	-9.3983	-8.2808
	WORST	-3.8341	-2.2623	-2.4945	-1.9579	-1.0896
	MEAN	-3.8514	-2.7689	-4.5584	-4.6781	-3.7356
	SD	0.0078	0.3421	2.1547	2.6561	2.1286
	P-value(E)	6.3864e-05	1.3093e-04	6.3864e-05	6.3864e-05	3.8932e-04
	P-value(EM)	'‖'	1.8165e-04	'‖'	'‖'	'‖'

SWO	BEST	-3.8379	-2.7585	-1.4098	-1.9429	-3.4058
	WORST	-3.6279	-1.8461	-0.6324	-0.7866	-0.9884
	MEAN	-3.7513	-2.2405	-0.9346	-1.1783	-1.5817
	SD	0.0579	0.2779	0.3051	0.3436	0.6943
	P-value(E)	6.3864e-05	1.3093e-04	6.3864e-05	1.5932e-04	1.6118e-04
	P-value(EM)	'‖'	1.8165e-04	'‖'	6.3864e-05	'‖'
WOA	BEST	-3.8622	-2.9763	-6.2016	-3.4370	-4.4309
	WORST	-3.6581	-2.1432	-1.4378	-1.0475	-1.6685
	MEAN	-3.7568	-2.7168	-3.2018	-2.2719	-2.7508
	SD	0.0746	0.2455	1.3968	0.6886	1.0772
	P-value(E)	6.3864e-05	1.3093e-04	6.3864e-05	3.8932e-04	2.9377e-04
	P-value(EM)	'‖'	1.8165e-04	'‖'	6.3864e-05	'‖'

Table 4: WRS test results on Table 3

	F1		F2		F3		F4		F5		F6		F7		F8		F9		F10		F11		F12	
	E	EM	E	EM	E	EM	E	EM	E	EM	E	EM	E	EM	E	EM	E	EM	E	EM	E	EM	E	EM
AVOA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
DE	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDO	-	-	-	-	+	+	-	-	-	-	-	-	-	-	-	-	+	N	N	N	N	N	-	-
GWO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
KOA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
LSO	N	N	+	+	+	+	+	+	-	-	-	-	-	-	-	-	+	N	N	N	N	N		
MSA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
NOA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RSA	N	N	+	+	+	+	+	+	-	-	-	-	-	-	-	-	+	N	N	N	N	N		
SMA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	S	-	-	-	-	-	-	-	-	-
SWO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
WOA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	F13		F14		F15		F16		F17		F18		F19		F20		F21		F22		F23			
	E	EM	E	EM	E	EM	E	EM	E	EM	E	EM	E	EM	E	EM	E	EM	E	EM	E	EM		
AVOA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
DE	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
EDO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
GWO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
KOA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
LSO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
MSA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
NOA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
RSA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
SMA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
SWO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
WOA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		

Total no of'+'=10, Total no of 'S'=1, Total no of 'N'=8, Total no of'-'=257 (For ESGO algorithm)
Total no of'+'=7, Total no of 'S'=0, Total no of 'N'=11, Total no of'-'=258 (For EMSGO algorithm)
Here, E represents ESGO and EM represents EMSGO algorithm. "-", "+", and "S" denote that the performance of other approaches is worse, better, and similar to ESGO and EMSGO respectively.


F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
F11
F12
F13
F14
F15
Fitness value
Number of Iterations
ESGO
WOA
EMSGO
DE
AVOA
EDO
KOA
LSO
MSA
GWO
NOA
RSA
SWO
SMA

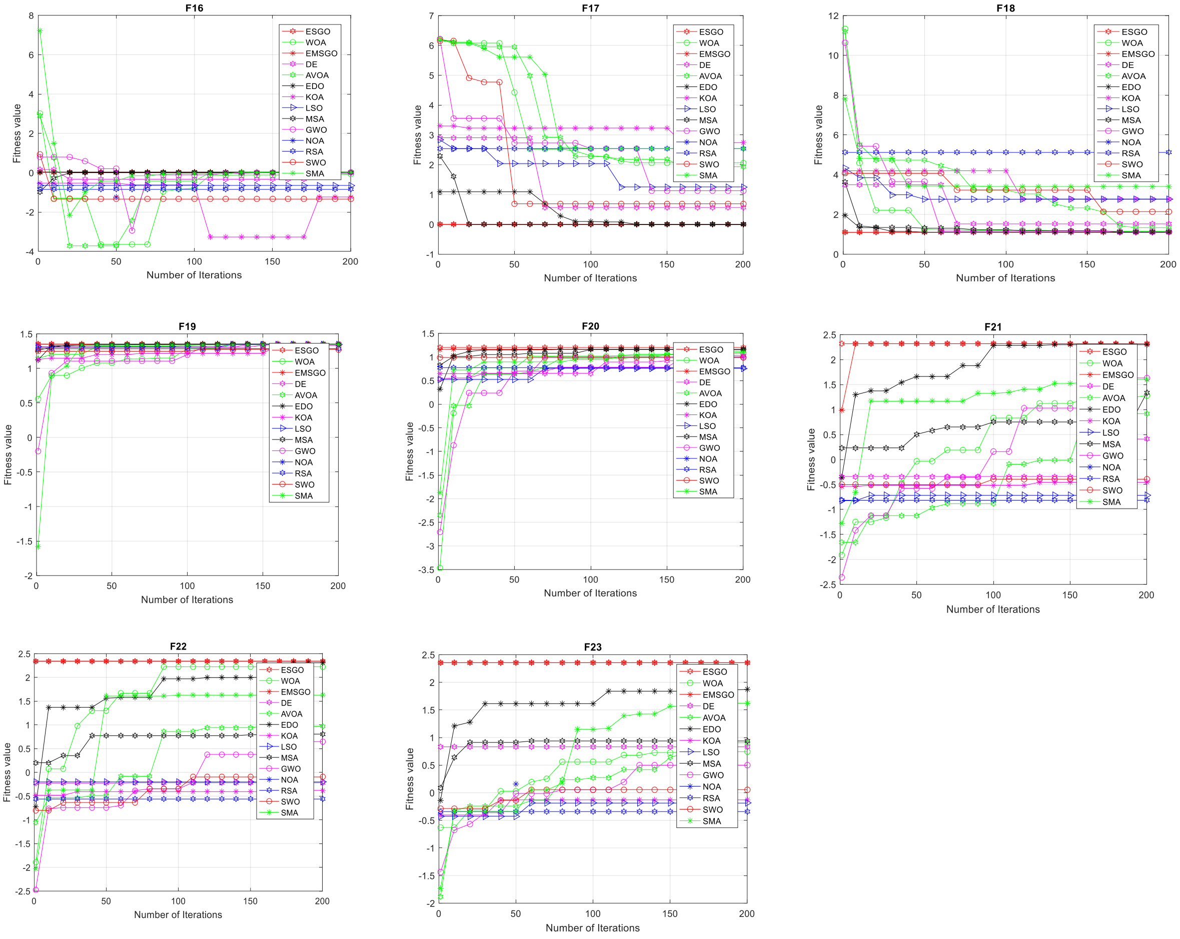


Figure 3: Convergence characteristics of algorithms

Discussion

In this experimental evaluation, the effectiveness of ESGO and EMSGO in navigating, exploiting, and avoiding local minima across a diverse range of benchmark functions, including both unimodal and multimodal functions, was examined.

Unimodal functions, characterized by a singular global optimum, serve as a measure of an algorithm's exploitation capability. The results presented in Table 3 for unimodal test functions (F1-F7) demonstrate the superior performance of ESGO and EMSGO, surpassing most other algorithms in all evaluated functions. These findings underscore the proficiency of ESGO and EMSGO in exploitation, showcasing their capacity to efficiently converge towards and exploit the optimal solution. This efficacy is attributed to the incorporation of the self-awareness probability (SAP) parameter.

Multimodal test functions, featuring multiple local optima that escalate with dimensionality, provide a platform for assessing an algorithm's exploration ability. Functions F8 through F23 represent multimodal scenarios. As indicated in Table 3, ESGO and EMSGO exhibit remarkable exploration capabilities, surpassing other methods. Across multimodal functions, ESGO and EMSGO not only achieve optimal solutions but also outperform all compared algorithms, demonstrating competitiveness with high-performance optimizers. The exploration prowess of ESGO and EMSGO can be attributed to the distinctive phases of optimization and the self-introspection parameter C.

Based on the WRS test results in Table 3,
For the ESGO algorithm:

- It performs worse than EDO for F3 and F9, LSO for F2, F3, F4, and F9, and RSA for F2, F3, F4, and F9.
- It matches the performance of EDO for F10 and F11, LSO for F1, F10, and F11, as well as RSA for F1, F10, and F11.
- ESGO outperforms other algorithms in all other functions.
- It consistently surpasses AVOA, DE, KOA, MSA, NOA, SMA, and SWO across all twenty-three functions.

For the EMSGO algorithm:

- It is inferior to EDO for F3, LSO for F2, F3, F4, and RSA for F2, F3, and F4.
- It matches the performance of EDO for F9, F10, and F11, LSO for F1, F9, F10, F11, and RSA for F1, F9, F10, and F11.

- EMSGO outshines other algorithms in the remaining functions.

It consistently outperforms AVOA, DE, KOA, MSA, NOA, SMA, and SWO for all twenty-three functions.

As depicted in Table 4, the ESGO algorithm surpasses AVOA, DE, EDO, GWO, KOA, LSO, MSA, NOA, RSA, SMA, SWO, and WOA in 23 cases out of 23, 23, 19, 23, 23, 16, 23, 23, 16, 22, 23, and 23 cases, respectively. Conversely, the ESGO algorithm performs less effectively than AVOA, DE, EDO, GWO, KOA, LSO, MSA, NOA, RSA, SMA, SWO, and WOA in zero, zero, two, zero, zero, four, zero, zero, four, one, zero, and zero cases, respectively. Additionally, the ESGO algorithm exhibits equivalence with EDO in two cases and with LSO and RSA in three and three cases, respectively. In summary, out of 276 instances, ESGO achieves equivalent results in 8 cases, the same solution in one case, a worse solution in 10 cases, and superior outcomes in 257 cases compared to other algorithms.

Similarly, as illustrated by Table 4, the EMSGO algorithm outperforms AVOA, DE, EDO, GWO, KOA, LSO, MSA, NOA, RSA, SMA, SWO, and WOA in 23 cases out of 23, 23, 19, 23, 23, 16, 23, 23, 16, 22, 23, and 23 cases, respectively. Conversely, the EMSGO algorithm performs less effectively than AVOA, DE, EDO, GWO, KOA, LSO, MSA, NOA, RSA, SMA, SWO, and WOA in zero, zero, one, zero, zero, three, zero, zero, three, one, zero, and zero cases, respectively. Additionally, the EMSGO algorithm exhibits equivalence with EDO in three cases and with LSO and RSA in four and four cases, respectively. In summary, out of 276 instances, EMSGO achieves equivalent results in 11 cases, the same solution in zero cases, a worse solution in 7 cases, and superior outcomes in 258 cases compared to other algorithms.

In conclusion, both ESGO and EMSGO demonstrate outstanding performance when addressing unimodal and multimodal functions.

Experiment 3: The performance comparison with well-known improved and hybrid metaheuristics algorithms

In this section, the same set of 23 optimization functions used in Experiments 1 and 2 is employed. The objective is to perform a comparative analysis between the simulation results obtained from ESGO, EMSGO, and the findings previously reported in reference [66] for two prominent optimization algorithms: the dynamic Harris Hawks Optimization with a mutation mechanism (DHHO/M) [67], and the Harris Hawks Optimization incorporating genetic operators such as crossover and mutation (HHOCM) [68]. Additionally, the analysis is extended to compare the simulation results from reference [69] for three other prominent optimization techniques: the Exponential Crow Search Algorithm (ECSA), the Power Crow Search Algorithm (PCSA), and the S-shaped Crow Search Algorithm (SCSA). Furthermore, the simulation results of a prominent hybrid algorithm, the Improved Hybrid Aquila Optimizer (IHAO) [70], and the Harris Hawks Optimization (HHO) [71] algorithm IHAOHHO [71] are also investigated.

Table 5: Simulation results for ESGO, EMSGO, DHHO/M, HHOCM, IHAOHHO, ECSA, PCSA, and SCSA

functions		ESGO	EMSGO	DHHO/M	HHOCM	IHAOHHO	ECSA	PCSA	SCSA
F1	BEST	**0**	**0**						
	MEAN	**0**	**0**	1.97e-95	**0**	3.37e-253	7.62e-28	1.41e-32	8.36e-35
	STD	**0**	**0**	6.74e-95	**0**	0	1.02e-27	1.87e-32	1.33e-34
F2	BEST	**0**	**0**						
	MEAN	**0**	**0**	1.326e-48	1.22e-203	1.56e-127	2.13e-11	1.02e-11	3.14e-12
	STD	**0**	**0**	6.07e-48	0	8.53e-127	2.13e-11	1.78e-11	7.20e-12
F3	BEST	2.20e-122	**0**						
	MEAN	1.46e-116	**0**	7.67e-70	**0**	2.74e-199	1.35e-22	2.27e-24	1.0e-24
	STD	2.37e-117	**0**	4.20e-69	**0**	0	5.36e-22	5.43e-24	4.37e-24
F4	BEST	**0**	**0**						
	MEAN	**0**	**0**	3.96e-43	4.55e-197	2.22e-129	2.87e-13	7.64e-13	4.20e-13
	STD	**0**	**0**	2.16e-42	0	1.11e-128	4.35e-13	1.28e-12	6.27e-12
F5	BEST	6.34e-21	**0**						
	MEAN	7.11e-20	**0**	6.70e-03	3.14e-02	5.39e-04	7.97e-01	1.25	1.32
	STD	8.47e-20	**0**	9.58e-03	5.02e-02	2.27e-03	1.62	1.83	1.91
F6	BEST	**0**	**0**						
	MEAN	**0**	**0**	7.39e-05	3.13e-04	3.59e-06	5.81e-28	7.08e-33	**0**
	STD	**0**	**0**	1.09e-04	3.83e-04	7.73e-06	1.08e-27	9.91e-33	**0**
F7	BEST	1.56e-05	**4.25e-07**						
	MEAN	2.53e-05	**5.29e-06**	1.58e-04	1.62e-04	9.53e-05	4.72e-04	4.50e-04	4.88e-05
	STD	1.18e-05	**3.23e-06**	1.44e-04	1.76e-04	7.67e-05	2.72e-04	2.98e-04	3.31e-05
F8	BEST	-1.05e+04	**-1.26e+04**						
	MEAN	-9.25e+03	**-1.26e+04**	-1.26e+ 04	-1.26e+04	-1.26e+04	-2.45e+03	-2.66e+03	-2.81e+03
	STD	7.86e+02	**0**	5.43e+02	5.50e+01	1.82e-01	3.53e+02	3.09e+02	3.76e+02
F9	BEST	0.00e+00	**0**						

	MEAN	12.5341	**0**	**0**	**0**	**0**	2.60	4.44	3.70
	STD	7.2981	**0**	**0**	**0**	**0**	7.06	6.00	7.08
F10	BEST	**8.88e-16**	**8.88e-16**						
	MEAN	**8.88e-16**	**8.88e-16**	**8.88e-16**	**8.88e-16**	**8.88e-16**	1.11e-01	1.09e-01	0.91e-01
	STD	**0**	0	0	0	0	2.20e-01	3.02e-01	1.36e-01
F11	BEST	**0**	**0**						
	MEAN	**0**	**0**	**0**	**0**	**0**	2.68e-02	1.23e-02	1.00e-02
	STD	**0**	**0**	**0**	**0**	**0**	2.86e-02	3.48e-02	3.19e-02
F12	BEST	1.59e-32	**1.57e-32**						
	MEAN	1.59e-32	**1.57e-32**	8.53e-06	1.57e-05	2.70e-07	2.23e-08	9.04e-08	8.11e-11
	STD	2.30e-34	**2.89e-48**	1.00e-05	2.27e-05	4.42e-07	1.47e-07	.76e-07	1.49e-10
F13	BEST	2.34e-32	**1.35e-32**						
	MEAN	2.34e-32	**1.35e-32**	9.52e-05	2.76e-04	3.02e-06	1.11e-04	1.05e-02	1.25e-02
	STD	2.21e-34	**2.10e-48**	1.10e-04	3.96e-04	5.08e-06	1.93e-04	2.12e-02	2.00e-02
F14	BEST	**9.98e-01**	**9.98e-01**						
	MEAN	**9.98e-01**	**9.98e-01**	1.29	1.16	1.59	9.98e-01	9.98e-01	9.98e-01
	STD	**0**	**0**	9.40e-01	5.27e-01	9.25e-01	3.43e-02	9.95e-02	5.64e-02
F15	BEST	3.07e-04	**3.07e-04**						
	MEAN	3.99e-04	**3.07e-04**	4.58e-04	5.61e-04	4.42e-04	3.27e-03	4.41e-03	1.23e-03
	STD	2.90e-04	**2.64e-16**	3.01e-04	4.29e-04	3.46e-04	5.61e-03	8.11e-03	3.64e-03
F16	BEST	**-1.0316**	-1.0316						
	MEAN	**-1.0316**	-1.0316	−1.0316	−1.0316	-1.0316	-1.0316	-1.0316	-1.0316
	STD	**0**	7.40e-17	5.75e-11	3.87e-09	3.19e-08	5.77e-16	6.19e-16	6.42e-16
F17	BEST								
	MEAN	**3.97e-01**	**3.97e-01**	3.98e-01	3.98e-01	3.98e-01	**3.97e-01**	**3.97e-01**	**3.97e-01**
	STD	**0**	**0**	3.99e-06	1.65e-06	4.04e-05	0	0	0
F18	BEST	3.0000	**3.0000**						
	MEAN	3.0000	**3.0000**	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
	STD	1.20e-15	**6.62e-16**	7.53e-08	1.55e-08	3.32e-06	2.16e-15	2.13e-15	2.04e-15
F19	BEST	-3.8628	**-3.8628**						
	MEAN	-3.86e+00	**-3.86**	-3.86	-3.86	-3.83	-3.86	3.86	3.86
	STD	9.3622e-16	**0**	3.09e-03	5.16e-04	7.1972e-02	2.61e-15	2.42e-15	2.37e-15
F20	BEST	-3.32e+00	**-3.32**						
	MEAN	-3.32e+00	**-3.32**	-3.11	-3.26	-3.08	-3.27	-3.27	-3.27
	STD	1.7813e-03	**2.23e-11**	8.39e-02	6.78e-02	1.19e-01	4.79e-02	5.29e-02	5.92e-02
F21	BEST	-1.02e+01	**-1.02e+01**						
	MEAN	-1.02e+01	**-1.02e+01**	-1.00	-5.06	-1.02	-2.05	-6.72	-6.57
	STD	1.32e-15	**0.00e+00**	1.26e-01	2.69e-04	1.45e-03	3.31	3.38	3.69
F22	BEST	-1.04e+01	**-1.04e+01**						
	MEAN	-1.04e+01	**-1.04e+01**	-1.02e+01	-5.09	-1.04e+01	-5.43	-3.96	-4.61
	STD	1.24e-13	**1.67e-15**	1.87e-01	1.06e-04	6.66e-04	3.52	3.80	3.53
F23	BEST	-1.05e+01	**-1.05e+01**						
	MEAN	-1.05e+01	**-1.05e+01**	-1.04e+01	-5.13	-1.05e+01	-6.00	-5.50	-4.55
	STD	1.32e-15	**2.78e-17**	1.54e-01	1.78e-04	6.67e-04	3.67	3.70	3.84

Table 6: Results of Friedman's Test on Table 5

Functions	ESGO	EMSGO	DHHO/M	HHOCM	IHAOHHO	ECSA	PCSA	SCSA	
F1	**2**	**2**	5	**2**	4	8	7	6	
F2	**1.5**	**1.5**	5	3	4	8	7	6	
F3	4	**1.5**	5	**1.5**	3	8	7	6	
F4	**1.5**	**1.5**	5	3	4	6	8	7	
F5	2	**1**	5	4	3	6	7	8	
F6	**2**	**2**	7	8	6	5	4	**2**	
F7	2	**1**	5	6	4	8	7	3	
F8	5	**2.5**	**2.5**	**2.5**	**2.5**	8	7	6	
F9	8	**1**	3	4	2	5	7	6	
F10	**3**	**3**	**3**	**3**	**3**	8	7	6	
F11	**3**	**3**	**3**	**3**	**3**	8	7	6	
F12	2	**1**	7	8	6	4	5	3	
F13	2	**1**	4	5	3	7	6	8	
F14	**1.5**	**1.5**	7	6	8	3	5	4	
F15	2	**1**	4	5	3	7	8	6	

F16	**1**	2	6	7	8	5	3	4	
F17	**3**	**3**	7	6	8	**3**	**3**	**3**	
F18	2	**1**	7	6	8	5	4	3	
F19	2	**1**	7	6	8	5	4	3	
F20	2	**1**	7	6	8	3	4	5	
F21	2	**1**	4	7	3	8	6	5	
F22	2	**1**	4	6	3	5	8	7	
F23	2	**1**	4	7	3	5	6	8	
Sum of ranks	57.5	35.5	116.5	115	107.5	138	137	121	
Average of ranks	2.5	1.543478	5.065217	5	4.673913	6	5.956522	5.26087	
Sum of ranks squared	191.75	66.25	642.25	657.5	611.25	900	873	709	

The assessment methodology and parameters employed for the ESGO and EMSGO algorithms closely adhere to the protocols outlined in Experiment 1. To ensure a fair comparison with the results from other algorithms, the dimension size was standardized to D = 30 for all functions, excluding fixed-dimensional benchmark functions. Additionally, the maximum number of fitness function evaluations was set to Max_FEs = 15,000, and 30 independent runs were conducted. Table 5 displays function values in terms of BEST, MEAN, and STD for the 23 classical benchmark functions, as discussed earlier. The BEST function value is provided exclusively for ESGO and EMSGO, as it is not reported for other algorithms in the imported papers. Bold font in the table denotes the most outstanding function value for each function. Evidently, EMSGO outperforms all listed algorithms in Table 5, exhibiting the lowest MEAN function value with the lowest STD for 22 out of the 23 functions. In addition, ESGO performs second-best in Table 5, obtaining the lowest STD for eight of the 23 functions and the lowest MEAN function value. Out of the 23 classical benchmark functions, HHOCM comes in third place with the lowest MEAN function value and the lowest STD for five of them

For each of the 23 classical benchmark functions listed in Table 5, Friedman's test [73], a nonparametric statistical test, was used to identify the lowest MEAN function value with the lowest STD. The findings of Friedman's test are shown in Table 6, revealing statistical insights into the ranks of ESGO, EMSGO, DHHO/M, HHOCM, IHAOHHO, ECSA, PCSA, and SCSA. The top-ranking algorithm is indicated in bold in this table. The algorithms are EMSGO, ESGO, IHAOHHO, HHOCM, DHHO/M, SCSA, PCSA, and ECSA, in that order of ranking. This suggests that out of all the algorithms that were looked at, EMSGO performs the best.

4.2 Process Synthesis and design problems of chemical and mechanical engineering

Real-world optimization problems are extremely difficult to solve due to the incredible complexity of objective functions and the profusion of nonlinear nonconvex equality and inequality constraints. This research focuses on a carefully chosen set of 26 limited problems taken from the mechanical and chemical engineering areas. In addition to the exhaustively documented 19 mechanical engineering problems in [64], the compilation includes seven process synthesis and design problems from chemical engineering. In these cases, the number of decision variables ranges from 2 to 30, the equality constraints from 0 to 4, and the inequality constraints from 1 to 86. See [64] for a detailed discussion of the problems and thorough formula definitions.

In the experimental phase, the parameter Max_FEs (maximum number of fitness function evaluations) is defined as:

$$\text{Max\_FEs}=\begin{cases} 1\times 10^5\,, & if\ D \le 10 \\ 2\times 10^5\,, & if\ 10 < D \le 30 \\ 4\times 10^5\,, & if\ 30 < \ D \le 50 \\ 8\times 10^5\,, & if\ 50 < D \le 150 \\ 10^6\,, & if\ 150 < D \end{cases}$$

With a fixed population size of 50, D here stands for the dimension (number of decision variables) for algorithms. Based on the learnings from Experiment 1, additional factors were chosen, and Table 7 provides an overview of the results. For constraint management, Deb's guidelines [74] are adopted, using a criterion that accepts impractical solutions if they show small violations, from 0.01 in the first iteration to 0.001 in the last. In problems pertaining to process synthesis, design, and optimization in mechanical engineering, this approach is very helpful because the global minimum frequently coincides with or is located close to the edge of the viable design space. Candidate solutions tend to gravitate towards these boundaries when this approach is used, increasing the likelihood of obtaining the global minimum [75]

Table 7: Details of 26 real-world constrained optimization problem,

	Name	F(x)	ESGO	EMSGO
P8	Process synthesis problem	2.0000000000e+00	**2.0000e+00(R)**	**2.0000e+00(R)**
P9	Process synthesis and design problem	2.5576545740e+00	**2.5577e+00(R)**	**2.5577e+00(R)**
P10	Process flow sheeting problem	1.0765430833e+00	**1.0765e+00(R)**	**1.0765e+00(R)**
P11	Two-reactor Problem	9.9238463653e+01	**9.9238e+01(R)**	**9.9238e+01(R)**
P12	Process synthesis problem	2.9248305537e+00	**2.9248e+00(R)**	**2.9248e+00(R)**
P13	Process design Problem	2.6887000000e+04	**2.6887e+04(R)**	**2.6887e+04(R)**
P14	Multi-product batch plant	5.3638942722e+04	**5.8477e+04**	5.9484e+04
Mechanical engineering problems				
P15	Weight Minimization of a Speed Reducer	2.9944244658e+03	**2.9944e+03(R)**	**2.9944e+03(R)**
P16	Optimal Design of Industrial refrigeration System	3.2213000814e-02	**3.2213e-02(R)**	**3.2213e-02(R)**
P17	Tension/compression spring design (case 1)	1.2665232788e-02	**1.2665e-02(R)**	1.2669e-02
P18	Pressure vessel design	5.8853327736e+03	**6.0597e+03**	6.3708e+03
P19	Welded beam design	1.6702177263e+00	**1.6702e+00(R)**	**1.6702e+00(R)**
P20	Three-bar truss design problem	2.6389584338e+02	**2.6390e+02(R)**	**2.6390e+02(R)**
P21	Multiple disk clutch brake design problem	2.3524245790e-01	**2.3524e-01(R)**	**2.3524e-01(R)**
P22	Planetary gear train design optimization problem	5.2576870748e-01	**5.3000e-01**	5.3319e-01
P23	Step-cone pulley problem	1.6069868725e+01	**1.6070e+01(R)**	1.6226e+01
P24	Robot gripper problem	2.5287918415e+00	**2.5288e+00(R)**	**2.5288e+00(R)**
P25	Hydro-static thrust bearing design problem	1.6254428092e+03	**1.6348e+03**	1.6475e+03
P26	Four-stage gear box problem	3.5359231973e+01	**3.5359e+01(R)**	**3.5359e+01(R)**
P27	10-bar truss design	5.2445076066e+02	**5.2453e+02**	5.2548e+02
P28	Rolling element bearing	1.4614135715e+04	**1.6958e+04**	**1.6958e+04**
P29	Gas Transmission Compressor Design	2.9648954173e+06	**2.9649e+06(R)**	**2.9649e+06(R)**
P30	Tension/compression spring design (case 2)	2.6138840583e+00	**2.6586e+00**	**2.6586e+00**
P31	Gear train design Problem	0.0000000000e+00	**0(R)**	1.4840e-26
P32	Himmelblau's Function	-3.0665538672e+04	**-3.0666e+04(R)**	**-3.0666e+04(R)**
P33	Topology Optimization	2.6393464970e+00	**2.6393e+00(R)**	**2.6393e+00(R)**

Discussion

The optimal solution is reached by the ESGO in 19 cases, whereas the ESGO is reached in 16 cases, according to Table 7. In two instances, the results are the same for both, while in five instances, ESGO receives a better result than EMSGO.

Overall discussion

From the entire experiment, it was discovered that ESGO performs better for real-world optimization problems, while EMSGO is more effective for handling classical optimization problems.

.

5 Conclusion

This paper introduces an innovative adaptation of the Social Group Optimization algorithm, namely Enhanced Social Group Optimization (ESGO) and Enhanced Modified Social Group Optimization (EMSGO). The Improving Phase of the SGO algorithm has been tailored to incorporate the concept of "hone." To evaluate the effectiveness of ESGO and EMSGO, extensive experiments were conducted across 23 benchmark functions, comparing their performance against twelve other optimization techniques and six recently introduced improved/hybrid algorithms. The test outcomes were rigorously assessed using Wilcoxon's rank test and Friedman's test, revealing that both ESGO and EMSGO significantly outperform the compared algorithms.

Furthermore, ESGO and EMSGO were applied to address 26 real-world optimization problems. ESGO successfully identified optimal solutions in 19 cases, while EMSGO achieved optimal solutions in 16 cases. The comprehensive experimentation indicates that ESGO excels in tackling real-world optimization problems, whereas EMSGO demonstrates superior performance in classical optimization challenges. Consequently, it is concluded that while these algorithms exhibit exceptional proficiency in classical optimization scenarios, their performance may vary when applied to real-world problems. Future research will explore their applicability in image processing, industry, neural networks, text analysis, and data mining as part of addressing real-world optimization challenges.

Author contributions

I am a single author of this manuscript.

Data availability

The datasets generated and/or analyzed during the current study and code are available from the corresponding author upon reasonable request.

Declarations

Conflict of Interest: The author declares that she has no conflict of interest in the publication of this paper.

References

[1] Faris, H., Ala'M, A. Z., Heidari, A. A., Aljarah, I., Mafarja, M., Hassonah, M. A., & Fujita, H. (2019). An intelligent system for spam detection and identification of the most relevant features based on evolutionary random weight networks. *Information Fusion*, *48*, 67-83. https://doi.org/10.1016/j.inffus.2018.08.002

[2] Zhou, Y., Liu, H., Cao, J., & Li, S. (2019). Composite learning fuzzy synchronization for incommensurate fractional-order chaotic systems with time-varying delays. *International Journal of Adaptive Control and Signal Processing*, *33*(12), 1739-1758. https://doi.org/10.1002/acs.2967

[3] Zhou, X., Lu, J., Huang, J., Zhong, M., & Wang, M. (2021). Enhancing artificial bee colony algorithm with multi-elite guidance. *Information Sciences*, *543*, 242-258 .https://doi.org/10.1016/j.ins.2020.07.037

[4] Wang, P., Zhou, Y., Luo, Q., Han, C., Niu, Y., & Lei, M. (2020). Complex-valued encoding metaheuristic optimization algorithm: A comprehensive survey. *Neurocomputing*, *407*, 313-342. https://doi.org/10.1016/j.neucom.2019.06.112

[5] Abd Elaziz, M., Yousri, D., Al-qaness, M. A., AbdelAty, A. M., Radwan, A. G., & Ewees, A. A. (2021). A Grunwald–Letnikov based Manta ray foraging optimizer for global optimization and image segmentation. *Engineering Applications of Artificial Intelligence*, *98*, 104105. https://doi.org/10.1016/j.engappai.2020.104105

[6] Alweshah, M. (2021). Solving feature selection problems by combining mutation and crossover operations with the monarch butterfly optimization algorithm. *Applied Intelligence*, *51*(6), 4058-4081. https://doi.org/10.1007/s10489-020-01981-0

[7] Xu, Z., Yang, H., Li, J., Zhang, X., Lu, B., & Gao, S. (2021). Comparative study on single and multiple chaotic maps incorporated grey wolf optimization algorithms. *IEEE Access*, *9*, 77416-77437. https://doi.org/10.1109/access.2021.3083220

[8] Morales-Castañeda, B., Zaldivar, D., Cuevas, E., Fausto, F., & Rodríguez, A. (2020) A better balance in metaheuristic algorithms: Does it exist? *Swarm and Evolutionary Computation*, *54*, 100671. https://doi.org/10.1016/j.swevo.2020.100671

[9] Satapathy, S. C., & Naik, A. (2014). Modified Teaching–Learning-Based Optimization algorithm for global numerical optimization—A comparative

study. *Swarm and Evolutionary Computation*, *16*, 28-
37. https://doi.org/10.1016/j.swevo.2013.12.005

[10] Aleti, A., & Moser, I. (2016). A systematic literature review of adaptive parameter control methods for evolutionary algorithms. *ACM Computing Surveys* 11 *(CSUR)*, *49*(3), 1-35. https://doi.org/10.1145/2996355

[11] Satapathy, S.C., Naik, A. & Parvathi, K. (2013). A teaching learning-based optimization based on orthogonal design for solving global optimization problems. *SpringerPlus* **2**, 130. https://doi.org/10.1186/2193-1801-2-130

[12] Zhong, X., & Cheng, P. (2021) An elite-guided hierarchical differential evolution algorithm. *Applied Intelligence*, *51*, 4962-4983. https://doi.org/10.1007/s10489-020-02091-7

[13] Wang, Y., Gao, S., Zhou, M., & Yu, Y. (2020) A multi-layered gravitational search algorithm for function optimization and real-world problems. *IEEE/CAA Journal of Automatica Sinica*, *8*(1), 94-109. doi: 10.1109/JAS.2020.1003462

[14] Gao, S., Wang, Y., Wang, J., & Cheng, J. (2017). Understanding differential evolution: A Poisson law derived from population interaction network. *Journal of Computational Science*, *21*, 140-149. https://doi.org/10.1016/j.jocs.2017.06.007

[15] Naik, A., Satapathy, S. C., & Abraham, A.(2020). Modified Social Group Optimization—A meta-heuristic algorithm to solve short-term hydrothermal scheduling. *Applied Soft Computing, 95*, 106524. https://doi.org/10.1016/j.asoc.2020.106524

[16] Yaghoubzadeh-Bavandpour, A., Bozorg-Haddad, O., Rajabi, M., Zolghadr-Asli, B., & Chu, X. (2020). Application of swarm intelligence and evolutionary computation algorithms for optimal reservoir operation. *Water Resources Management*, *36*(7), 2275-2292.
https://doi.org/10.1007/s11269-022-03141-0

[17] Xia, X., Gui, L., Zhang, Y., Xu, X., Yu, F., Wu, H., ... & Li, K. (2021). A fitness-based adaptive differential evolution algorithm. *Information Sciences*, *549*, 116-141. https://doi.org/10.1016/j.ins.2020.11.015

[18] Kumar, S., & Sikander, A. (2022). Optimum mobile robot path planning using improved artificial bee colony algorithm and evolutionary programming. *Arabian Journal for Science and Engineering*, *47*(3), 3519-3539. https://doi.org/10.1007/s13369-021-06326-8

[19] Rosso, M. M., Cucuzza, R., Aloisio, A., & Marano, G. C.(2022). Enhanced multi-strategy particle swarm optimization for constrained problems with an evolutionary-strategies-based unfeasible local search operator. *Applied Sciences*, *12*(5), 2285. https://doi.org/10.3390/app12052285

[20] Li, W., Meng, X., Huang, Y., & Fu, Z. H.(2022). Multipopulation cooperative particle swarm optimization with a mixed mutation strategy. *Information Sciences*, *529*, 179-196. https://doi.org/10.1016/j.ins.2020.02.034

[21] Luo, Q., Wang, H., Zheng, Y., & He, J. (2020). Research on path planning of mobile robot based on improved ant colony algorithm. *Neural Computing and Applications*, *32*, 1555-1566. https://doi.org/10.1007/s00521-019-04172-2

[22] Wang, H., Wang, W., Zhou, X., Zhao, J., Wang, Y., Xiao, S., & Xu, M. (2021). Artificial bee colony algorithm based on knowledge fusion. *Complex & Intelligent Systems*, *7*, 1139-1152. https://doi.org/10.1007/s40747-020-00171-2

[23] Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in engineering software*, *95*, 51-67. https://doi.org/10.1016/j.advengsoft.2016.01.008

[24] Arora, S., & Singh, S. (2019). Butterfly optimization algorithm: a novel approach for global optimization. *Soft Computing*, *23*, 715-734. https://doi.org/10.1007/s00500-018-3102-4

[25] Dhiman, G., & Kumar, V. (2019). Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems. Knowledge-Based Systems, 165, 169-196. https://doi.org/10.1016/j.knosys.2018.11.024

[26] Dhiman, G., & Kaur, A. (2019). STOA: a bio-inspired based optimization algorithm for industrial engineering problems. *Engineering Applications of Artificial Intelligence*, *82*, 148-174. https://doi.org/10.1016/j.engappai.2019.03.021

[27] Khishe, M., & Mosavi, M. R. (2020). Chimp optimization algorithm. *Expert systems with applications*, *149*, 113338. https://doi.org/10.1016/j.eswa.2020.113338

[28] Chou, J. S., & Truong, D. N. (2021). A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean. *Applied Mathematics and Computation*, *389*, 125535. https://doi.org/10.1016/j.amc.2020.125535

[29] Naik, A., Parvathi, K., Satapathy, S.C., Nayak, R., Panda, B.S.(2013). QoS Multicast Routing Using Teaching Learning Based Optimization. In: Kumar M., A., R., S., Kumar, T. (eds) Proceedings of International Conference on Advances in Computing. Advances in Intelligent Systems and Computing, vol 174. Springer, New Delhi. https://doi.org/10.1007/978-81-322-0740-5_6

[30] Satapathy, S., & Naik, A. (2016). Social group optimization (SGO): A new population evolutionary optimization technique. *Complex & Intelligent Systems,* *2*(3), 173–203. https://doi.org/10.1007/s40747-016-0022-8

[31] Naik, A., Satapathy, S.C. (2021). Past present future: a new human-based algorithm for stochastic optimization. *Soft Comput* 25, 12915–12976. https://doi.org/10.1007/s00500-021-06229-8

[32] Sadollah, A., Bahreininejad, A., Eskandar, H., & Hamdi, M. (2013). Mine blast algorithm: A new population-based algorithm for solving constrained engineering optimization problems. *Applied Soft Computing*, *13*(5), 2592-2612. https://doi.org/10.1016/j.asoc.2012.11.026

[33] Wang, K., Li, X., Gao, L., Li, P., & Gupta, S. M. (2021). A genetic simulated annealing algorithm for parallel partial disassembly line balancing problem. *Applied Soft Computing*, *107*, 107404. https://doi.org/10.1016/j.asoc.2021.107404

[34] Pelusi, D., Mascella, R., Tallini, L., Nayak, J., Naik, B., & Deng, Y. (2020). Improving exploration and exploitation via a hyperbolic gravitational search algorithm. *Knowledge-Based Systems*, *193*, 105404. https://doi.org/10.1016/j.knosys.2019.105404

[35] Eskandar, H., Sadollah, A., Bahreininejad, A., & Hamdi, M. (2012) Water cycle algorithm–A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers & Structures*, *110*, 151-166. https://doi.org/10.1016/j.compstruc.2012.07.010

[36] Kaveh, A., & Khayatazad, M. (2012) A new meta-heuristic method: ray optimization. *Computers & structures*, *112*, 283-294. https://doi.org/10.1016/j.compstruc.2012.09.003

[37] Alatas, B. (2011). ACROA: artificial chemical reaction optimization algorithm for global optimization. *Expert Systems with Applications*, *38*(10), 13170-13180. https://doi.org/10.1016/j.eswa.2011.04.126

[38] Naik, A. (2023). Multi-objective social group optimization for machining process. *Evol. Intel.* https://doi.org/10.1007/s12065-023-00856-w

[39] Naik, A. (2023). Chaotic Social Group Optimization for Structural Engineering Design Problems. *J Bionic Eng* **20**, 1852–1877. https://doi.org/10.1007/s42235-023-00340-2

[40] Singh, A. K., Kumar, A., Mahmud, M., Kaiser, M. S., & Kishore, A. (2021). COVID-19 infection detection from chest X-ray images using hybrid social group optimization and support vector classifier. *Cognitive Computation*, 1-13. https://doi.org/10.1007/s12559-021-09848-3

[41] Meesala, Y. N., Parida, A. K., & Naik, A. (2024). Optimized feature selection using modified social group
optimization. *Informatica*, *48*(11).https://doi.org/10.31449/inf.v48i11.6160

[42] Meesala, Y. N., Parida, A. K., & Naik, A. (2024). Simultaneous Clustering and Feature Selection using Social Group Optimization with Dynamic Threshold Setting for Microarray Data. *Informatica*, *48*(23). https://doi.org/10.31449/inf.v48i23.7019

[43] Huynh, V. H., Nguyen, T. H., Pham, H. C., Huynh, T. M. D., Nguyen, T. C., & Tran, D. H. (2021). Multiple objective social group optimization for time–cost–quality–carbon dioxide in generalized construction projects. *International Journal of Civil Engineering*, *19*, 805-822. https://doi.org/10.1007/s40999-020-00581-w

[44] Naik, A., Satapathy, S. C., Ashour, A. S., & Dey, N. (2018). Social group optimization for ghttps://doi.org/10.1016/j.cma.2023.116200lobal optimization of multimodal functions and data clustering problems. *Neural Computing and Applications*, *30*, 271-287. https://doi.org/10.1007/s00521-016-2686-9

[45] Naik, A & Chokkalingam, P. (2021) Binary social group optimization algorithm for solving 0-1 knapsack problem. *Decision Science Letters* , 11(1), 55-72. https://doi.org/10.5267/j.dsl.2021.8.004

[46] Naik, A. (2023). Marine predators social group optimization: a hybrid approach. *Evol. Intel.* https://doi.org/10.1007/s12065-023-00891-7

[47] Wolpert, D. H., & Macready, W. G. (1997) No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation, 1*(1), 67–82. doi: 10.1109/4235.585893

[48] Abdollahzadeh, B., Gharehchopogh, F. S., & Mirjalili, S. (2021). African vulture's optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. Computers & Industrial Engineering, 158, 107408. Elsevier BV. Retrieved from https://doi.org/10.1016%2Fj.cie.2021.107408

[49] Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, *11*, 341-359. https://doi.org/10.1023/A:1008202821328

[50] Abdel-Basset, M., El-Shahat, D., Jameel, M., & Abouhawwash, M. (2023). Exponential distribution optimizer (EDO): a novel math-inspired algorithm for global optimization and engineering

problems. *Artificial Intelligence Review*, 1-72. https://doi.org/10.1007/s10462-023-10403-9

[51] Mirjalili, S. M. S. M., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer Adv Eng Softw 69: 46–61.
https://doi.org/10.1016/j.advengsoft.2013.12.007

[52] Abdel-Basset, M., Mohamed, R., Azeem, S. A. A., Jameel, M., & Abouhawwash, M. (2023). Kepler optimization algorithm: A new metaheuristic algorithm inspired by Kepler's laws of planetary motion. *Knowledge-Based Systems*, *268*, 110454., https://doi.org/10.1016/j.knosys.2023.110454.

[53] Abdel-Basset, M., Mohamed, R., Sallam, K. M., & Chakrabortty, R. K. (2022). Light spectrum optimizer: a novel physics-inspired metaheuristic optimization algorithm. *Mathematics*, *10*(19), 3466. https://doi.org/10.3390/math10193466

[54] Abdel-Basset, M., Mohamed, R., Zidan, M., Jameel, M., & Abouhawwash, M. (2023). Mantis Search Algorithm: A novel bio-inspired algorithm for global optimization and engineering design problems. *Computer Methods in Applied Mechanics and Engineering*, *415*, 116200. https://doi.org/10.1016/j.cma.2023.116200

[55] Abdel-Basset, M., Mohamed, R., Jameel, M., & Abouhawwash, M. (2023). Nutcracker optimizer: A novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems. *Knowledge-Based Systems*, *262*, 110248. https://doi.org/10.1016/j.knosys.2022.110248

[56] Abualigah, L., Abd Elaziz, M., Sumari, P., Geem, Z. W., & Gandomi, A. H. (2022). Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer. *Expert Systems with Applications*, *191*, 116158.,
https://doi.org/10.1016/j.eswa.2021.116158.

[57] Li, S., Chen, H., Wang, M., Heidari, A. A., & Mirjalili, S. (2020). Slime mould algorithm: A new method for stochastic optimization. *Future Generation Computer Systems*, *111*, 300-323.s, doi: https://doi.org/10.1016/j.future.2020.03.055

[58] Abdel-Basset, M., Mohamed, R., Jameel, M., & Abouhawwash, M. (2023). Spider wasp optimizer: A novel meta-heuristic optimization algorithm. *Artificial Intelligence Review*, 1-64. https://doi.org/10.1007/s10462-023-10446-y

[59] Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks' optimization: algorithm and applications. Future Gener Comput Syst 97:849–872. https://doi.org/10.1016/ j. future.2019.02.028.

[60] Shareef, H., Ibrahim, A. A., & Mutlag, A. H. (2015). Lightning search algorithm. *Applied Soft Computing*, *36*, 315-333. https://doi.org/10.1016/j.asoc.2015.07.028

[61] Nematollahi, A. F., Rahiminejad, A., & Vahidi, B. (2017). A novel physical based meta-heuristic optimization method known as lightning attachment procedure optimization. *Applied Soft Computing*, *59*, 596-621. https://doi.org/10.1016/j.asoc.2017.06.033

[62] Nematollahi AF, Rahiminejad A, Vahidi B (2020) A novel meta-heuristic optimization method based on golden ratio in nature, Soft Comput. 24, pp.1117–1151, http://dx.doi.org/10.1007/s00500-019-03949-w.

[63] Moghdani, R., & Salimifard, K. (2018). Volleyball premier league algorithm. *Applied Soft Computing*, *64*, 161-185. https://doi.org/10.1016/j.asoc.2017.11.043

[64] Kumar, A., Wu, G., Ali, M. Z., Mallipeddi, R., Suganthan, P. N., & Das, S. (2020). A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. *Swarm and Evolutionary Computation*, 56, 100693. https://doi.org/10.1016/j.swevo.2020.100693

[65] Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, *1*(1), 3-18. https://doi.org/10.1016/j.swevo.2011.02.002

[66] Song, M., Jia, H., Abualigah, L., Liu, Q., Lin, Z., Wu, D., & Altalhi, M. (2022). Modified harris hawk's optimization algorithm with exploration factor and random walk strategy. *Computational Intelligence and Neuroscience*, *2022*. https://doi.org/10.1155/2022/4673665

[67] Jia, H., Lang, C., Oliva, D., Song, W., & Peng, X. (2019). Dynamic harris hawk's optimization with mutation mechanism for satellite image segmentation. *Remote sensing*, *11*(12), 1421. https://doi.org/10.3390/rs11121421

[68] Houssein, E. H., Neggaz, N., Hosney, M. E., Mohamed, W. M., & Hassaballah, M. (2021). Enhanced Harris hawk's optimization with genetic operators for selection chemical descriptors and compounds activities. *Neural Computing and Applications*, *33*, 13601-13618. https://doi.org/10.1007/s00521-021-05991-y

[69] Sheta, A., Braik, M., Al-Hiary, H., & Mirjalili, S. (2023). Improved versions of crow search algorithm for solving global numerical optimization problems. *Applied Intelligence*, 1-45. https://doi.org/10.1007/s10489-023-04732-z

[70] Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A. A., Al-Qaness, M. A., & Gandomi, A. H. (2021). Aquila optimizer: a novel meta-heuristic optimization algorithm. *Computers & Industrial Engineering*, *157*, 107250. https://doi.org/10.1016/j.cie.2021.107250

[71] Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future generation computer systems*, *97*, 849-872. https://doi.org/10.1016/j.future.2019.02.028

[72] Wang, S., Jia, H., Liu, Q., & Zheng, R. (2021). An improved hybrid Aquila Optimizer and Harris Hawks Optimization for global optimization. *Math. Biosci. Eng*, *18*(6), 7076-7109.doi: 10.3934/mbe.2021352

[73] Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The annals of mathematical statistics*, *11*(1), 86-92. https://doi.org/10.1214/aoms/1177731944

[74] Deb, K. (2000). An efficient constraint handling method for genetic algorithms. *Computer methods in applied mechanics and engineering*, *186*(2-4), 311-338. https://doi.org/10.1016/S0045-7825(99)00389-8

[75] Kaveh, A., & Talatahari, S. (2009). A particle swarm ant colony optimization for truss structures with discrete variables. *Journal of Constructional Steel Research*, *65*(8-9), 1558-1568. https://doi.org/10.1016/j.jcsr.2009.04.021

A Secure and Scalable Sidechain Model for Fog Computing in Healthcare Systems

Ramzi Haraty, Ali Amhaz
Department of Computer Science and Mathematics Lebanese American University Beirut, Lebanon
Email: rharaty@lau.edu, ali.amhaz02@lau.edu

Keywords: blockchain, sidechain, fog computing, cloud computing, scalability, healthcare

Received: December 24, 2023

With the enormous amount of data produced daily, cloud and fog computing presented efficient and effective models for real-time data exchange. Nevertheless, this technology came with a cost at the security level, where it became an easy target for malicious actions that could quickly spread throughout the model. Blockchain, a recent and promising technology, has shown to be a suitable solution for securing transactions in the fog environment because of the distributed ledger structure that makes it resistant to many types of attacks. Scalability, however, introduced the main drawback of a blockchain by making it inefficient in some real-world applications, especially in the medical field, which includes a lot of data exchange. This work will suggest a scalable and secure model for fog and cloud computing in healthcare systems that depend on sidechains and the clustering of the available fog nodes. The importance of the model is highlighted, and experimental results showed promising outcomes.

Povzetek: V raziskavi je predlagan varen in razširljiv model stranske verige za megleno računalništvo v zdravstvenih sistemih, ki izboljšuje varnost in učinkovitost obdelave podatkov.

1 Introduction

The technological revolution in the last century has led to significant development in the software and hardware of information systems. For example, in the old banking systems, committing a transaction needed the manual assistance of one or more employees. However, almost all banking services nowadays are automated and allow clients to do various transactions from their homes or shops using online applications and micro hardware (i.e., electronic ships).

A significant part of this enormous development was the proliferation of the Internet of Things (IoT) devices. Those devices, which support connecting to the internet network, granted a variety of data manipulation actions such as gathering, transmitting, and processing. In addition, they helped in dispensing many human-controlled actions that consume time and resources. Many fields started using IoT devices because of their low prices and the ability to perform critical tasks without human supervision. Healthcare institutions, including hospitals, started using such devices to keep track of the patient's health records (i.e., blood pressure, temperature) and add them to the primary system for later use. Moreover, modern agriculture adopted IoT sensors connected to the internet to monitor the soil state for a better harvest.

This massive development of information systems and the amount of data produced (especially by IoT devices) brought the need to invent and enhance those systems to satisfy growing demands, including storage, processing power, and availability. Cloud computing came then to solve these problems, offering various services to facilitate data manipulation. It allowed the accomplishment of many tasks using remote servers provided by several international companies (i.e., Google and Apple). For example, cloud storage offered by Google supplied users with terabytes of storage at a low cost. Moreover, it eliminated the risk of losing the physical data resulting from any emergency. In addition, cloud services facilitate the deployment of large programs that needs many computer resources and cannot be done from the user side.

Although cloud computing presented the solution to many problems, it raised others. Such a technology consists of a centralized structure that serves millions of users in the same place, thus, causing unwanted latency in some critical applications. For example, in automated car projects, response time and availability are crucial measures that can lead to life-threatening problems. Those cars need quick operations toward any action that could happen on the roads (i.e., a child crossing the street). Based on that, fog computing [1] came as a solution to give a better performance. It provided services similar to cloud computing but with better performance. The distributed and close-to-user structure helped a lot in increasing the response time with much fewer failures.

With their distributed architecture, Fog systems were more secure than cloud computing. Nevertheless, the communication between the different fog nodes represented the main vulnerability that a hacker could exploit. For instance, a malicious transaction targeting a specific fog node could quickly spread throughout the system, making it very hard to recover to its original state. Moreover, heterogeneous models' damage levels would be much higher [2]. To tackle this problem, researchers proposed two different approaches [3, 4]. The first approach depends on preventing malicious transactions before entering the system, while the other suggests detecting and recovering the damage.

In the case of detecting and recovering malicious actions, studies focused on building efficient and effective algorithms [2, 5] that could scan the system for unusual behavior and directly start the recovery process in the case of attacks. In [6], the researchers focused on machine learning to discover any intrusion and to recover it. On the other hand, the prevention systems mainly focused on blockchain to approve any transaction before entering the fog system. For example, in [7], the authors presented a blockchain model to protect the system and facilitate the data exchange between the clients and the doctors in the hospital.

Recently, blockchain technology has become a significant target for many applications because of its high security and the ability to control the flow of transactions to any system, especially fog networks. This combination (blockchain + fog) allowed to filter the transactions of IoT devices by forcing the proof of work and validation between different nodes [8]. Blockchain effectiveness and efficiency are measured using a set of metrics to study the scalability and compatibility with the given systems [9].

1.1 Overview

This section provides an overview of the key topics addressed in this work: IoT devices, cloud computing, fog computing, and blockchain technology.

IoT devices are connected to the internet to gather, transmit, or process data. Their affordability and ability to function in challenging environments have led to widespread adoption across various fields. In healthcare systems, for instance, IoT devices have become essential for monitoring patients' vital signs and issuing instant alerts during emergencies. These devices also streamline processes like medication delivery. A notable example is an IoT innovation designed to monitor blood glucose levels and administer insulin automatically [10-11].

Cloud computing offers remote access to resources such as storage, processing power, and networks, all delivered over the internet. This technology enhances service quality and ensures quick, reliable access to resources [12-13].

Fog computing, while similar to cloud computing, operates as a decentralized system positioned between cloud services and end users. It manages, stores, and processes data closer to the client, acting as a complementary component to traditional cloud services.

Blockchain technology, introduced by Satoshi Nakamoto in 2008, marked a turning point in data security and management. It relies on a decentralized structure, eliminating the need for a single control point. Blockchain is a distributed ledger where every user holds a copy of the chain, updated in real-time. Transactions are recorded in blocks, each containing a unique "hash" to prevent unauthorized alterations, and every block links to its predecessor to maintain the chain's integrity. When a user initiates a transaction, a new block is created, requiring approval from the majority or all users. This consensus mechanism ensures transparency and prevents tampering. Once added, a block becomes immutable, preserving the chain's integrity. To address scalability issues, sidechains were developed as an innovative solution. These smaller, independent chains interact with the main blockchain to exchange assets when needed. Sidechains reduce the time required to add new blocks and offer localized privacy. They can also have different structures and mechanisms from the main chain. The interaction between sidechains and the main chain relies on protocols such as the Two-Way Peg Protocol, which ensures secure and efficient data exchange. This protocol can be symmetric, used for chains with similar structures, or asymmetric, enabling cross-chain communication between differing systems like Bitcoin and Ethereum [14-15]. Figure 1 illustrates the architecture of blockchain, emphasizing its structure and security mechanisms, including the role of hashes and the genesis block.

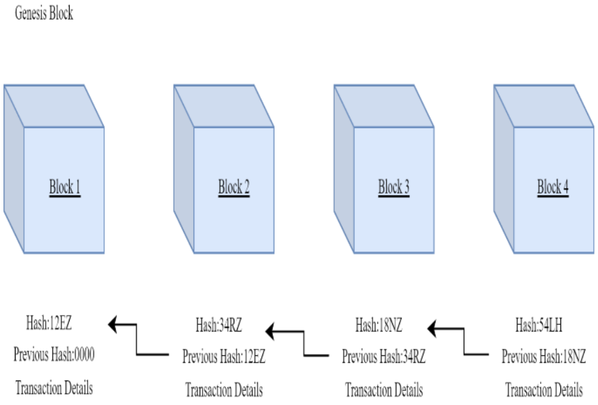


Figure 1: Blockchain architecture.

1.2 Problem statement

The fog architecture, while offering several advantages, is vulnerable to malicious transactions that can spread across the system, especially in decentralized environments like healthcare. This issue is challenging to resolve due to the interdependence of data across nodes, where a compromised node can affect others. Security solutions for fog systems generally fall into two categories: detection and recovery [2, 5, 8], which focus on identifying and mitigating malicious transactions, and prevention mechanisms aimed at blocking such transactions before they occur. Blockchain technology has been proposed as a solution to enhance security by validating transactions and providing a detailed, tamper-proof ledger [19-21]. However, as the number of fog nodes grows, scalability becomes a major concern, particularly in high-transaction environments like healthcare. This paper proposes a scalable blockchain approach for fog computing in healthcare, utilizing multiple sidechains to reduce transaction processing time and improve overall system performance.

1.3 Innovations in our work

Despite the superiority of fog computing over the cloud, it still needs a set of security restrictions to curb the vulnerabilities that take time to handle. Those vulnerabilities came from the decentralized architecture of fog that makes it easy to spread malicious transactions between the correlated nodes. Moreover, the separated structure imposes many challenges in tracking the attacks and recovering their effects.

Blockchain, a recent and promising technology, protects the environments that include committing transactions and exchanging data. Accordingly, this work will suggest a blockchain model that secures the fog nodes using a sidechain approach. The model will cluster the nodes based on their frequency of communication to form a sidechain and ensure local privacy between them. Within the same cluster, when a node needs to commit a transaction, it is approved first by most nodes, then a new block is created on the ledger. On the other hand, when there is communication between nodes from different clusters, the transactions are committed through a main chain formed from the unclustered nodes.

Our research offers several key innovations that address limitations in existing studies:

1. Integration of Blockchain for Secure Data Sharing: We utilize blockchain technology to protect data exchanged between fog nodes, ensuring secure, immutable transactions across the system.
2. Scalability Through Sidechains:
   To address the scalability issues commonly associated with blockchain, we propose the formation of sidechains within clusters of fog nodes. This approach reduces the validation time for each transaction by localizing processing within smaller, manageable groups.
3. Elimination of Centralized Communication:
   Unlike many existing models, our approach eliminates the need for centralized communication between fog nodes in different clusters, enhancing system efficiency and reducing bottlenecks.
4. Privacy Assurance Within Clusters:
   By confining sensitive data and operations within individual clusters, our model ensures robust privacy for intra-cluster communication.

1.4 Organization of the paper

In section 2, the paper will review the literature review related to the topic. Then, section 3 will suggest a new model that provides a secure and scalable blockchain solution for a healthcare system. Section 4 will show and analyze the obtained results after the simulations. Finally, the conclusion will be presented in section 5.

2 Related work

This section reviews studies on cloud and fog technologies in information systems, focusing on their integration and security measures, especially blockchain and sidechain models, for improved data exchange. Security solutions are categorized into detection and recovery methods and prevention methods. Detection and recovery approaches assess attack damage and implement recovery algorithms, while prevention relies on blockchain to ensure trustworthy transactions.

2.1 Cloud and fog computing in information systems

Cloud computing has been widely adopted for its computational and financial benefits.

- Smart Cities: A hierarchical cloud model was proposed in [22], with horizontal layers for interface management and vertical layers for security and data actions, improving data availability and user interaction.

- Healthcare: In [23], cloud services were examined for strengths (accessibility) and limitations (data management), with a focus on public and private cloud models.

- Agriculture: A cloud architecture in [24] addressed traditional system inefficiencies, enhancing flexibility and enabling better weather tracking for production.

Fog computing emerged as a solution to the limitations of cloud services, particularly for IoT and real-time applications/themes:

- Characteristics of Fog: As described in [25], fog nodes bridge IoT and cloud systems, leveraging decentralized architecture, numerous nodes to prevent single-point failures, and proximity to end devices for real-time communication.
- Real-Time Applications: The authors in [26] introduced a fog model with "Third Party Memory Management" for real-time IoT requests, reducing cloud dependency.
- Autonomous Cars: In [27], fog layers integrated with machine learning (Support Vector Machine) enhanced real-time response and trajectory planning, achieving promising results in simulations.
- Detection and Recovery: Techniques like IDS-based models ([2]), node differentiation ([5]), and trust-building mechanisms ([31]) enhance database integrity and reduce attack impact.
- Prevention: Blockchain models ([35], [19]) improve secure transactions, while sidechains ([8], [17]) address scalability and performance issues.

Table 1 presents a consolidated perspective that demonstrates the evolution of cloud and fog systems, their applications, and advanced security mechanisms.

Table 1: Related works – cloud and fog computing.

Reference	Focus Area	Key Contribution	Advantages
[22]	Cloud Computing in Smart Cities	Proposed a hierarchical model for integrating cloud computing into smart cities.	- Horizontal layer for interface establishment. - Vertical layer for security and data management.
[23]	Cloud Computing in Healthcare	Studied strengths, weaknesses, and models of cloud computing in healthcare applications.	- Highlighted public vs. private clouds. - Addressed accessibility and service management differences.
[24]	Cloud in Agriculture	Introduced a cloud architecture to modernize agricultural information systems and processes.	- Improved flexibility and weather tracking. - Enhanced production chain and data storage.
[25]	Fog Computing for IoT	Explored the essential role of fog computing in IoT device development and its characteristics.	- Reduced cloud pressure. - Enhanced decentralization and real-time communication.
[26]	Fog for Real-Time Applications	Presented a fog-based model for real-time IoT response with a "Third Party Memory Management" unit.	- Differentiated between normal and real-time requests. - Reduced unnecessary cloud data uploads.
[27]	Fog in Autonomous Cars	Integrated fog computing with Support Vector Machines to enhance autonomous car communication.	- Decentralized structure for better trajectory planning. - Improved response time and reduced latency.

2.2 Fog computing in healthcare systems

Fog computing has been increasingly adopted in healthcare to address the limitations of traditional cloud-based models, especially for real-time, low-latency applications.

- Real-Time Notifications: A fog model proposed in [28] enables real-time patient health monitoring with low latency by dividing operations into four layers: sensors, fog for data analysis, cloud for storage, and a management layer for oversight.
- Comparison with Cloud: In [29], a hybrid fog-cloud architecture demonstrated 28% faster response times and enhanced security via decentralization, which mitigated certain attack risks.
- Enhanced Security: The model in [30] incorporated VM selection for better IoT management and a cryptographic mechanism for public and private key generation, achieving improved latency and performance in iFogSim simulations.

2.3 Security in fog computing models

Security mechanisms in fog systems are categorized into detection and recovery and prevention, focusing heavily on blockchain integration:

Detection and recovery:

- Data Integrity: Algorithms in [2] and [5] detect malicious transactions, assess damage, and initiate recovery using logs and IDS tools, albeit with limitations in simulation and reliance on IDS accuracy.
- Node Isolation: Models like COMMITMENT ([31]) and DataIDS ([33]) reduce attack intensity by isolating malicious nodes and utilizing dependency graphs for anomaly detection.

Prevention with blockchain:

- FogChain for Healthcare: A blockchain-based architecture in [35] improved transaction throughput and response time by 66% compared to cloud systems.
- Scalability with Sidechains: Studies in [8] and [17] introduced sidechains to enhance blockchain scalability, enabling independent yet coordinated sub-chains to handle increased user demands efficiently.

For more details on the discussed models and their metrics, refer to Table 2, which summarizes their scope, methodologies, and performance outcomes.

Table 2: Related works - healthcare and fog security.

Reference	Focus Area	Key Contribution	Advantages	Limitations
[2]	Detection & Recovery in Fog Healthcare	Introduced an IDS-based model for assessing and recovering from database attacks in fog nodes.	- Efficient damage assessment. - Distinguished bad transactions for future use.	- No real-world simulation. - IDS reliance might lead to inaccuracies.

Reference	Focus Area	Key Contribution	Advantages	Limitations
[5]	Smart City Fog Data Recovery	Differentiated private/public fog nodes and developed damage assessment and recovery algorithms.	- Focused on managing public and utility data. - Recovery algorithms fix attacked databases.	- No simulation performed.
[8]	Blockchain Scalability via Sidechains	Integrated fog with sidechains to improve blockchain scalability and processing power.	- Better transaction rates. - Effective use of processing power.	None mentioned.
[17]	Sidechain Efficiency in Fog Computing	Presented a fog-sidechain-root architecture for scalability and transaction validation.	- Improved throughput, latency, and efficiency. - Supported access control mechanism.	None mentioned.
[19]	Blockchain and IoT Integration in Fog	Proposed a blockchain-based fog model for secure IoT data exchange with performance-testing algorithms.	- Secure environment for IoT-fog data exchange. - Good performance metrics.	- Scalability not considered. - Performance tested using local parameters.
[20]	Smart Cities with Blockchain & Fog	Developed a fog-blockchain-cloud model for security and performance in smart cities.	- Enhanced response time and energy efficiency. - Adopted encryption and authentication mechanisms.	- Did not address scalability issues.
[21]	Blockchain-Based Fog Security	Introduced blockchain and encrypted signatures to secure IoT data at fog nodes.	- Enhanced defense via blockchain transparency. - Good response time and scalability.	None mentioned.
[28]	Fog in Healthcare	Proposed a fog-based notification system for real-time health records with four layers.	- Low latency and fast response time. - Reduced data overhead on the cloud.	None mentioned.
[29]	Fog vs. Cloud in Healthcare	Compared fog and cloud models in healthcare based on performance and security.	- 28% faster response time than cloud. - Effective defense via decentralized structure.	None mentioned.
[30]	Fog Security Mechanisms	Added security with patient authentication, VM-based fog node selection, and cryptographic key management.	- Improved latency and system performance. - Simulated on iFogSim software.	None mentioned.
[31]	Malicious Node Mitigation (COMMITMENT)	Proposed a fog model to isolate malicious nodes and reduce attack intensity with trust records.	- Reduced attack intensity by 66%. - Decreased latency by 15 seconds.	None mentioned.
[32]	Malicious Node Detection	Studied behavior between fog servers	- Effective for fog-based vehicle	None mentioned.

Reference	Focus Area	Key Contribution	Advantages	Limitations
		to detect unusual activity and issued alerts based on thresholds.	networks. - Real-time threat detection.	
[33]	DataIDS Model for Database Attack Detection	Introduced dependency graphs for detecting abnormal fog node behavior.	- Effective against noise, replay, and stuck-at attacks. - Adequate experimental response.	- Lacked recovery mechanisms for attacks.
[34]	Machine Learning for Fog Security	Surveyed ML/AI integration for intrusion detection and data security in fog systems.	- Highlighted ML algorithms like Random Forest, Naive Bayes, and PCA.	None mentioned.
[35]	Blockchain in Fog for Healthcare	Suggested a FogChain model integrating blockchain for real-time health data exchange.	- 66% better response time than cloud. - Suitable for healthcare IoT data.	None mentioned.

3 The suggested model

3.1 Overview

The proposed model begins by clustering fog nodes based on their communication frequency, using the k-means clustering algorithm. A parameter k specifies the minimum number of nodes in each cluster, promoting efficient and localized data exchange within the clusters. Before clustering, the 10 least-interacting fog nodes are excluded and assigned to the main blockchain (central ledger). This decision aligns with recommendations suggesting a minimum of seven nodes to form a secure blockchain, as they ensure over 66% agreement to tolerate up to two untrusted participants. To ensure the security and integrity of transactions, the model employs the SHA-256 hashing algorithm for generating transaction hashes and maintaining a tamper-proof ledger. SHA-256, a standard in blockchain systems, provides robust cryptographic security by converting input data into fixed-length, irreversible hash values. For encryption, the model utilizes Elliptic Curve Cryptography (ECC), a highly secure and computationally efficient public-key encryption algorithm. ECC ensures secure communication and data exchange between nodes while offering smaller key sizes compared to RSA, making it well-suited for resource-constrained fog computing environments. The clustering process works as follows:

1. Clustering nodes:

- The k-means algorithm clusters fog nodes based on the frequency of communication links. Nodes that frequently exchange data are grouped into the same cluster, fostering local privacy and efficient transaction handling.
- The number of clusters (N) is predefined, ensuring balanced cluster sizes and optimal system performance.

2. Main blockchain (Central ledger):

- The 10 least-interacting nodes form the central ledger. These nodes maintain the global blockchain and facilitate inter-cluster communication.

3. Sidechains:

- Each cluster forms a sidechain, where the fog nodes act as users of the chain. Transactions within a sidechain are monitored and validated by the nodes in the cluster. This design provides enhanced privacy for entities requiring secrecy, as data within a sidechain remains isolated from the central ledger.
- The Plasma framework is used to create and manage sidechains, enabling efficient data processing and scalability. Transactions in sidechains are handled using Ethereum protocols, incorporating features like smart contracts, Decentralized Autonomous Organizations (DAO), and digital tokens for advanced functionality.

This approach offers significant advancements in security, scalability, and data integrity within the proposed system. By employing sidechains, the model ensures that sensitive data is confined and processed within specific clusters, thereby enhancing localized security and protecting information from unauthorized access or exposure. The localization of data minimizes the risk of breaches across the broader network, fostering a secure environment tailored to the needs of entities requiring heightened privacy. Additionally, the integration of sidechains addresses scalability challenges by alleviating the transaction load on the main blockchain. This design enables more efficient processing, as transactions within sidechains are handled independently of the central ledger. Consequently, this improves resource allocation and significantly reduces latency, ensuring that the system can accommodate increased demands without compromising performance.

Furthermore, the adoption of SHA-256 hashing and ECC provides robust mechanisms for consensus and security. The SHA-256 hashing algorithm ensures the integrity of transactions by generating unique, immutable hash values, effectively preventing unauthorized tampering or data corruption. Simultaneously, ECC encryption secures data exchanges between nodes, offering a high level of cryptographic protection with smaller key sizes, which is particularly advantageous in resource-constrained environments. Together, these cryptographic techniques fortify the system against potential vulnerabilities, ensuring that all transactions are authenticated and secure. Figure 2 illustrates the detailed process, showcasing the steps involved in clustering, the formation of sidechains, and their seamless integration with the main blockchain. This visual representation highlights the model's ability to deliver a secure, scalable, and efficient architecture, specifically designed to meet the stringent requirements of applications such as healthcare systems.

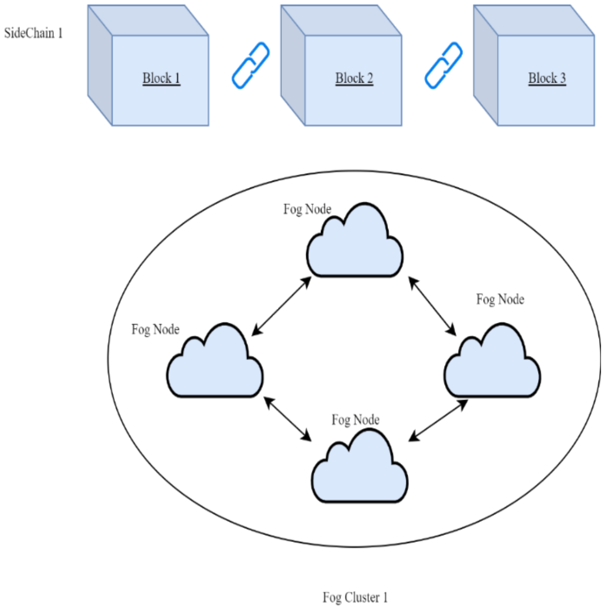


Figure 2: A Fog cluster.

Each block consists of a set of parameters that are essential for its functioning (see figure 3):

Block ID (same as Transaction ID): is a unique attribute that refers to a specific block.

Hash: As stated earlier, each block has a unique hash that ensures that the block is not altered or modified by any unauthorized users.

Previous Block Hash: This parameter creates the ledger structure by linking the blocks. When a block is modified, it will change its hash, thus making the link inconsistent. Thus, it is considered the primary defense mechanism in the blockchain.

Encrypted content: The healthcare data is encrypted and saved in this block attribute.

Signature: This attribute links the creation of the block to a specific entity without knowing the actual identity.

Timestamp: Records the date of the creation of the block.

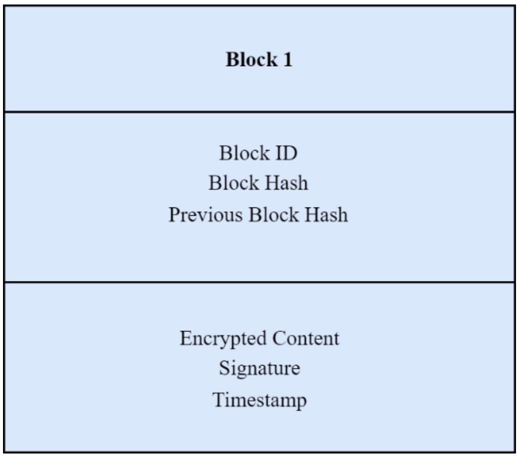


Figure 1: Block structure.

The main chain will be formed of the fog nodes in the system that do not belong to any cluster or preselected ones. This chain will be responsible for exchanging the transactions between the different sidechains. The communication between the sidechains and the main chain is performed through a protocol called a 2-way peg, which ensures the integrity of the data transmitted between them. This protocol is the most crucial component in cross-chain data exchange because it ensures the proper communication and information transfer from one chain to the other. Moreover, it obliges both chains through a digital contract to abide by the confirmed data transactions. Like side chain creation, the main chain involves the same features the plasma framework provides. The main chain is built based on the data provided for the ten selected nodes and allows the cross-chain data exchange using the 2-way-peg protocol.

Coordinator:

The coordinator presented in the model is a computer program responsible for the encryption/decryption process to ensure that the data is only accessible by authorized users. It plays a role in helping the recipient node to find the intended data after being uploaded to the main chain.

Encryption/Decryption process:

This process is done based on the private and public key concepts that can protect the data from unauthorized access. The public key will be shared between all the nodes and has the role of encrypting the data. On the other hand, the private key is given to specific nodes with the right to decrypt the data (see figure 4).

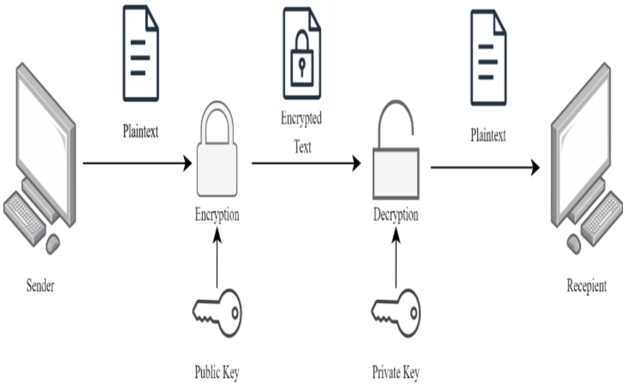


Figure 2: Public and Private Key Concept.

First, the node uploading the data will encrypt it using the public key provided to all the nodes (sent by the coordinator). In the next step, the coordinator will follow a predefined set of privileges to send the private key to the authorized recipients to ensure their right to access the data.

Implementing the different chains in the model is done through the plasma framework [37] that allows the available active nodes to be divided into chains. The framework creates the chains using an interface that permits adding the nodes to each chain (including the main chain) and setting the different attributes like proof of work, time to approve a block, and the data exchanged. Moreover, this software specifies the data exchange scheme that will select the data used in the model and the flow of data and transactions between the different entities in the simulation. The importance of this framework is in the 2-way-peg protocol, which ensures a smooth data transfer between any two chains. This protocol works by locking the data on the sending chain first, and then using a smart contract, the data will be transferred without any modification to its intended destination.

➢ **Pseudo code:**

This subsection will present the pseudo-code for the creation of the whole model (see figure 5). It will include the clustering method, the creation of the chains, and the data mapping to the chains.

```
k // minimum number of nodes per cluster
N // number of clusters
D //data
Exclude the ten least active nodes from D
Run k-means (D, k, N)
Establish SideChains + MainChain
Map the data to the different chains;
Run the transactions
```

The pseudocode outlines a process for clustering nodes and managing data with a focus on minimizing network load and ensuring efficient transaction handling. Initially, a minimum number of nodes per cluster (denoted by *k*) and the total number of clusters (*N*) are defined. The data set *D* is processed by first excluding the ten least active nodes, likely to enhance performance by removing underutilized or irrelevant nodes. The k-means algorithm is then applied to partition the data into *N* clusters, with k representing the minimum number of nodes per cluster. Following clustering, the system establishes both SideChains and a MainChain to handle and validate transactions securely. The data is then mapped to the appropriate chains based on the clustering results, and transactions are executed across the system, ensuring that the data is processed efficiently within the established structure.

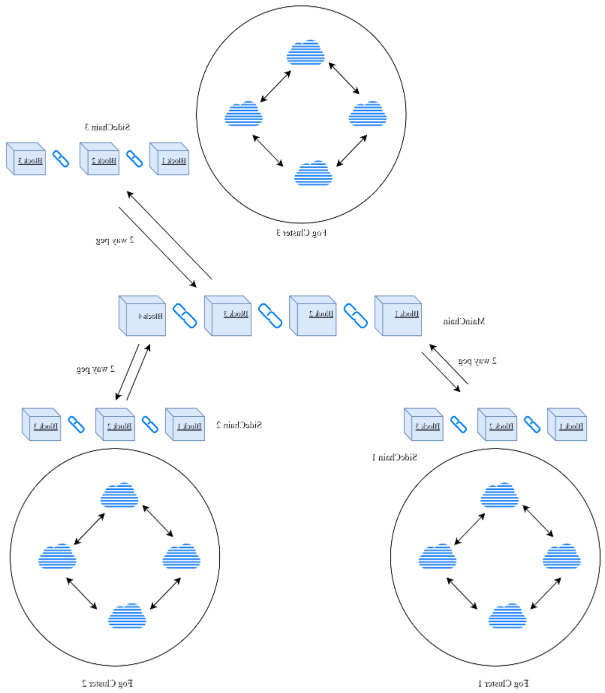


Figure 5: The complete model.

3.2 Model functionality

3.2.1 In the sidechain of a specific cluster

When a transaction is committed in a given cluster, a block is created containing the encrypted data with a specific hash pointing to the previous block's hash. This process uses the plasma framework that links the different blocks and updates the ledger at each node within the cluster. This block is then sent to the other fog nodes in the same cluster to check whether it is accepted (All/Majority of the users). If the approval is achieved, then the block is added to the sidechain of all the users (in the same cluster); else, it will be deleted.

➢ **Pseudo code:**

The pseudo-code below will present the sidechain's functionality in a specific cluster, including the approval mechanism and the node creation process.

```
Di → database of fog node i
Tj → Transaction j
Li → SideChain ledger of node i
Bj → Block of transaction j
If Tj.is_committed(Di)
        Bj.Create();
If a majority of nodes approve
        Bj.Add_To(Li);
Else
        Bj.Remove();
```

The pseudocode describes the functionality of a sidechain within a specific cluster, focusing on the approval mechanism and node creation process for transactions. Each fog node *i* has its own database (*Di*) and sidechain ledger (*Li*). When a transaction *Tj* occurs, it is first checked to see if it is committed to the node's database (*Tj*.Is_committed(*Di*)). If the transaction is committed, a new block *Bj* is created to represent *Tj*. Next, the block is subject to approval by a majority of the nodes in the cluster. If the majority approves, the block is added to the node's sidechain ledger (*Li*). If the approval is not obtained, the block is removed from the process. This ensures that only validated transactions are recorded on the sidechain, maintaining the integrity of the distributed ledger within the cluster.

3.2.2 Exchanging data between clusters

When two fog nodes in two different clusters need to exchange information, the data will be sent first to the main chain through the 2-way-peg protocol. This protocol is responsible for ensuring the integrity of the data while being transferred from one chain to the other. It will lock the data in the side chain and wait until the smart contract is initiated to transfer it from one chain to the other. Then, the fog node could get the intended data from the main chain to the targeted cluster (also using the 2-way-peg). It is worth mentioning here that at every included chain, a new block will be added to the ledger.

➢ **Pseudo code:**

This part shows the functionality of the whole model when communication between the different clusters is involved. It will include how the data is transferred from one block to the other and blocks creation locations.

```
1.  Di → database of fog node i
2.  Dy → database of fog node y
3.  Tj → Transaction j //sending data to main chain
4.  Tk → Transaction k //receiving data from the main chain
5.  Tm → Transaction m // posting data on the main chain
6.  Lm → Main Ledger
7.  Li → SideChain ledger of node i
8.  Ly → SideChain ledger of node y
9.  Bj → Block of transaction j
10. Bk → Block of transaction k
11. Bm → Block of transaction m
12. If Tj.is_Commited(Di)
13.       Bj.Create();
14. If a majority of nodes approve (Cluster i)
15.        Bj.Add_To(Li);
16. Else
17.        Bj.Remove();
18.        End_Process();
19. If Tm.is_Commited(Di)
20.       Bm.Create();
21. If a majority of nodes approve (Main Chain)
22.       Bm.Add_To(Lm);
23. Else
24.       Bm_Remove();
25.       End_Process();
26. If Tk.is_Commited(Di)
27.       Bk.Create();
28. If a majority of nodes approve (Cluster y)
29.       Bk.Add_To(Ly);
30. Else
31.       Bk.Remove();
32.       End_Process();
```

This pseudocode describes the process of transferring data between different clusters, focusing on transaction creation, approval, and the movement of data across sidechains and the main ledger. First, transactions *Tj*, *Tk*, and *Tm* are identified, with *Tj* representing the transaction sending data to the main chain, *Tk* receiving data from the main chain, and *Tm* posting data on the main chain. When a transaction, such as *Tj*, is committed in the database of a fog node *i* (*Di*), a corresponding block *Bj* is created. The block is then subject to approval by a majority of nodes in the cluster *i* (lines 14–17). If approved, it is added to the sidechain ledger *Li* of node *I*; otherwise, it is removed. Similarly, when *Tm* is committed, a block *Bm* is created and added to the main ledger *Lm* after approval by the majority of nodes in the main chain (lines 20–24). For transactions like *Tk*, when data is received from the main chain, the transaction is committed in node *i*, a block *Bk* is created, and it is added to the sidechain ledger *Ly* of node *y* after receiving approval from the majority of nodes in cluster *y* (lines 26–32). This process ensures that data flows between clusters in a secure and validated manner, with transaction blocks only being added to the respective ledgers after proper approval.

3.3 An Example: Demonstrating intercluster and intracluster communication

This section provides an example to illustrate how the proposed model facilitates both intercluster and intracluster communication. By exploring real-world scenarios, we demonstrate the functionality of the system, highlighting its mechanisms for secure data exchange within a single cluster and across multiple clusters.

3.3.1 Data exchange within a cluster

To understand intracluster communication, consider the case of a patient named Jad. After certain medical procedures, the responsible department uploaded Jad's data to the sidechain, making it accessible to other authorized entities within the same cluster. Table 3 displays the patient's record.

Table 3: Patient Record.

ID	Name	Temp	Weight
1234	Jad	38	70

The process begins with the fog nodes responsible for data management creating a new block containing Jad's encrypted data and an associated hash. This block is proposed to the cluster's distributed ledger. Once a majority of the fog nodes approve the block, it is added to the ledger. The approval and voting process is orchestrated by the plasma framework, which ensures consensus among the nodes and oversees decisions regarding the block creation. Following this successful transaction, the data is now securely stored on the sidechain. Suppose a doctor operating from another fog node within the same cluster needs access to Jad's data. The doctor must request permission through the coordinator, who manages access control. The coordinator provides the necessary private decryption key, enabling the requesting node to decrypt and access the specific content on the sidechain's distributed ledger. This example illustrates the robust and secure mechanisms employed by the model to facilitate data sharing within a single cluster.

3.3.2 Data exchange between clusters

To explain intercluster communication, consider a scenario involving a patient named Sami. Sami enters the emergency department, part of fog cluster 1, and his initial record is created as shown in Table 4.

Table 4: Another patient record.

ID	Name	Temp	Weight
4321	Sami	37	75

After being transferred to the X-ray department, which belongs to fog cluster 2, the doctors in the new department request access to Sami's medical data from fog cluster 1. In this case, fog cluster 1 initiates a request to create a new block on the main chain containing the relevant data. The coordinator plays a crucial role in facilitating this operation by managing the exchange of cryptographic keys between the two clusters. Using the 2-way-peg protocol, the sidechain of fog cluster 1 transfers the data to the main chain. Subsequently, a cross-chain operation enables fog cluster 2 to retrieve the data from the main chain and integrate it into its sidechain ledger. Finally, the coordinator provides the private decryption key to authorized personnel, allowing them to access the decrypted data. This example highlights the seamless and secure data exchange between clusters, demonstrating the efficiency of the proposed model in handling intercluster communication.

4 Experimental results

This part of the paper will examine the experimental results after the simulation of the suggested model. It will first introduce the simulation software used to get the results. Then it will go over the dataset. Finally, it will show the achieved results.

4.1 Simulation software

The Plasma framework was utilized to simulate the functionality of the proposed model. Plasma supports InterLedger Protocols (ILP), which facilitate seamless data exchange between the main blockchain and sidechains. The framework allows for efficient transaction handling while enabling integration with smart contracts, critical for secure and autonomous operations. In addition, Truffle and Ganache were employed for smart contract development and testing. Truffle provided the necessary tools to compile and deploy the contracts, while Ganache simulated the blockchain environment locally, ensuring smooth testing of transaction workflows before deployment.

4.2 Dataset

For this study, a simulated dataset representing healthcare systems was used to evaluate the model. The dataset included medical reports, such as X-ray images, prescriptions, and text-based patient records. These records reflect real-world scenarios where sensitive information must be securely managed and quickly accessed.

- Block Structure: Each block in both the sidechains and mainchain could contain only one medical report.
- Block Size: The maximum block size was restricted to 2.53 KB to align with typical constraints in blockchain-based systems. This assumption allowed for testing the system's ability to manage fine-grained data distribution effectively and securely.

4.3 Hardware setup

The simulation was performed on a system with the following specifications:

- Processor: Intel Core i7, 2.30 GHz
- RAM: 8 GB
- Storage: 1 TB HDD
- Operating System: Windows 64-bit

The hardware setup ensured sufficient resources to handle the computational demands of running the Plasma framework and its associated tools.

4.4 Performance metrics and results

The performance of the proposed model was evaluated based on the following key metrics:

- Transactions Per Minute (TPM): The number of transactions processed in a given time, reflecting the system's throughput.
- Latency: The time taken for a transaction to be verified and added to a block.
- Response Time: The delay experienced by users when querying data or submitting a transaction.
- Data Exchange Size: The amount of data exchanged between sidechains and the main blockchain.

The following flowchart provides a clear visualization of the experimental process, from data preparation to performance evaluation:

[Start] --> [Dataset Preparation] --> [Simulation Environment Setup]
--> [Deploy Clustering Algorithm] --> [Form Sidechains and Mainchain]
--> [Simulate Transactions on Plasma Framework]
--> [Evaluate Metrics: TPM, Latency, Response Time, Data Exchange Size]
--> [Analyze and Compare Results with Default Blockchain Model] --> [End]

This flowchart illustrates the systematic steps taken to implement and evaluate the proposed model, ensuring a structured approach to testing its capabilities.

4.5 Performance

This subsection will focus on presenting the results achieved after the simulation of the Plasma framework. The metrics "Transaction per minute," "Latency," and "Response time" will be used to study the performance and scalability of the model in comparison with the default blockchain approach. Moreover, the data exchange size will be tracked as well. In addition, the achieved results will be discussed and analyzed.

The experimental results of the proposed model are presented in detail, focusing on a comparative analysis of performance between a single blockchain and a model incorporating five sidechains. These results are illustrated in Figure 6, which highlights the number of committed transactions as a function of time. The graph demonstrates a clear advantage in productivity for the model with five sidechains compared to the single-chain approach.

The sidechain model exhibits a significantly higher number of committed transactions over the same time period, indicating enhanced throughput and overall efficiency. This improvement can be attributed to several key factors: 1) The clustering approach used in the sidechain model reduces the number of nodes participating in each cluster. This localization simplifies the approval process for transactions, resulting in faster consensus, 2) Smaller clusters decrease network congestion, as communication is confined to a limited subset of nodes, enhancing the speed and reliability of data exchange, 3) By distributing transaction activity across five sidechains, the model alleviates the workload on individual chains. This parallelization ensures smoother data exchange, even under high transaction loads, and 4) Each sidechain operates independently but adheres to the same security protocols, maintaining data integrity while improving processing times.

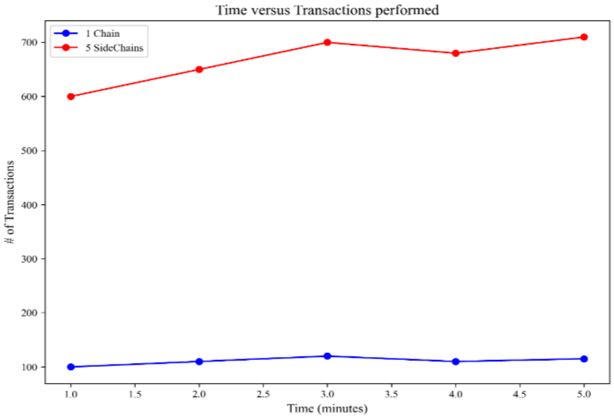


Figure 6: Time versus transactions performed.

Figure 7 examines a critical performance metric for blockchain systems: latency, defined as the time elapsed between the submission of a transaction and its final acceptance or rejection on the blockchain. This metric provides insight into the responsiveness and efficiency of the blockchain model. The results depicted in the graph highlight a noticeable reduction in latency for the proposed five-sidechain model compared to the traditional single-chain model. The decreased latency can be attributed to the introduction of sidechains significantly lowers the number of transactions each chain must handle. As a result, the time required to process and validate a transaction—whether to approve or reject it—is minimized, leading to faster transaction finalization. This is also attributed to efficient resource allocation where the smaller clusters created by the sidechain model distribute the computational workload across multiple chains, reduced the processing pressure on any single chain. This distribution allows for smoother data exchange and quicker transaction handling, despite the maximum block size of 2.53 KB.

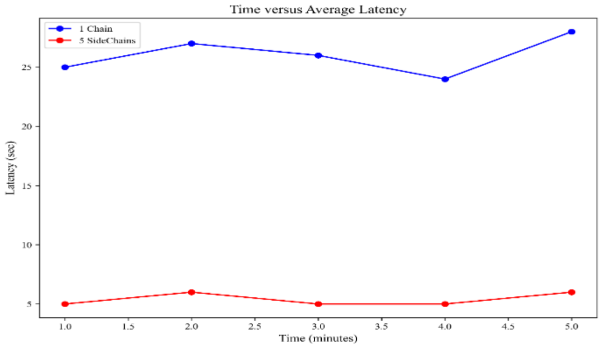


Figure 7: Time versus average latency.

The simulation also examined the data exchange size within the proposed model, which refers to the volume of data processed through the blockchain via transactions. Figure 8 illustrates the amount of data entering the two blockchain models over time. The graph clearly demonstrates that the proposed model significantly outperforms the traditional single-chain architecture, allowing a greater volume of data to be processed. The reduced transaction latency in the proposed model facilitated faster data approval, enabling higher throughput, and the enhanced architecture processed more transactions in less time, leading to greater data flow into the blockchain.

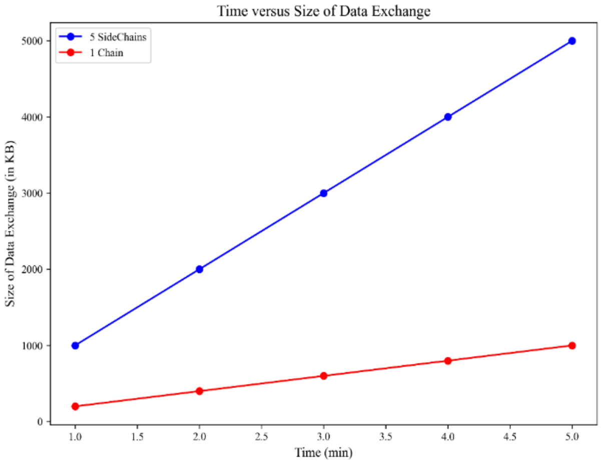


Figure 8: Time versus size of eata exchange.

Figures 9 and 10 examine the scalability of the proposed model by increasing the number of clusters to 30, assessing its effectiveness in adding blocks to the ledgers. The results clearly demonstrate that a higher number of clusters significantly enhances system performance by increasing both data exchange capacity and the number of transactions processed. This improvement highlights the model's scalability and its suitability for handling complex fog computing systems with a large number of nodes. The clustering technique eliminates the limitations of the single-chain structure, which often struggles with scalability due to the extensive validation required across numerous nodes. By leveraging sidechain architecture, the model reduces the effort involved in block creation and improves throughput. Furthermore, the sidechains operate without requiring constant connectivity to the main chain, effectively mitigating bottleneck issues and ensuring smooth data flow.

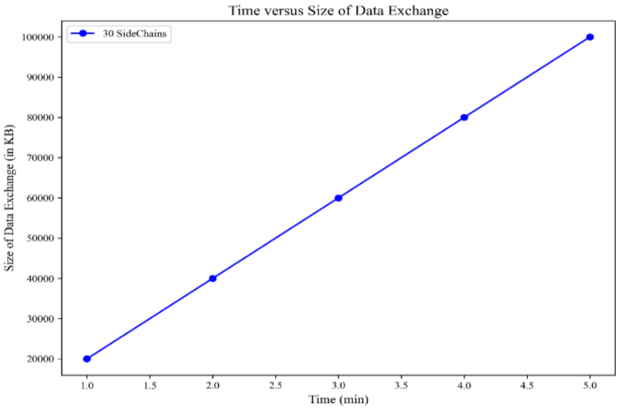


Figure 9: Time versus size of data exchange.

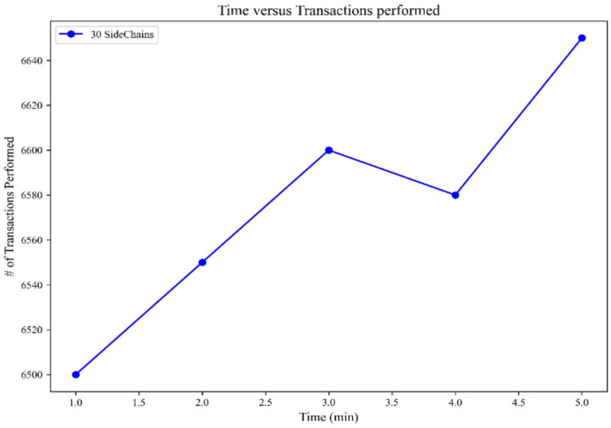


Figure 10: Time versus transactions performed.

We conducted a comparative analysis between our proposed model and the system presented in [35], which introduced a novel architecture designed to secure healthcare records in a fog computing environment. The referenced model emphasized the implementation of an additional fog layer to enhance system security, improve throughput, and deliver real-time services. To ensure a fair comparison, we simulated the same block sizes used in [35], specifically 1 KB and 0.1 KB. Figure 11 provides a visual comparison of throughput, measured in Transactions Per Second (TPS), between our model, utilizing 30 clusters, and the system from [35]. The bar graph highlights the clear superiority of our approach in handling a higher number of transactions. This performance advantage can be attributed to the clustering algorithm employed in our model, which organizes the system into sidechains. This approach significantly reduces the validation time required for each transaction by distributing the workload across multiple clusters, thereby enhancing overall throughput. The results underline the efficiency of our model in managing transactional processes within a fog computing environment, especially when compared to traditional architectures.

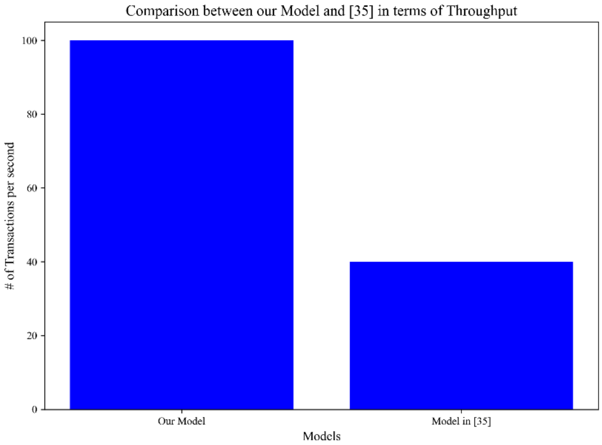


Figure 11: Comparison between our model and [35] in terms of throughput.

In addition to evaluating throughput, we also compared the latency performance of our model with the results presented in [35] as well as the latency of a traditional cloud-based model. Figure 12 clearly demonstrates the superior latency performance of our approach. The key factor contributing to this advantage is the use of sidechains, each consisting of fewer nodes compared to the single chain architecture. With a smaller number of nodes in each sidechain, the transaction validation process becomes significantly faster, as fewer participants are involved in the approval process. This reduction in the number of nodes per sidechain leads to quicker transaction commitment, thereby minimizing latency. The results highlight how our model's structure—leveraging sidechains and clustering—enables more efficient processing and faster response times, outperforming both the system in [35] and the conventional cloud model in terms of transaction validation speed.

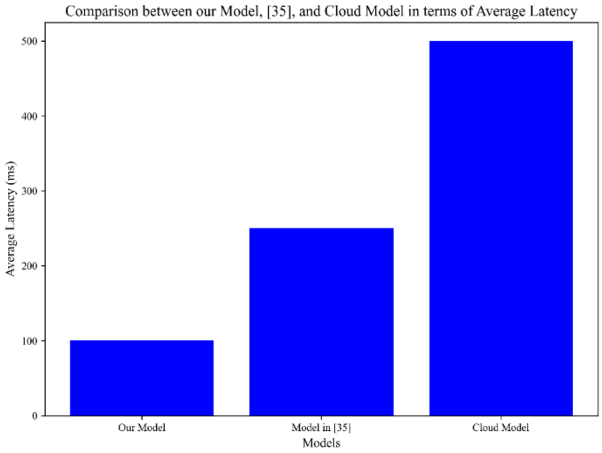


Figure 12: Comparison between our model, [35], and the cloud in terms of latency.

5 Conclusion and future work

Cloud computing has significantly enhanced information systems by providing greater processing power, flexibility, and storage. However, the increasing complexity of applications and the rise of IoT devices have outgrown the cloud's capabilities, leading to the introduction of the fog layer [38]. Fog computing offers faster response times and reduced latency, but remains vulnerable to malicious transactions between nodes [39]. This work proposes a fog model using a clustering algorithm to create sidechains for transaction monitoring, addressing scalability challenges and improving blockchain technology's limitations.

Future work could incorporate real-world data into simulations for more accurate results and identify issues not visible with synthetic data. Exploring different clustering algorithms or metrics, such as communication frequency, could provide new insights. Additionally, using alternative blockchain metrics, like response time, could offer a more comprehensive evaluation of the model's performance.

References

[1] Elhadad, A., Alanazi, F., Taloba, A. I., & Abozeid, A. (2021), Fog Computing Service in the Healthcare Monitoring System for Managing the Real-time Notification. *Journal of Healthcare Engineering*, 2022(1), 5337733, pp. 1–11. https://doi.org/10.1155/2022/5337733.

[2] Alazeb, A., & Panda, B. (2019), Ensuring Data Integrity in Fog Computing Based Health-Care Systems, *Security, Privacy, and Anonymity in Computation, Communication, and Storage*, vol 11611, Springer, pp. 63–77. https://doi.org/10.1007/978-3-030-24907-6_6.

[3] Jiang, Y., Wang, C., Wang, Y., & Gao, L. (2019), A Cross-Chain Solution to Integrating Multiple Blockchains for IoT Data Management. *Sensors*, 19(9), 2042. https://doi.org/10.3390/s19092042.

[4] Yi, S., Qin, Z., & Li, Q. (2015), Security and Privacy Issues of Fog Computing: A Survey, *Wireless Algorithms, Systems, and Applications*, vol 9204, Springer, pp. 685–695. https://doi.org/10.1007/978-3-319-21837-3_67.

[5] Alazeb, A., & Panda, B. (2019), Maintaining Data Integrity in Fog Computing Based Critical Infrastructure Systems. *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, IEEE, Las Vegas, NV, USA, pp. 40-47. https://doi.org/10.1109/csci49370.2019.00014.

[6] Amoli, P., Plosila, J., Tenhunen, H., Hämäläinen, T., & Hosseinpour, F. (2016), An Intrusion Detection System for Fog Computing and IoT-based Logistic Systems using a Smart Data Approach, *International Journal of Digital Content Technology and its Applications*, 10.

[7] Ismail, L., & Materwala, H. (2020), Blockchain Paradigm for Healthcare: Performance Evaluation. *Symmetry*, 12(8), 1200. https://doi.org/10.3390/sym12081200.

[8] Ziegler, M. H., Grossmann, M., & Krieger, U. R. (2019). Integration of Fog Computing and

Blockchain Technology Using the Plasma Framework, *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, Seoul, Korea (South), pp. 120-123. https://doi.org/10.1109/bloc.2019.8751308.

[9] Blockchain Metrics | Arcitura Patterns. (n.d.), *ARCITURA*. Retrieved on January 15, 2025, from http://patterns.arcitura.com/blockchain-patterns/blockchain-metrics.

[10] Gia, T. N., Ali, M., Dhaou, I. B., Rahmani, A. M., Westerlund, T., Liljeberg, P., & Tenhunen, H. (2017), IoT-based Continuous Glucose Monitoring System: A Feasibility Study, *Procedia Computer Science*, 109, pp. 327–334. https://doi.org/10.1016/j.procs.2017.05.359.

[11] Kua, J., Loke, S., Arora, C., Fernando, N., & Ranaweera, C. (2021), Internet of Things in Space: A Review of Opportunities and Challenges from Satellite-Aided Computing to Digitally-Enhanced Space Living. *Sensors*, 21(23), 8117. https://doi.org/10.3390/s21238117.

[12] How Google Workspace Uses Encryption to Protect Your Data. (n.d.). *Google*. Retrieved on January 15, 2025, from https://services.google.com/fh/files/misc/google-workspace-encryption-wp.pdf.

[13] Fatoum, H., Hanna, S., Halamka, J. D., Sicker, D. C., Spangenberg, P., & Hashmi, S. K. (2021), Blockchain Integration with Digital Technology and the Future of Health Care Ecosystems: Systematic Review, *Journal of Medical Internet Research*, 23(11), e19846. https://doi.org/10.2196/19846.

[14] A Cautionary Tale: How a Bug in Dropbox Permanently Deleted 8,000 of My Photos. *PetaPixel*. Retrieved on January 15, 2025 from https://petapixel.com/2014/07/31/cautionary-tale-bug-dropbox-permanently-deleted-8000-photos/.

[15] Nakamoto, S., (2008), Bitcoin: A Peer-to-Peer Electronic Cash System, Available at *SSRN*: https://ssrn.com/abstract=3440802 or http://dx.doi.org/10.2139/ssrn.3440802.

[16] Foley, S., Karlsen, J. R., & Putniņš, T. J. (2019), Sex, Drugs, and Bitcoin: How Much Illegal Activity is Financed through Cryptocurrencies? *The Review of Financial Studies*, 32(5), pp. 1798–1853. https://doi.org/10.1093/rfs/hhz015.

[17] Jiang, Y., Wang, C., Wang, Y., & Gao, L. (2019), A Cross-Chain Solution to Integrating Multiple Blockchains for IoT Data Management, *Sensors*, 19(9), 2042. https://doi.org/10.3390/s19092042.

[18] Back, S.A., Corallo, M., Dashjr, L., Friedenbach, M., Maxwell, G., Miller, A.K., Poelstra, A., & Timón, J, (2014), Enabling Blockchain Innovations with Pegged.

[19] Alam, T. (2019), IoT-Fog: A Communication Framework using Blockchain in the Internet of Things, *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 7(6). https://doi.org/10.36227/techrxiv.12657200.v1.

[20] Singh, P., Nayyar, A., Kaur, A., & Ghosh, U. (2020), Blockchain and Fog-Based Architecture for Internet of Everything in Smart Cities, *Future Internet*, 12(4), 61. https://doi.org/10.3390/fi12040061.

[21] Ngabo, D., Wang, D., Iwendi, C., Anajemba, J. H., Ajao, L. A., & Biamba, C. (2021), Blockchain-Based Security Mechanism for the Medical Data at Fog Computing Architecture of Internet of Things. *Electronics*, 10(17), 2110. https://doi.org/10.3390/electronics10172110.

[22] Agarwal, N. & Agarwal, G. (2017), Role of Cloud Computing in Development of Smart City. *International Journal of Science Technology & Engineering*, National Conference on Road Map for Smart Cities of Rajasthan, pp. 228-232.

[23] Devadass, L., Sekaran, S. S., & Thinakaran, R. (2017), Cloud Computing in Healthcare, *International Journal of Students' Research in Technology & Management*, 5(1), pp. 25–31. https://doi.org/10.18510/ijsrtm.2017.516.

[24] Choudhary, S., Jadoun, R., & Mandoria, H. (2016), Role of Cloud Computing Technology in Agriculture Fields, *Computer Engineering and Intelligent Systems*, vol 7, no. 3, pp. 1-7.

[25] Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012), Fog Computing and its Role in the Internet of Things, *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing* - MCC '12, ACM, Helsinki, Finland, pp. 13-16. https://doi.org/10.1145/2342509.2342513.

[26] Ema, R. R., Islam, T., & Ahmed, M. H. (2019), Suitability of Using Fog Computing Alongside Cloud Computing, *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, IEEE, Kanpur, India, pp. 1-4. https://doi.org/10.1109/icccnt45670.2019.8944906.

[27] Du, H., Leng, S., Wu, F., Chen, X., & Mao, S. (2020), A New Vehicular Fog Computing Architecture for Cooperative Sensing of Autonomous Driving, *IEEE Access*, vol 8, pp. 10997–11006. https://doi.org/10.1109/access.2020.2964029.

[28] Elhadad, A., Alanazi, F., Taloba, A. I., & Abozeid, A. (2022), Fog Computing Service in the Healthcare Monitoring System for Managing the Real-Time Notification, *Journal of Healthcare Engineering*, vol 1, pp. 1–11. https://doi.org/10.1155/2022/5337733.

[29] Quy, V. K., Hau, N. V., Anh, D. V., & Ngoc, L. A. (2021), Smart Healthcare IoT Applications Based on Fog Computing: Architecture, Applications, and Challenges, *Complex & Intelligent Systems*, vol 8, pp. 3805-3815. https://doi.org/10.1007/s40747-021-00582-9.

[30] Awaisi, K. S., Hussain, S., Ahmed, M., Khan, A. A., & Ahmed, G. (2020), Leveraging IoT and Fog Computing in Healthcare Systems, *IEEE Internet of Things Magazine*, 3(2), pp. 52–56. https://doi.org/10.1109/iotm.0001.1900096.

[31] Al-khafajiy, M., Baker, T., Asim, M., Guo, Z., Ranjan, R., Longo, A., Puthal, D., & Taylor, M. (2020), COMMITMENT: A Fog Computing Trust Management Approach, *Journal of Parallel and*

Distributed Computing, 137, pp. 1–16. https://doi.org/10.1016/j.jpdc.2019.10.006.

[32] Gu, K., Dong, X., & Jia, W. (2022), Malicious Node Detection Scheme Based on Correlation of Data and Network Topology in Fog Computing-Based VANETs, *IEEE Transactions on Cloud Computing*, 10(2), pp. 1215–1232. https://doi.org/10.1109/tcc.2020.2985050.

[33] Fantacci, R., Nizzi, F., Pecorella, T., Pierucci, L., & Roveri, M. (2019), False Data Detection for Fog and Internet of Things Networks, *Sensors*, 19(19), 4235. https://doi.org/10.3390/s19194235.

[34] Ahanger, T. A., Tariq, U., Ibrahim, A., Ullah, I., Bouteraa, Y., & Gebali, F. (2022), Securing IoT-Empowered Fog Computing Systems: Machine Learning Perspective. *Mathematics*, 10(8), 1298. https://doi.org/10.3390/math10081298.

[35] Mayer, A. H., Rodrigues, V. F., Costa, C. A., Righi, R. da, Roehrs, A., & Antunes, R. S. (2021), Fogchain: A Fog Computing Architecture Integrating Blockchain and Internet of Things for Personal Health Records. *IEEE Access*, vol 9, pp. 122723–122737. https://doi.org/10.1109/access.2021.3109822.

[36] Na, S., Xumin, L., & Yong, G. (2010), Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm, *2010 Third International Symposium on Intelligent Information Technology and Security Informatics*, IEEE, Jian, China, pp. 63-67. https://doi.org/10.1109/iitsi.2010.74.

[37] Poon, J. (2017), Plasma: Scalable Autonomous Smart Contracts.

[38] Dhaini, M., Jaber, M., Fakhereldine, A., Hamdan, S. and Haraty, R. A. (2021), Green Computing Approaches – A Survey. *Informatica* 45 (2021), pp. 1-12. https://doi.org/10.31449/inf.v45i1.2998.

[39] Alasady, A. S., Awadh, W. A., and Hashim, M. S. (2023), Non-Dominated Sorting Genetic Optimization-Based Fog Cloudlet Computing for Wireless Metropolitan Area Networks, *Informatica* 47 (2023), pp. 1–8, https://doi.org/10.31449/inf.v47i10.5118.

Fetal Health Risk Classification Using Important Feature Selection and Cart Model on Cardiotocography Data

Ahmad Ilham[1,2], Thahta Ardhika Prabu Nagara[1,2], Mudyawati Kamaruddin[3,4], Laelatul Khikmah[2,5], Teddy Mantoro[6]
[1]Department of Informatics, Universitas Muhammadiyah Semarang, Semarang, Indonesia
[2]Intelligent Data Science Research Group, Universitas Muhammadiyah Semarang, Semarang, Indonesia
[3]Postgraduate Program of Medical Laboratory Science, Universitas Muhammadiyah Semarang, Semarang, Indonesia
[4]Interdisciplinary Research Laboratory for Experimental Plasma Medicine, Universitas Muhammadiyah Semarang, Semarang, Indonesia
[5]Department of Statistics, Institut Teknologi Statistika dan Bisnis Muhammadiyah Semarang, Semarang, Indonesia
[6]School of Computer Science, Nusa Putra University, West Java, Indonesia
E-mail: ahmadilham@unimus.ac.id, mudyawati@unimus.ac.id, c2c023166@student.unimus.ac.id,
laelatul.khikmah@itesa.ac.id, teddy@ieee.org

Keywords: Cardiotocography, fetal health risk classification, important feature selection, cart, machine learning in healthcare

Received: January 22, 2024

Fetal mortality and newborn health issues require urgent attention because of high maternal and infant mortality rates during labor, highlighting the critical need for accurate fetal condition monitoring to reduce complications. This study proposes the development of a fetal health risk classification model based on Important Feature Selection (IFS) and a Classification and Regression Tree (CART) using cardiotocography (CTG) data from the UCI Machine Learning Repository. The IFS method was used to select the most relevant features, reduce model complexity, and increase generalization to prevent overfitting. The IFS-CART model was tested with 10-fold cross-validation and showed an accuracy of 94.50%, superior to the conventional CART, which only reached 93.83%. In addition, the average values of the True Positive Rate (TPR) and True Negative Rate (TNR) also increased, indicating that this model is effective in distinguishing normal, suspected, and pathological fetal conditions. Evaluation using the area under the curve receiver operating characteristic (AUC-ROC) showed that the model had high performance in detecting at-risk conditions, with an AUC of 0.981 for the "suspect" class. This finding confirmed that IFS-CART is not only accurate but also has high interpretability, making it easy for medical personnel to use for clinical decision support. The results of this study show that IFS-CART can serve as a reliable decision support system for real-time fetal health monitoring. Further implementation is expected to improve diagnostic accuracy and prevent complications during pregnancy and labor

Povzetek: Opisan je nov model za klasifikacijo tveganja za zdravje ploda z uporabo izbire pomembnih značilnosti in metode CART na podatkih kardiotokografije.

1 Introduction

Fetal hypoxia occurs when oxygen supply to the fetus is insufficient during labor. This condition can cause severe consequences, including intrapartum stillbirth, asphyxia, neonatal encephalopathy, neonatal death, and neurodevelopmental impairment [1]. The incidence of fetal hypoxia in European hospitals ranges from 0.06% to 2.8% [2]. Globally, intrapartum fetal hypoxia results in approximately 1.3 million stillbirths, 0.9 million neonatal deaths, and 0.6 to 1 million cases of long-term disability from neonatal hypoxic-ischemic encephalopathy annually [3]. These statistics underscore the urgency of addressing this issue to prevent further cases. Labor naturally induces a degree of hypoxic stress as uterine contractions (UC) potentially impair maternal placental perfusion, compromising fetal oxygen delivery. Clinicians face the challenge of identifying the small number of cases where physiological protective mechanisms fail to compensate for labor-induced hypoxic stress, leading to significant cerebral injury [4]. Effective fetal monitoring during labor is crucial to prevent the devastating effects of fetal hypoxia. However, it must also be sufficiently discriminatory to minimize unnecessary iatrogenic interventions, such as caesarean sections, which carry their own risks to both mother and baby [5].

Cardiotocography (CTG) is a commonly used screening tool for monitoring fetal conditions, such as fetal heart rate (FHR) and uterine contractions (UC). Unfortunately, the interpretation of CTG is subjective and depends on the clinician's experience, which often leads to erroneous diagnoses and unnecessary medical interventions [6]. This has led to an increased interest in machine learning to provide a more objective and accurate analysis. A promising approach is the classification and regression tree (CART), which has a structure that is easy to interpret [7] .

However, CART is prone to overfitting, particularly when all features in the dataset are used without selection. To overcome this drawback, this study proposes the integration of an important feature selection (IFS) method with CART. IFS helps select the most relevant features, reduce model complexity, and increase robustness to overfitting, thereby improving classification accuracy. In addition, IFS increases the resistance to overfitting, thereby improving classification accuracy.

The main contribution of this research is the development of an IFS-CART model that is not only accurate but also easy to interpret. The CART model is not only accurate but also easy to interpret, providing practical benefits for medical personnel performing risk assessments without the need for an in-depth understanding of computational models. This research also demonstrates how the combination of IFS and CART can overcome the limitations of previous state-of-the-art models, such as SVM and Naive Bayes, which, although accurate, lack high interpretability and require complex computations.

The remainder of this paper is organized as follows. Section 2 presents a comprehensive review of related studies, highlighting previous research and machine learning techniques. Section 3 outlines the research methodology, including the dataset, preprocessing steps, model development, and validation procedures, to ensure the reliability of the results. Section 4 presents the experimental findings with an in-depth discussion, provides critical insights, and presents the results in the context of existing literature. Finally, Section 5 draws conclusions from the study and proposes recommendations for future research to strengthen the findings presented.

2 Related works

Various machine learning techniques have been applied to analyze cardiotocography (CTG) data to improve the accuracy and reliability of fetal health condition diagnosis. Innovations in these techniques are evolving along with the increasing need for models that are not only accurate but also interpretative and easy to use in a clinical context.

Sahin and Subasi [8] initially investigated the integration of a support vector machine (SVM) with an empirical mode decomposition (EMD) metaheuristic technique to enhance the accuracy of fetal condition classification. The model attained an accuracy of 86% by utilizing a limited dataset of 90 samples. Nevertheless, they encounter challenges regarding computational complexity and constraints in the interpretation of the results. In the same study, the authors evaluated the naive bayes (NB) model, which is recognized for its simplicity and computational efficiency, achieving an accuracy of 96.77%. However, NB performance frequently deteriorates when applied to datasets containing irrelevant or redundant features, indicating the necessity for more effective feature selection in fetal health predictive models.

In 2016, Yılmaz [9] introduced and compared three neural network architectures: multilayer perceptron neural network (MLPNN), probabilistic neural network (PNN), and general regression neural network (GRNN). Each of these architectures was designed to address specific challenges in fetal health data analysis, especially in detecting complex nonlinear patterns in CTG data. The MLPNN model demonstrated the ability to detect nonlinear relationships with an accuracy of 90.35%, recall of 78.71%, specificity of 90.50%, precision of 84.44%, and F1-score of 81.47%. However, MLPNN require a long training time and are prone to overfitting, especially when the data features are large or poorly structured. In contrast, PNNs excel at classification speed and are well suited for processing biomedical data, such as physiological signals on CTG, with an accuracy of 92.15%, recall of 82.82%, specificity of 92.24%, precision of 87.63%, and F1-score of 85.16%. However, PNN require large memory resources, which is an obstacle in large-scale clinical applications. In contrast, the GRNN models offer continuous prediction capabilities with 91.86% accuracy, 83.92% recall, 92.62% specificity, 85.81% precision, and an 84.85% F1-score. However, the proposed GRNN model is less optimal for distinct classifications and requires careful parameterization to achieve the best performance. These limitations suggest that although neural networks are capable of good performance, their computational complexity and susceptibility to overfitting limit their applicability in clinical contexts where quick and intuitive interpretation is required.

More recently, Chuatak et al. [7] applied a classification and regression tree (CART) to a CTG dataset with 2,126 samples from the UCI Repository. The CART model is known for its easy-to-understand tree structure and ability to produce clear classification rules for medical personnel without requiring a deep understanding of the computational models. In this study, CART achieved an accuracy of 93.65%. However, the main limitation of CART is its susceptibility to overfitting when all dataset features are used without selection. Thus, additional techniques, such as boosting or bagging, are required to achieve optimal performance, especially on large datasets with many features.

In 2024, Shalini et al. [10] extended the exploration of machine learning models on CTG data by comparing various models, including Linear Regression, Random Forest, K-Nearest Neighbors (KNN), Gradient Boosting Classifier, Decision Tree, and Support Vector Classifier. Based on the same dataset of 2,126 samples, the Random Forest and Decision Tree models achieved 93% and 85% accuracy, respectively. The Linear Regression, KNN, and gradient boosting classifier models demonstrated varying performances, with accuracies ranging from 89% to 90%, whereas the support vector classifier obtained a lower accuracy of 81%. This study showed that although some of these models provide reasonably good results, they have limitations in both accuracy and interpretability, making them less than ideal in clinical contexts that require quick and accurate decisions.

To facilitate a clear comparison, Table 2 summarizes the performance of the various machine learning models on the CTG data.

Through these studies, it can be seen that neural network-based models, such as MLPNN, PNN, and GRNN, perform quite well in the classification and prediction of fetal conditions. However, their high computational requirements and interpretation complexity are obstacles in clinical applications that require fast and reliable decisions. In addition, traditional models such as SVM and NB demonstrate fast performance; however, they have limitations when handling data with redundant or overlapping features, which often results in decreased accuracy under certain conditions.

This study selected CART because of its ability to generate classification rules that are clear and easy to interpret by medical personnel, without the need for in-depth knowledge of computational models. However, without proper feature selection, CART is prone to overfitting when applied to large datasets with many features. To address this issue, this study proposes the integration of an essential feature selection (IFS) method with CART. IFS allows the model to retain only the most relevant features, thereby reducing complexity, increasing generalizability, and ensuring reliable results in clinical practice.

3 Methodology

3.1 Dataset

In this study we used the cardiotocography (CTG) dataset from the UCI Machine Learning Repository [11]. The dataset comprises 2126 entries with information on third-trimester pregnant women. It includes 21 features and one feature class (NSP) used to determine the fetal heart rate (FHR) and uterine contractions (UC) in the CTG dataset. A comprehensive overview of the CTG dataset used in this study is presented in Table 3.1.

3.2 Decision tree-based learning

Decision tree-based learning represents an effective approach for addressing regression and classification problems. Support Vector Machines (SVM) and Decision Trees (DT) are among the machine learning techniques utilized for these tasks [12]. DT offers distinct advantages, including independence from predictor parameter distribution assumptions and computational efficiency. Furthermore, decision trees can effectively handle missing data [13].

A hypothetical study employs a vector of two independent variables (X_1, X_2) to construct a tree. Figure 1 illustrates a model comprising four internal nodes and five leaves, used to estimate the target parameter. The tree's growth progresses from top to bottom, with initial comparisons made between X_1 and the threshold value T_1. Subsequent steps depend on whether X_1 exceeds T_1, determining the branch selection.

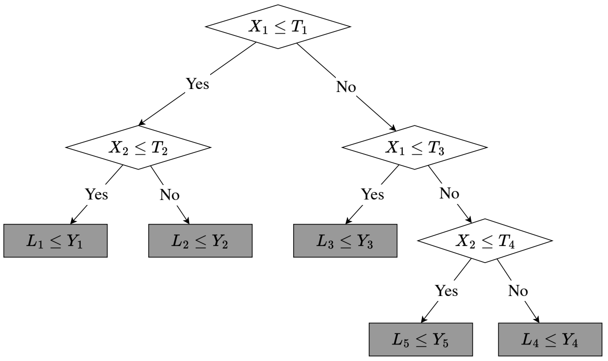


Figure 1: A typical decision trees

Various techniques exist for creating decision tree-based regressor classification models, including fuzzy ID3 [14], ID3 [15], C4.5 [16], and CART [17]. DT offers superior interpretability and visualization compared to black-box models like artificial neural networks, facilitating easier comprehension of results in regression and classification problems.

However, DT presents certain limitations. In regression analysis, it can only estimate continuous values. Additionally, the tree structure may become complex due to numerous branches in classification and regression tasks. The modeling data significantly influences the DT structure, potentially leading to inconsistent results and structural variations across datasets.

The CART model, widely adopted for its efficiency in processing qualitative and quantitative data [17], employs recursive binary splitting. This study utilizes the CART algorithm with the Gini splitting rule. The permutation approach and mini measure identify relevant dataset characteristics.

The percentage of samples in category k (0 or 1) at a node is expressed in Equation (1):

$$p_k = \frac{n_k}{n} \tag{1}$$

where p represents the data class probability at node τ. The mini impurity $i(\tau)$ is calculated using Equation (2).

$$i(\tau) = 1 - \sum_k p_k^2 \tag{2}$$

For a two-class problem in (3):

$$i(\tau) = 1 - p_1^2 - p_0^2 \tag{3}$$

The Gini impurity change when nodes split into subnodes τ_1 and τ_2 is formulated in Equation (4):

Table 1: Performances of several machine learning models on CTG data based on previous studies

Studies	Models	Accuracy (%)	Precision (%)	Recall (%)	Dataset Size
[8]	SVM + EMD	86.0	-	-	90
[8]	NB	96.77	-	-	90
[9]	MLPNN	90.35	84.44	78.71	2,126
[9]	PNN	92.15	87.63	82.82	2,126
[9]	GRNN	91.86	85.81	83.92	2,126
[7]	CART	93.65	-	-	2,126
[10]	Linear Regression	89	82.67	77	2,126
[10]	Random Forest	93	87.33	85.67	2,126
[10]	KNN	90	82.67	76.67	2,126
[10]	Gradient Boosting Classifier	90	80.67	77.67	2,126
[10]	Decision Tree	85	85	85	2,126
[10]	Support Vector Classifier	81	80.67	79.67	2,126

$$\Delta i(\tau) = i(\tau) - p_l i(\tau_1) - p_r i(\tau_2) \tag{4}$$

The algorithm tracks and aggregates each node's $i(\tau)$ drop. Using a single CART Gini Importance is formulated in (5):

$$I_G(\theta) = \sum_{\tau} \sum_{T} \Delta i_\theta(\tau, T) \tag{5}$$

The Gini importance identifies relevant features for the classification objective function, indicating feature frequency as a separator and its discriminative value. In equation (5), T represents the number of trees in the model, and $I_G(\theta)$ denotes the corresponding Gini importance.

3.3 Proposed model

Figure 2 illustrates the block diagram of our proposed model. We initiated the process by selecting the CTG dataset, comprising 2216 original samples. Subsequently, we divided and categorized the data to enhance system transparency. We implemented decision trees and rule-based classifiers using 10-fold cross-validation for modeling categorization. The folds served as training data to develop the model. We utilized the remaining data as a test set to validate the resulting model and determine performance metrics, such as accuracy.

3.3.1 Data preparation

Each feature in the cardiotocography (CTG) dataset exhibits distinct value ranges and units, such as fetal heart rate (FHR) and uterine contractions (UC). This scale disparity among features can lead to imbalances. Classification algorithms like CART may overemphasize features with larger values, compromising model performance. Consequently, normalization becomes essential to ensure equal feature contribution in machine learning processes and prevent model bias.

This study employs zero-mean normalization to transform numerical features. This technique ensures a mean of zero and uniform variance for each feature, enhancing data distribution consistency. Zero-mean normalization proves particularly valuable when handling high-variance data, enabling optimal model performance unaffected by feature scale differences.

The zero-mean normalization equation is expressed as in Equation (6):

$$x' = \frac{x - \mu}{\sigma} \tag{6}$$

where: x' represents the normalized feature value, x denotes the original feature value, μ signifies the average feature value, and σ indicates the feature's standard deviation.

This normalization step is crucial for maintaining data integrity and improving overall model accuracy in the analysis of cardiotocography dataset.

3.3.2 Feature preprocessing and selection

Feature selection using Important Feature Selection (IFS) follows the normalization process. This step is crucial in analyzing medical data like CTGs, which often contain numerous characteristics, not all relevant for fetal health status classification. IFS selects features that significantly contribute to classifying fetal status (normal, suspect, or pathological).

The proposed method employs the Information Gain criterion to assess feature relevance based on its impact in reducing data entropy. Features with minimal contributions are eliminated, retaining only important ones for model training. This approach reduces dataset dimensionality, enhances computational efficiency, and accelerates training. It also minimizes overfitting risk, resulting in a more accurate and interpretable model.

Equation (7) calculates feature significance in IFS using the Information Gain criterion:

Table 2: CTG dataset description

Features	Information
BV	FHR baseline (beats per minute)
AC	FHR of acceleration per second
FM	FHR of fetal movements per second
UC	FHR of uterine contractions per second
LD	FHR of light deceleration per second
SD	FHR of severe deceleration per second
PD	FHR during prolonged deceleration per second
ASTV	Percentage of time with abnormal short-term variability
MSTV	Mean short-term variability
ALTV	Percentage of time with abnormal long-term variability
MLTV	Mean long-term variability
HW	Width of the FHR histogram
HMin	Minimum FHR histogram
HMax	Maximum FHR histogram
HNMax	FHR of histogram peaks
NZ	The FHR of the histogram zeros is given by
HMo	Histogram mode
HMean	Histogram mean
HMed	Histogram median
HV	Histogram variance
HT	Histogram tendency
NSP	Fetal state class code (N=normal; S=suspect; P=pathologic)

$$I_G(S, A) = E(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} \times E(S_v) \qquad (7)$$

Where $I_G(S, A)$ represents feature A's information gain relative to dataset S, S_v is the data subset based on a particular value of feature A, and $E(S)$ measures uncertainty or entropy in dataset S.

Information gain quantitatively evaluates each feature's contribution to reducing classification target uncertainty. It provides a direct indication of a feature's ability to improve model class separation by calculating entropy change before and after feature use.

This method was chosen for its ability to explicitly measure individual feature relevance to the classification target, offering advantages over techniques like Principal Component Analysis (PCA). While PCA reduces dataset dimensionality, it transforms features into a new dimensional space without considering specific contributions to the classification target, often losing original feature interpretability.

The Information Gain criterion ensures retained original features directly relate to the classification target, maintaining model result interpretability and clarity. Features with low I_G values are eliminated, while those with high values are retained. This process reduces dataset complexity, improves computational efficiency, and minimizes overfitting risk.

By retaining the most relevant features, the model produces more accurate predictions while maintaining a clear relationship between important features and classification results. This approach is vital in clinical applications and data-driven analyses where interpretability and efficiency are key factors for model effectiveness.

3.3.3 Overfitting control and hyperparameter optimization

IFS-CART implements critical steps to maintain high accuracy while resisting overfitting. These steps include hyperparameter optimization and cross-validation. Overfitting occurs when a model performs exceptionally well on training data but fails to predict new data accurately, leading to decreased performance in real-world conditions. To mitigate this issue, the study optimizes key hyperparameters such as maximum tree depth (`max_depth`), pruning, and minimum samples per leaf. These optimizations reduce model complexity without sacrificing accuracy.

Cost-complexity pruning serves as a primary technique. This method eliminates tree branches that minimally contribute to model performance, maintaining model simplicity and preventing overfitting by reducing tree size. The pruning process is optimized using Equation (8).

$$R_\alpha(T) = R(T) + \alpha \cdot |T| \qquad (8)$$

Here, $R_\alpha(T)$ represents the tree's cost after considering its complexity, $R(T)$ denotes the total classification er-

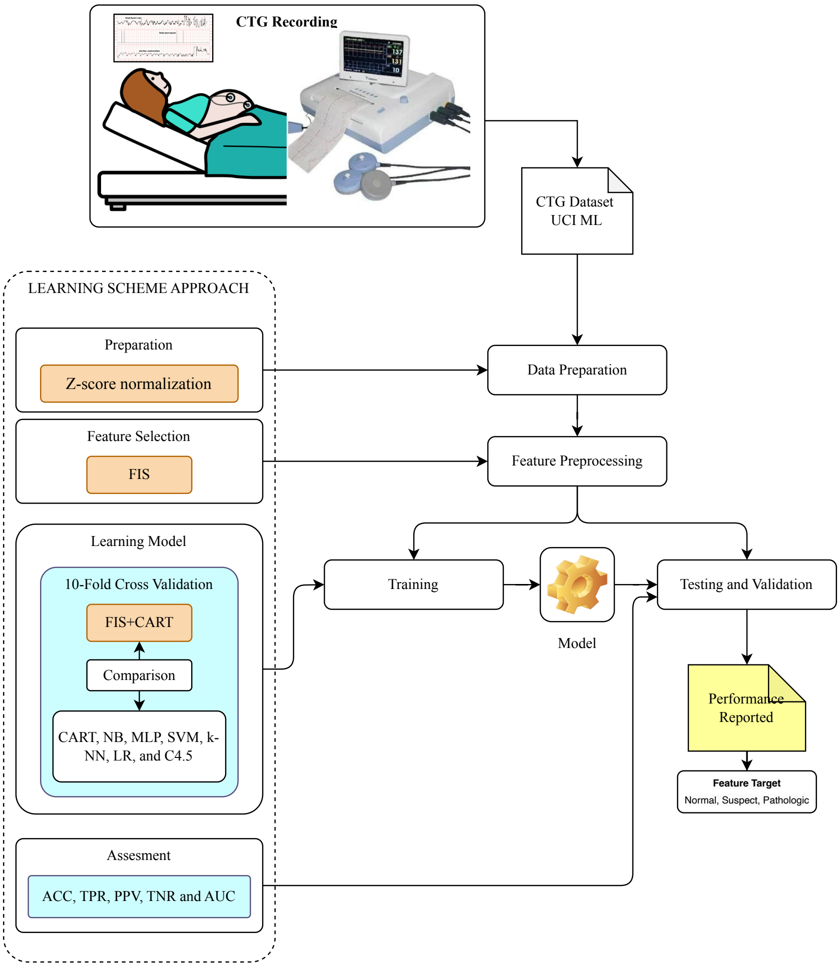


Figure 2: Schematic model of fetal health risk status classification using the CTG dataset

ror, and $|T|$ signifies the number of tree nodes. The parameter α controls complexity; a higher α value results in a smaller, simpler tree. This optimization retains only nodes that significantly contribute to classification results, enhancing model efficiency and interpretability.

3.3.4 Model stability and generalizability via cross-validation

Cross-validation ensures model stability and generalizability. This study employed 10-fold cross-validation to optimize tree structure and maintain performance consistency. The dataset was divided into 10 subsets. Each subset served as test data, while the remaining nine were used for training. This process repeated until all folds were tested.

Cross-validation tests the model across various data configurations, providing a comprehensive performance assessment. It reduces overfitting risk and enhances result reliability. By averaging performance metrics from each fold, the model demonstrates consistent performance independent of specific data.

This validation approach is crucial in medical applications. It increases confidence in the model's ability to function optimally beyond training data, including real clinical scenarios [18, 19, 20, 21, 22].

3.3.5 Evaluation of the implications for clinical applications

To ensure the comprehensive performance of the IFS-CART model, this study used several confusion matrix-based metrics. In addition to accuracy (ACC), metrics such as recall or true positive rate (TPR), precision or positive predictive value (PPV), and specificity or true negative rate (TNR) were evaluated to provide a deeper understanding of the model's ability to classify different classes (normal, suspect, and pathological). In Equations (9), (10), (11), and (12), provideive the appropriate formulas for the ACC, TPR, PPV, and TNR.

$$ACC = \frac{TP + TN}{TP + FP + FN + TN} \times 100 \quad (9)$$

$$TPR = \frac{TP}{TP + FN} \times 100 \quad (10)$$

$$PPV = \frac{TP}{TP + FP} \times 100 \quad (11)$$

$$TNR = \frac{TN}{TN + FP} \times 100 \quad (12)$$

Where TP (True Positive) denotes the number of correct predictions for each target class (normal, suspect, pathological), and FP (False Positive) and FN (False Negative) describe the prediction errors that may affect clinical decisions. Using these metrics, the model is not only measured based on overall accuracy but also assessed in terms of its ability to distinguish high-risk cases (suspect and pathological) from normal cases.

In addition, this study also applied the area under the curve receiver operating characteristic (AUC-ROC) to provide an additional evaluation of the model's ability to distinguish positive and negative classes at various classification thresholds. The AUC-ROC is particularly relevant in the medical context as it can assess how well the model predicts suspect and pathological conditions compared to normal classes. The ROC curve plots the TPR against the false positive rate at various thresholds, and AUC values close to 1.0 indicate that the model has excellent and consistent classification ability.

4 Results and discussion

A computing platform with an Intel Core i5 2.5GHz dual-core CPU, 16 GB of RAM, and the 64-bit operating system macOS Catalina was used for the experiments. KNIME version 4.7.0 produced model performance as the calculation output, including accuracy, recall, and Precision.

4.1 Results

First, we applied the CART model without importance feature selection (IFS) on the CTG dataset. All features in this dataset are used for the optimal classification analysis of the data. The experimental results are presented in Table 4.1. The model produced good ACC (93.83%), and the average values of the TPR (93.83%), PPV (93.83%), and TNR (96.91%) measures also obtained good average values. The results are quite good, but this model still indicates overfitting, which can be seen in some incorrect predictions that should be predicted correctly.

In the second experiment, the importance feature selection (IFS) method was implemented to select important and influential features to tackle overfitting in the CART model. Figure 3 presents the results of analyzing relevant features using IFS obtained from the CTG dataset. We found that the MSTV feature had the greatest effect with an importance value of more than 20.50% when using all features in the CTG dataset. Some features, including NZ and SD, had no impact on the CART model's development, where they were of 0% importance in tree building. The HT feature had the lowest effect (0.95%) compared to the other features. The results show that MSTV, ASTV, ALTV, and HMe have the highest significance. Based on this result, all features with less than 2% importance were removed from the new dataset; thus, 18 features remained in this process. The new CTG data is then used for reclassification using the CART-based important feature selection model (IFS-CART). The aim of this stage is to obtain logic classification results based on important features. The results of the second set of experiments are presented in Table 4.

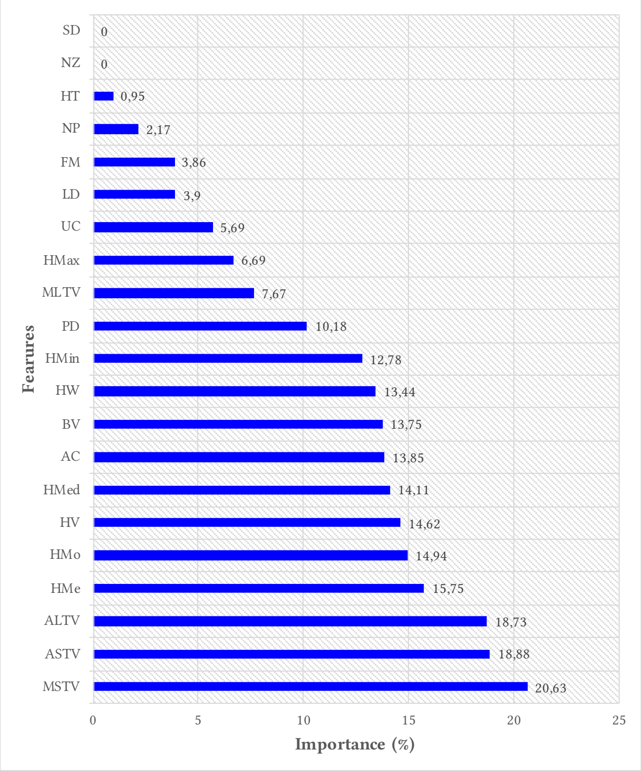


Figure 3: Feature importance of the proposed CART model on the CTG dataset using all features

As can be seen in Table 4, this model resulted in an ACC (94.50%) which is quite impressive as it is up 0.67% from the first experiment results. Meanwhile, the average values of TPR (94.49%), PPV (94.47%), and TNR (97.25%) across all classes also improved compared to the first experimental results. In this case, the proposed model mostly produced relatively good classification averages and obtained impressive class prediction results. Based on these results, the proposed model is quite promising because it can improve the classification performance of the CART model on the CTG dataset with a superb average value.

A more detailed comparison of the first and second ex-

Table 3: Confusion matrix of CART without IFS on the CTG dataset

		Predicted Class			ACC (%)	TPR (%)	PPV (%)	TNR (%)
		(1)	(2)	(3)				
Targeted Class	(1)	446	35	6	93.83	91.58	90.84	95.51
	(2)	37	467	4		91.93	92.66	96.23
	(3)	8	2	485		97.98	97.98	98.99
Average						93.83	93.83	96.91

Note: (1) Normal; (2) Suspect; (3) Pathological

Table 4: Confusion matrix of CART with IFS on the CTG dataset

		Predicted Class			ACC (%)	TPR (%)	PPV (%)	TNR (%)
		(1)	(2)	(3)				
Targeted Class	(1)	443	33	11	94.50	90.97	92.87	96.61
	(2)	29	476	3		93.70	93.33	96.54
	(3)	5	1	489		98.79	97.22	98.59
Average						94.49	94.47	97.25

Note: (1) Normal; (2) Suspect; (3) Pathological

periments is presented in Table 5. The best model performance results are indicated by bold numbers. As shown in Table 5, the second experiment with ACC value (94.50%) outperformed the first experiment. TPR outperformed the first experiment in classes 2 and 3 with values of (2 = 93.70%) and (3 = 98.79%), respectively. In addition, PPV and TNR also outperformed in class (1) and (2), with the respective values of PPV in class (1 = 92.87%) and class (2 = 93.33%), while the respective values of TNR in class (1 = 96.61%) and class (2 = 96.54%). In contrast to the first experiment, the classification performance was superior only in class TPR (1= 91.58%) and only for PPV and TNR in class 3 with values of 97.98% and 98.99%, respectively. In the second experiment, the average TPR, PPV, and TNR values were superior to those in the first experiment, with average values of 94.49%, 94.47%, and 97.25%, respectively. Based on the experimental results, overall, the proposed model in the second experiment outperforms the first experiment in which the ACC and average values of TPR, PPV, and TNR were quite impressive.

In the third experiment, we compared the proposed model with other models such as NB, MLP, SVM, k-NN, LR, and C4.5, as shown in Table 6. The results show that the accuracy (ACC), error classification (E-CL), correct classification (C-CL), and incorrect classification (I-CL) of our proposed model gives better results than the other models with respective values of ACC (94.73%), E-CL (5.27%), C-CL (2014), and I-CL (112). The second, third, and fourth best models based on accuracy were C4.4 with ACC (92.24%), k-NN with ACC (90.31%), and Logistic Regression with ACC (89.24%). The best result was MLP with ACC (81.70%).

To ensure that the IFS-CART model is not only accurate but also resistant to overfitting and has good generalizability, an evaluation was conducted on the training data and separate test datasets. This evaluation involves the calculation of accuracy metrics as the main performance indicator, which is reinforced by AUC-ROC analysis to provide a more comprehensive understanding of the model's ability to distinguish each target class: normal (1.0), suspect (2.0), and pathological (3.0). The results are shown in Figures 4 and 5.

As shown in Figure 4, the performance of the IFS-CART model remained consistent between the training and test data, with an accuracy of 94.73% on the training data and 93.80% on the test data. This minimal difference in performance indicates that the model does not experience overfitting and can generalize well to new data. In addition to accuracy, similar consistency was observed in the precision, recall, and specificity metrics, indicating that the model maintained high performance on both datasets. These results demonstrate that hyperparameter optimization, including pruning and limiting the maximum tree depth, successfully prevented the model from learning noise or irrelevant patterns from the training data. As shown in Figure 4, the performance of the IFS-CART model remained consistent between the training and test data, with an accuracy of 94.73% on the training data and 93.80% on the test data. This minimal difference in performance indicates that the model does not experience overfitting and can generalize well to new data. In addition to accuracy, similar consistency was observed in the precision, recall, and specificity metrics, indicating that the model maintained high performance on both datasets. These results demonstrate that hyperparameter optimization, including pruning and limiting the maximum tree depth, successfully prevented the model from learning noise or irrelevant patterns from the training data.

Table 5: Comparison results between the first and second experiment of CTG dataset

		Predicted Class				ACC (%)	TPR (%)	PPV (%)	TNR (%)
			(1)	(2)	(3)				
CART	Targeted Class	(1)	446	35	6	93.83	91.58	90.84	95.51
		(2)	37	467	4		91.93	92.66	96.23
		(3)	8	2	485		97.98	97.98	98.99
	Average						93.83	93.83	96.91
CART+IFS	Targeted Class	(1)	443	33	11	94.73	90.97	92.87	96.61
		(2)	29	476	3		93.70	93.33	96.54
		(3)	5	1	489		98.79	97.22	98.59
	Average						94.49	94.47	97.25

Note: (1) Normal; (2) Suspect; (3) Pathological

Table 6: Comparative effectiveness of proposed and traditional machine learning models on the CTG dataset

Approaches	Models	ACC (%)	E-CL* (%)	C-CL**	I-CL***
Machine Learning Traditional	NB	83.90	16.04	1785	341
	MLP	81.70	18.30	1737	389
	SVM	87.58	12.42	1862	264
	k-NN	90.31	9.69	1920	206
	LR	89.37	10.63	1900	226
Decision Tree-Based	C4.5	92.24	7.76	1961	165
	CART	93.83	7.79	1962	164
	Proposed Model	94.73	5.27	2014	112

Note: *Error Classification (E-CL), **Correctly Classified (C-CL), ***Incorrectly Classified (I-CL)

The evaluation of the AUC-ROC also confirms that the model is excellent in distinguishing the positive and negative classes. Figure 5 shows the ROC curves for each target class.

As shown in Figure 5, for the suspect class, the proposed model performed best with an AUC of 0.981. The ROC curve for this class is close to the upper left corner of the graph, indicating that the model achieved a high detection rate with minimal error. This performance is particularly relevant in a clinical context because the suspect class requires more intensive monitoring and early intervention to avoid the development of more serious conditions. In the pathological class, the model also performed reasonably well, with an AUC of 0.778. Although these results were sufficient to detect most high-risk conditions, some prediction errors indicated a feature overlap between the suspect and pathological classes. This reduces the accuracy of classification in certain cases, but it still provides a strong basis for medical personnel to detect critical conditions early. Further improvements to feature selection and threshold optimization can improve accuracy and reduce errors in this class. In contrast, the model performed very poorly in the normal class, exhibiting an AUC of 0.097. The ROC curve for this class almost follows a random line, indicating that the model has difficulty distinguishing between the normal and suspect classes. This low performance is likely due to an imbalance in the number of samples or feature overlap between the two classes, which makes prediction under normal conditions inaccurate. This emphasizes the importance of improving feature selection and dataset balancing to improve prediction accuracy in the normal class and reduce the possibility of unnecessary false positives. Overall, the AUC-ROC evaluation results showed that the IFS-CART model has great potential for detecting high-risk conditions, such as suspicious and pathological conditions, with good accuracy. Despite the weakness in normal class detection, this model remains relevant as a reliable diagnostic tool for real-time monitoring of fetal health. With additional optimization, the model can further strengthen support for rapid and accurate clinical decision-making, ensuring that at-risk conditions are detected in time to prevent more serious complications.

Finally, we compared the proposed model with previous studies. The results are presented in Table 4.1.

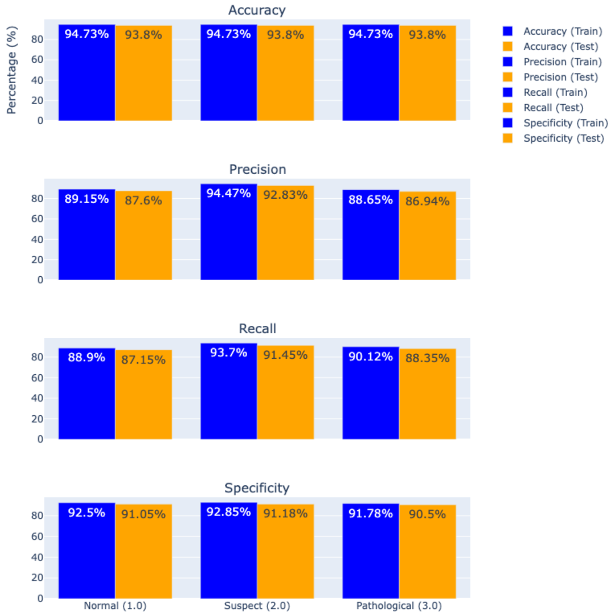


Figure 4: Comparison of the performance of the IFS-CART model with training and test data for each class (normal, suspect, and pathological)

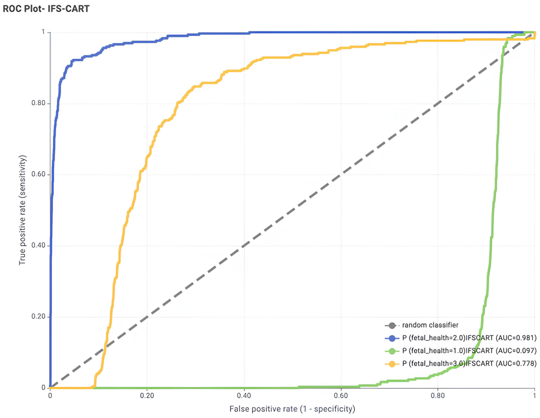


Figure 5: Receiver operating characteristic curves and area under the curve of the IFS-CART model for each fetal health class: normal (1.0), suspect (2.0), and pathological (3.0)

The comparison results presented in Table 7 show that the proposed IFS-CART model achieved the highest accuracy of 94.73%, outperforming various machine learning models reported in previous studies. The conventional CART model reported in reference [7] achieved an accuracy of 93.65%, whereas the random forest and logistic regression models reported in reference [10] only achieved an accuracy of 85%. The Decision Tree and Gradient Boosting Classifier models, with 93% and 90% accuracy, respectively, and the K-Nearest Neighbor (k-NN) model, with 90% accuracy [10], also fall below the accuracy level of the proposed model. In addition, some neural network-based models, such as the Multi-Layer Perceptron Neural Network (MLPNN) with an accuracy of 90.35%, the Probabilistic Neural Network (PNN) with an accuracy of 92.15%, and the General Regression Neural Network (GRNN) with an accuracy of 91.86% [9], although performing quite well, still show lower accuracy than the IFS-CART model. Based on these data, the proposed model showed a significant improvement in accuracy compared with other existing models, making it the highest performing model for fetal health risk classification in the context of this study.

Table 7: Comparison between the proposed model and prior studies

Studies	ACC (%)
CART [7, 11]	93.65
Random Forest [10]	85
Decision Tree [10]	93
K-Nearest Neighbor [10]	90
Logistic Regression [10]	85
Gradient Boosting Classifier [10]	90
Support Vector Machine [10]	81
MLPNN [9]	90.35
PNN [9]	92.15
GRNN [9]	91.86
Proposed Model	**94.73**

4.2 Discussions

The experimental results demonstrate that integrating the Important Feature Selection (IFS) method with the Classification and Regression Tree (CART) algorithm significantly improves fetal health classification using cardiotocography (CTG) data. The application of IFS successfully reduces dataset dimensionality by identifying the most relevant features for classification, resulting in a simpler model with reduced risk of overfitting. The IFS-CART model not only achieves high accuracy but also maintains interpretability, which is crucial in clinical applications, allowing medical professionals to understand and effectively use this model in decision-making.

Comparison of the first and second experiment results shows significant performance improvement after IFS implementation. Accuracy increased from 93.83% to 94.50%, average True Positive Rate (TPR) improved from 93.83% to 94.49%, and positive predictive value (PPV) rose from 93.83% to 94.47%. These results confirm that proper feature selection is critical for enhancing prediction accuracy by reducing the influence of less relevant features. Additionally, the increase in True Negative Rate (TNR) from 96.91% to 97.25% indicates the model's improved ability to identify negative cases, which can help reduce unnecessary medical interventions.

In the third experiment, the proposed IFS-CART model was compared with other machine learning models from previous studies to assess its superiority. With the highest accuracy of 94.73%, IFS-CART outperformed various other models. For instance, conventional CART from previous references achieved 93.65% accuracy [7], while random forest and logistic regression had 85% accuracy [10], demonstrating that IFS-CART offers higher accuracy and better efficiency. The Decision Tree model with 93% accuracy [10] approached IFS-CART's performance but still fell short. These results highlight that although traditional decision tree models offer good interpretability, the IFS-CART model provides superior accuracy crucial for reliability in medical applications.

Furthermore, K-Nearest Neighbor and Gradient Boosting Classifier algorithms, each with 90% accuracy [10], as well as Support Vector Machine (SVM) with 81% accuracy [10], showed limitations in handling the complexity of fetal health classification compared to IFS-CART. Neural network-based models, such as multi-layer perceptron neural network (MLPNN) with 90.35% accuracy, Probabilistic Neural Network (PNN) with 92.15%, and General Regression Neural Network (GRNN) with 91.86% [9], also approached IFS-CART's performance but required significantly more computational resources and were prone to overfitting. In contrast, IFS-CART is not only more computationally efficient but also easier to understand, allowing medical professionals to interpret results quickly.

The Area Under Curve-Receiver Operating Characteristic (AUC-ROC) evaluation shows that IFS-CART performs well in distinguishing the ”suspect” class with an AUC value of 0.981 and the ”pathological” class with an AUC of 0.778. High performance in the ”suspect” class is particularly important in clinical settings, where at-risk cases require intensive monitoring and early intervention to prevent serious complications. However, the low AUC value for the ”normal” class (0.097) indicates the model's difficulty in distinguishing between normal and at-risk conditions. This difficulty may be due to sample imbalance and feature overlap between normal and at-risk classes, which can affect prediction accuracy for these classes. These results highlight the importance of additional optimization, such as class data balancing and more effective feature selection, to improve model performance in detecting normal cases.

Additionally, the application of cost-complexity pruning and tree depth limitation successfully reduced the risk of overfitting, reflected in the consistency of model performance between training and testing data. From a practical perspective, the IFS-CART model provides significant value to medical professionals due to its high interpretability. This model allows clinicians to utilize prediction results quickly and accurately without requiring a deep understanding of complex computational algorithms. This easy interpretability is highly relevant in clinical contexts, where quick and accurate decisions are crucial for reducing the risk of complications during childbirth.

Furthermore, the model's flexibility in handling data variations and its ability to be integrated into real-time fetal health monitoring systems demonstrate its initial potential as a reliable and effective solution in medical applications. However, these findings require further validation using broader and more diverse datasets to ensure model generalization and reliability across various clinical conditions. Sub optimal performance in detecting normal cases indicates the need for further strategies, such as enhanced feature selection algorithms or class data balancing, to reduce false-positive predictions and improve accuracy.

Overall, this study shows that the combination of IFS and CART is an effective strategy for classifying fetal health risks using CTG data. These findings are in line with previous studies that have demonstrated the benefits of integrating feature selection techniques with machine learning algorithms to improve medical classification accuracy [2]. For example, [23] found that the use of feature selection improved model performance in predicting pregnancy outcomes based on CTG data. Finally, our findings not only improve accuracy, but also offer important interpretability and efficiency in medical applications. With continuous optimization and validation, this model can support faster and more accurate clinical decision-making.

5 Conclusions

In conclusion, this study highlights the theoretical and practical implications of developing a fetal health risk classification model based on Essential Feature Selection (IFS) and Classification and Regression Trees (CART) tested on cardiotocography (CTG) data. The results of this study confirm the importance of relevant feature selection for improving the accuracy of prediction models, which supports more accurate clinical decision-making and potentially reduces the risk of maternal and fetal health complications.

The primary contribution of this study was the finding that the integration of IFS and CART can improve the consistency and interpretability of the model, making it more practical for use by medical personnel without in-depth knowledge of computational algorithms. This finding could encourage the use of predictive models to detect fetal risk conditions, which is clinically crucial in managing cases of labor-related complications. Although the model performed well in the “at-risk” category, this study identified the need for improved accuracy in distinguishing normal fetal conditions, especially to reduce the likelihood of unnecessary medical interventions.

In future research, subsequent investigations should focus on refining feature selection techniques and data balancing methods to enhance prediction accuracy across all fetal condition categories. Furthermore, additional testing with more extensive and diverse datasets will strengthen the generalizability of the model in various clinical contexts.

The exploration of more advanced machine learning models and automation in CTG data assessment are also crucial steps to improve the accuracy and efficiency of predicting real-time fetal health.

Author contributions

Ahmad Ilham: Conceptualized and led the research, designed the methodology, developed models, analyzed performance, supervised the project, and significantly contributed to manuscript writing. Thahta Ardhika Prabu Nagara: Assisted in data collection and preprocessing as part of student mentorship. Mudyawati Kamaruddin: Provided expertise in interpreting medical data, particularly cardiotocography (CTG), and reviewed clinical aspects and implications of research findings. Laelatul Khikmah: Performed statistical analysis, ensured data normalization, and contributed to experimental design and validation. Teddy Mantoro: Provided guidance on machine learning approach, ensured compliance with computational standards, and assisted with manuscript revision by offering technical insights for model refinement.

Acknowledgement

The authors would like to thank to Director of Research Muhammadiyah Higher Education Research and Development Council Muhammadiyah Central Leadership (DRMHER–DCMCL). This research work was funded by DRMHER–DCMCL under Grant No. 1687.249/PD/I.3/D/2022.

References

[1] C. E. Wood and M. Keller-Wood, "Current paradigms and new perspectives on fetal hypoxia: implications for fetal brain development in late gestation," *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*, vol. 317, no. 1, pp. R1–R13, 2019. https://doi.org/10.1152/ajpregu.00008.2019

[2] F. Francis, S. Luz, H. Wu, S. J. Stock, and R. Townsend, "Machine learning on cardiotocography data to classify fetal outcomes: A scoping review," *Computers in Biology and Medicine*, vol. 172, p. 108220, 2024, https://doi.org/10.1016/j.compbiomed.2024.108220

[3] D. Ayres-de-Campos, *Obstetric emergencies*. Springer, 2016. Springer International Publishing, 2017. https://doi.org/10.1007/978-3-319-41656-4

[4] J. E. Lawn, H. Blencowe, P. Waiswa, A. Amouzou, C. Mathers, D. Hogan, V. Flenady, J. F. Frøen, Z. U. Qureshi, C. Calderwood, S. Shiekh, F. B. Jassir, D. You, E. M. McClure, M. Mathai, S. Cousens, and others, "Stillbirths: rates, risk factors, and acceleration towards 2030," *The Lancet*, vol. 387, no. 10018, pp. 587–603, 2016. https://doi.org/10.1016/S0140-6736(15)00837-5

[5] S. Rajagopal, K. Ruetzler, K. Ghadimi, E. M. Horn, M. Kelava, K. T. Kudelko, I. Moreno-Duarte, I. Preston, L. L. R. Bovino, N. R. Smilowitz, A. Vaidya, and the American Heart Association Council on Cardiopulmonary, Critical Care, Perioperative and Resuscitation, and the Council on Cardiovascular and Stroke Nursing, "Evaluation and Management of Pulmonary Hypertension in Noncardiac Surgery: A Scientific Statement From the American Heart Association," *Circulation*, vol. 147, no. 17, pp. 1317–1343, 2023. https://doi.org/10.1161/CIR.0000000000001136.

[6] K. Barnova, R. Martinek, R. Vilimkova Kahankova, R. Jaros, V. Snasel, and S. Mirjalili, "Artificial Intelligence and Machine Learning in Electronic Fetal Monitoring," *Archives of Computational Methods in Engineering*, vol. 31, no. 5, pp. 2557–2588, Jul. 2024. https://doi.org/10.1007/s11831-023-10055-6

[7] J. V. Y. Chuatak, E. R. C. Comentan, R. L. H. G. Moreno, R. K. C. Billones, R. G. Baldovino, and J. C. V. Puno, "A decision tree-based classification of fetal health using cardiotocograms," 2023, p. 020003. https://doi.org/10.1063/5.0111194

[8] H. Sahin and A. Subasi, "Classification of the cardiotocogram data for anticipation of fetal risks using machine learning techniques," *Appl Soft Comput*, vol. 33, pp. 231–238, Aug. 2015. doi:10.1016/j.asoc.2015.04.038

[9] E. Yılmaz, "Fetal State Assessment from Cardiotocogram Data Using Artificial Neural Networks," *J Med Biol Eng*, vol. 36, no. 6, pp. 820–832, Dec. 2016. doi:10.1007/s40846-016-0191-3

[10] Y. Salini, S. N. Mohanty, J. V. N. Ramesh, M. Yang, and M. M. V. Chalapathi, "Cardiotocography Data Analysis for Fetal Health Classification Using Machine Learning Models," *IEEE Access*, vol. 12, pp. 26005–26022, 2024. https://doi.org/10.1109/ACCESS.2024.3364755

[11] D. Campos and J. Bernardes. "Cardiotocography," *UCI Machine Learning Repository*, 2000. [Online]. Available: https://doi.org/10.24432/C51S4N

[12] F. Mesa, R. Ospina-Ospina, and D. Devia-Narvaez, "Comparison of Support Vector Machines and Classification and Regression Tree Classifiers on the Iris Data Set," *Journal of Southwest Jiaotong University*, vol. 58, no. 2, 2023. https://doi.org/10.35741/issn.0258-2724.58.2.59

[13] J. Josse, J. M. Chen, N. Prost, G. Varoquaux, and E. Scornet, "On the consistency of supervised learning with missing values," *Statistical Papers*, Sep. 2024. https://doi.org/10.1007/s00362-024-01550-4

[14] S. Chanmee and K. Kesorn, "Semantic decision Trees: A new learning system for the ID3-Based algorithm using a knowledge base," *Advanced Engineering Informatics*, vol. 58, p. 102156, Oct. 2023. https://doi.org/10.1016/j.aei.2023.102156

[15] A. Agarwal, K. Jain, and R. K. Yadav, "A mathematical model based on modified ID3 algorithm for healthcare diagnostics model," *International Journal of System Assurance Engineering and Management*, vol. 14, no. 6, pp. 2376–2386, Dec. 2023. https://doi.org/10.1007/s13198-023-02086-w.

[16] Y.-C. Chiang, Y.-C. Hsieh, L.-C. Lu, and S.-Y. Ou, "Prediction of Diagnosis-Related Groups for Appendectomy Patients Using C4.5 and Neural Network," *Healthcare*, vol. 11, no. 11, p. 1598, May 2023. https://doi.org/10.3390/healthcare11111598

[17] M. Ozcan and S. Peker, "A classification and regression tree algorithm for heart disease modeling and prediction," *Healthcare Analytics*, vol. 3, p. 100130, Nov. 2023. https://doi.org/10.1016/j.health.2022.100130

[18] A. R. Kadhim, R. S. Khudeyer, and M. Alabbas, "Facial Sentiment Analysis Using Convolutional Neural Network and Fuzzy Systems," *Informatica*, vol. 48, no. 12, Sep. 2024. https://doi.org/10.31449/inf.v48i12.6151

[19] C. Pal, S. Das, A. Akuli, S. K. Adhikari, and A. Dey, "Cocoa-Net: Performance Analysis on Classification of Cocoa Beans Using Structural Image Feature," *Informatica*, vol. 48, no. 12, Sep. 2024. https://doi.org/10.31449/inf.v48i12.5762

[20] X. Ying, "An Overview of Overfitting and its Solutions," *J Phys Conf Ser*, vol. 1168, no. 2, p. 022022, Feb. 2019. https://doi.org/10.1088/1742-6596/1168/2/022022

[21] A. Ilham, A. Kindarto, A. Kareem Oleiwi, and L. Khikmah, "CFCM-SMOTE: A Robust Fetal Health Classification to Improve Precision Modelling in Multi-Class Scenarios," *International Journal of Computing and Digital Systems*, vol. 15, no. 1, pp. 1–9, 2024. https://doi.org/10.12785/ijcds/160137

[22] A. Ilham, R. Satria Wahono, C. Supriyanto, and A. Wijaya, "U-control Chart Based Differential Evolution Clustering for Determining the Number of Cluster in k -Means," *International Journal of Intelligent Engineering and Systems*, vol. 12, no. 4, pp. 306–316, Aug. 2019. https://doi.org/10.22266/ijies2019.0831.28

[23] Cömert, Zafer and Şengür, Abdulkadir and Budak, Ümit and Kocamaz, Adnan Fatih, "Prediction of intrapartum fetal hypoxia considering feature selection algorithms and machine learning models," *Health Information Science and Systems*, vol. 12, no. 4, 2019. https://doi.org/10.1007/s13755-019-0079-z

Malicious iOS Apps Detection Through Multi-Criteria Decision-Making Approach

Arpita Jadhav Bhatt[1*], Neetu Sardana[2]
Department of Computer Science & Engineering and Information Technology, Jaypee Institute of Information Technology, India
E-mail: arpitajadhav@gmail.com, neetu.sardana@jiit.ac.in
[*]Corresponding author

Keywords: multi-criteria decision making (MCDM), ensemble learning, analytic hierarchy process, iOS app, privacy leaks

Received: September 2, 2024

In today's era, smartphones are used in daily lives because they are ubiquitous and can be customized by installing third-party apps. As a result, the menaces because of these apps, which are potentially risky for user's privacy, have increased. Information on smartphones is perhaps, more personal than compared to data stored on desktops or computers, making it an easy target for intruders. After Android, the most prevalently used mobile operating system is Apple's iOS. Both Android and iOS follow permission-based access control to protect user's privacy. However, the users are unaware whether the app is breaching the user's privacy. To combat this problem, in the paper we propose a hybrid approach to detect malicious iOS apps based on its permissions. In the first phase, weights have been assigned to app permissions using multi-criteria decision-making (MCDM) approach namely Analytic Hierarchy Process (AHP), and in the second phase machine learning& ensemble learning techniques have been employed to train the classifiers for detecting malicious apps. To test the efficacy of the proposed method dataset comprising 1150 apps from 12 app categories has been used. The results demonstrate the proposed approach improves the efficacy of detecting malicious iOS apps for majority of categories.

Povzetek: Raziskava predlaga hibridni pristop za zaznavanje zlonamernih iOS aplikacij z uporabo analitičnega hierarhičnega procesa za dodelitev uteži dovoljenjem aplikacij in strojnega učenja za klasifikacijo, kar izboljša zaznavanje.

1 Introduction

Apple's iOS (iPhone Operating System) is one of the most widely used mobile operating systems after its counterfeit Android. Apple manufactures and distributes different types of iOS devices such as iPad, iPod touch, iPhone, Apple Watch, Apple TV, etc. Apple has a huge customer base because it provides a lot of unique features such as multitasking, multi-touch gestures, internal accelerometers, voice assistant Siri, etc.(Wikipedia). Apple also offers a platform for its developers to publish and distribute their applications also called apps through its online store 'App Store'. This online store is also a repository of billions of apps from which iOS users can download apps from different categories (Apple Inc.). With the presence of 1.96 million apps, the App Store is the world's second-largest online store than its counterpart Android which has 2.87 million apps on its official store Play Store. Such a fast-growing platform motivates developers, IT industries, marketing firms, and organizations to develop feature-rich apps. It also allures the customers or users to download these apps.

With the upsurge of smart devices, the number of apps, and the number of smartphone users, smartphones and smart devices have also become a device for storing a large amount of user's personal data (Erickson et al.). This includes personal information such as address book, photo gallery, email IDs, passwords, calendar events, etc. Additionally, a smartphone always generates contextual data via its sensors. Such crucial information of users is undoubtedly more personal when compared with the data, which is stored on personal computers because smartphones stay with individuals throughout the day and generate a lot of contextual data by sensors. These sensors are not present on personal computers; therefore, it can be inferred that smartphones contain a lot of user's personal data which makes them a valuable resource for the developers of malicious apps who might intend to develop privacy-infringing apps and access user's data.

To preserve the privacy of its users both Apple and Android follow a permission-based access control policy (Krupp). The policy ensures that the app will notify its users about all the permissions and resources the app will use during its run-time. Android follows an install-time and run-time permission policy in which the users are aware of the permissions the app will acquire during its usage during app installation. Whereas, Apple follows a run-time permission policy in which the users are informed about app permissions during app usage (Khan et al.). Additionally, Apple provides privacy controls through its inbuilt 'setting app', through which they can explicitly define the permission for every app which has been installed (Krupp). Additionally, Apple follows a strict code signing process in which the apps that have been submitted by the developers on the App Store are critically examined before they are published. However, past attacks on iOS devices via privacy-infringing apps have demonstrated these methods adopted by Apple are inadequate to preserve the privacy of the users.

A lot of research work has been conducted to highlight the privacy breach by apps. Research conducted by Wired has identified that thousands of iOS and Android apps leak data from the cloud. The apps leaked lots of user's personal data such as medical information, phone number, device identifiers, passwords, etc.(WIRED). A similar kind of research conducted by Oxford researchers identified that a lot of third-party apps are sharing data with Facebook and Google. A total of 959,000 apps were analyzed. The study identified that 88% of these apps were transferring data to Alphabet which is Google's parent company, while Twitter, Facebook, Microsoft, and Amazon received 34%, 43%, and 18% of user's data respectively (Millman). To name a few the data collected by these companies included age, location, gender, etc. A report by Pt security highlighted the threats by the third-party mobile apps which included insecure data storage by apps, Escalated privileges taken by the app, side-loaded software, client-side vulnerabilities, etc.(Ptsecurity.com).

The issue of privacy leaks by apps also increases because of the issues in the current permission model adopted by Apple. The users do not have fine-grained access control over the data which is shared by the app in use. Smartphones do offer coarse-grained privacy as well as security controls where the users can either deny or allow permission for an explicit resource by an app. However, the problem with this approach is that once a user grants permission to the app he/she does not have any control over restricting the app from sharing the data. For example, once a user grants location permission to an app, the user cannot restrict the app from accessing a particular location. Likewise, once a user grants permission to access the address book, the user cannot restrict the app from accessing a specific contact. Since the users do not have the option to specify the accuracy of the sensors while accessing location, device accelerometer, and locally shared data, thus the apps must restrict their access. However, developers of malicious apps intentionally create over-privileged apps. Users are allured by the features of apps and thus grant permissions to the friendly-looking apps. However, the users never know if the app is using their data locally or sharing it with third-party domains without their consent. To eradicate this, problem we present a privacy detection model that uses app permissions during static analysis. The model first extracts user permission using the concept of reverse engineering and constructs a Boolean value permission matrix P_mxn where m represents the number of apps under analysis and n represents no. of permissions. Here a total of 1150 iOS apps from 12 app categories are tested for 10 different user permissions. Machine learning and ensemble-based techniques are employed to train the classifiers and determine malicious iOS apps. Apps are reverse-engineered from 12 different categories. In order to improve the precision of classifiers, the correlation of permissions for each category has been computed using Analytic Hierarchy Process (AHP). Later, the weighing factors obtained from AHP for each permission category-wise have been used to construct a weighted permission matrix P[mxn] to train the classifiers. The motivation for using a weighted permission matrix is that every permission has a different weight for a category. Apple provides a set of predefined app categories on the App Store which helps the developer to choose the best category before uploading the app. The category also defines the necessary features that the app provides the users during its usage. For example, the category navigation specifies that the apps belonging to the navigation category will fetch the user's location to guide them and navigate them. Likewise, an app belonging to the photo and video category will require access to the user's photo gallery and camera to serve its intended purpose. Apple provides a limited set of permissions, which require explicit approval during app usage. The problem of privacy leaks exists because it is very difficult to determine a benign app, an over-privileged app, or a malicious app with this limited permission set. Even the operating system cannot determine the intention of an app during its run-time. In other words, it is very difficult to identify malicious iOS apps as there exists a thin boundary line to identify how well a permission is correlated to a category. Using our approach, we identify the most significant permissions within a category using AHP approach that helps in identifying the malicious iOS apps. To address the challenges in the existing model for handling privacy breaches, we use the AHP technique to find the correlation of permission for a category to detect privacy violations by apps. For example, the permission of a user's location is an essential feature for an app belonging to the 'navigation' category because it functions after it receives the user's approval to access the GPS location. The same permission might have a different weight for the 'books' category as the prime purpose of this category is to provide stories, comics, graphic novels, and interactive content for which location permission may not be mandatory permission. Based on the aforementioned facts and guidelines provided by Apple we propose a heuristic approach to determine privacy leaks by iOS apps.

The foremost contributions of the paper have been listed below.

- A novel hybrid approach that integrates MCDM approach, AHP with Machine learning & ensemble learning techniques has been proposed to detect malicious iOS apps.

Table 1: Summary of research works on detecting privacy breach

Authors	Objective	Technique/Method	Data Set
Abdirashid et al. (Sahal et al.)	Develop permission-based malware detection model	Utilize feature selection techniques	1000 samples of apk files
Jin et al. (Sun et al.)	Develop malware detection system SigPID	Identify the most significant app permission to classify malware apps	310926 benign Android apps, 5494 malicious Android apps
Wang et al. (Wang et al.).	Explore the permission-induced risk to classify Android apps	Ranking of permissions, identification of malicious apps using PCA and SFS	310,926 benign and 4868 malicious apps
Jing et al. (Jing et al.)	Develop a risk assessment framework	Computes a risk assessment baseline by monitoring the run time behaviour of apps	14 Android apps
Adhikari et al. (Adhikari et al.)	Analyze run time behaviour of health care apps	Computation of safe score and risk score of iOS apps during their usage	20 iOS apps
Kang et al.(Kang et al.)	Malware detection using static analysis	Permission-based analysis of malware	51,179 benign Android apps and 4,554 malware Android apps
Huang et al.(Huang et al.)	Detect malicious Android apps based on their permissions	Grouping of permissions to Boolean vector and then training in machine learning classifiers	124,769 benign Android apps and 480 Malicious Android apps

- The proposed approach has been evaluated on 1150 apps belonging to twelve category iOS apps. Each app possesses ten permissions.
- The proposed approach improves the malware detection accuracy best case value of 14%.

The rest of the paper is organized as follows. Section 2 describes the related work on machine learning and ensemble techniques used to detect malicious apps. The section also describes the techniques that have been used in this paper. Section 3 describes our proposed heuristic approach of multicriteria decision-making approach using AHP. Section 4 describes the experimental results and analysis. The paper is concluded in Section 5.

2 Machine learning, ensembling techniques to detect privacy leaks based on app permissions

Machine learning and ensembling learning techniques have been employed by many researchers to detect privacy violations by apps. Ping et al. have developed an ensemble classifier ‘Enclamald’ to identify the contrasting permission patterns to illustrate the important difference between malicious apps the benign ones based on permission usage(Xiong et al.). Liu et al. have proposed a two-layered permission-based detection model for Android apps. In their work, they considered both requested permission and used permission by apps to detect malicious apps using machine learning techniques (Liu and Liu). Congyi et al. used an implementation of ensemble learning- XGBoost method to detect malicious Android apps based on permission usage(Congyi and Guangshun). Alba et al. used feature selection and ensembling techniques to classify Android malware(Coronado-De-Alba et al.). Idrees et al. proposed a model PIndoid, permissions, and intent-based framework, to detect malicious Android apps. It uses a combination of permissions and intents integrated with the ensemble method to improve the malware detection accuracy(Idrees et al.). Abdirashid et al. proposed a model to detect unknown malware by using a permission-based approach to enhance the accuracy as well as efficacy(Sahal et al.). The authors have improved the feature selection technique by incorporating weighing method TF-IDFCF, based on the class frequency of the app features. Jin et al. developed a malware detection system SigPID capable of coping with malware and its variants(Sun et al.). The authors have used three levels of pruning to mine app permission and identify the most significant permission capable of distinguishing malware apps from benign ones. Wang et al. have applied different ranking algorithms to classify malicious Android apps based on different ranking techniques namely principal component analysis(PCA) and sequential forward selection (SFS) to detect risky app permissions along with their subsets(Wang et al.). The authors have used a data set of 310926 benign and 4868 malicious apps and employed several machine learning classifiers. Jing Y et al. developed an automated risk assessment framework RiskMon which utilizes machine learning models to rank and assess the risks by Android apps(Jing et al.). The highlight of the tool was that it continuously monitors the behaviour of Android apps by combining the expectations of app users with their run-time behaviour. Run time behaviour of 20 iOS mobile health care iOS apps was analyzed by Adhikari R et al.to determine their strengths and weaknesses by assigning a safe score and risk score to them based on their run time analysis(Adhikari et al.).

Table 1 details the summary of research works on detecting privacy breaches.

However, the limitation of the above approaches is the differentiation of benign apps from malicious apps if they all request a similar set of app permissions. Since in Android, a large set of app permissions (approximately 320+) is already available, hence application of feature selection techniques, machine learning techniques, and reverse engineering is easy. However, in the case of the iOS platform a limited set of 10-13 permissions is available which varies with the iOS version. Hence, distinguishing a benign app from a malicious app is very difficult because there exists a thin boundary between a malicious and a benign app.

As most of the work has been done for the Android platform, in this paper we propose the detection of malicious apps for the iOS platform using AHP and MCDM approach to detect malicious iOS with a minimal permission set. We have also employed several ensembling techniques. The machine learning classifiers that were used for evaluating the proposed method are listed below(Mesevage), (Abaker and Saeed; Chehal et al.), (Harahsheh and Chen).

(i) **Naïve Bayes (NB):** It is a Bayesian classification method and is based on Bayes' Theorem. The Bayesian classification method builds a probabilistic classifier that is based on modelling the underlying features for different classes. The classification technique predicts class member probability that a given sample/tuple belongs to a particular class. The advantages of using this technique are that it needs less training data, is highly scalable, and can be used for both multi-class as well as binary classification problems(Mesevage).

(ii) **Decision Tree (DT):** It constructs a tree structure in a top-down recursive manner based on the divide and conquer manner. The decision tree is a tree structure where the internal node represents a test on an attribute every branch represents the output of the test and the leaf nodes depict the class distribution and are easy to interpret(Chehal et al.).

(iii) **Random Forest (RF):** Random Forest generates many classification trees. Every tree gives a classification and the forest selects the classification that has the most votes [15].

(iv) **Neural Network (NN):** The basic unit of a neural network is neurons, which take inputs, perform mathematical computations with them, and generate output. Every input is multiplied by a weight, and then all weighted inputs are added together with a bias function, and then the sum is passed through an activation function(Zhou).

(v) **Support Vector Machine (SVM):** It is a fast machine learning algorithm used for solving multiclass classification problems for larger data sets. It can work with high-dimensional data comprising thousands of features and attributes. The algorithm can be used in text classification problems with high-dimensional spaces [15].

The ensembling approaches that have been employed are bagging using J48 (decision tree) and boosting.

(vi) **Bagging (Bg):** The technique is based on creating multiple subsets from the original dataset. The instances from the dataset are selected with replacements. Then a base model also called a weak model is created for each of the subsets. The models are run in parallel and they work independently of each other. The final predictions are computed by combining the predictions from all models(Idrees et al.).

(vii) **Boosting (Bo):** In boosting a subset is created from the original dataset and instances are given equal weights initially. The base model is constructed on the previously created subset and is utilized to make predictions for the complete dataset. Errors are determined to employ real and anticipated values. Higher weights are allocated for the perceptions that are incorrectly anticipated. A strong learner is defined using the weighted mean of weak learners(Idrees et al.).

The following section describes the proposed approach to determine malicious iOS apps using a multi-criteria decision-making approach.

3 Classifying iOS Apps using proposed hybrid approach

Figure 1 shows the proposed framework to classify iOS apps using machine learning and ensemble techniques based on ranked permissions. Permissions are ranked using AHP. In the proposed method, the apps are installed from the AppStore, and their features are fetched (here features refer to the app permissions such as location, camera, photo gallery, etc.). We have considered ten features for twelve categories of iOS apps. Each app has a set of features in the form of a permission vector. Generally, permission for each feature is either present or absent corresponding to an app, and the features if present are considered to be equally important. In reality, the features of each category app have different weights. Based on this belief, we have ranked the permission set of each category of apps. On the basis of permission usage across the category, we have applied correlation coefficient and ranked permissions across the category. For example, an app belonging to the Social Networking category can have app permissions like photo, camera, location, internet, etc. whereas a simple flashlight app from the utilities category may require only camera permission. Thus, the ranking of permission for camera would be entirely different in social networking and utility category.

The proposed method has two phases. In the first phase, the app features are assigned weights based on the app category. In the second phase, the classification algorithms are applied for the identification of malicious iOS apps.

Assigning weights to app permissions

To rank the permissions of an app we are using the AHP, a MCDM approach. Step 1 is to identify the AHP Hierarchy. In this step, first, the goal is defined. In this work, the goal is to classify an iOS app as Malign or benign. Second, the criteria are identified. Here, the criterion is the apps category.

Figure 1: Framework to classify iOS App using machine learning and ensemble technique based on ranked permissions

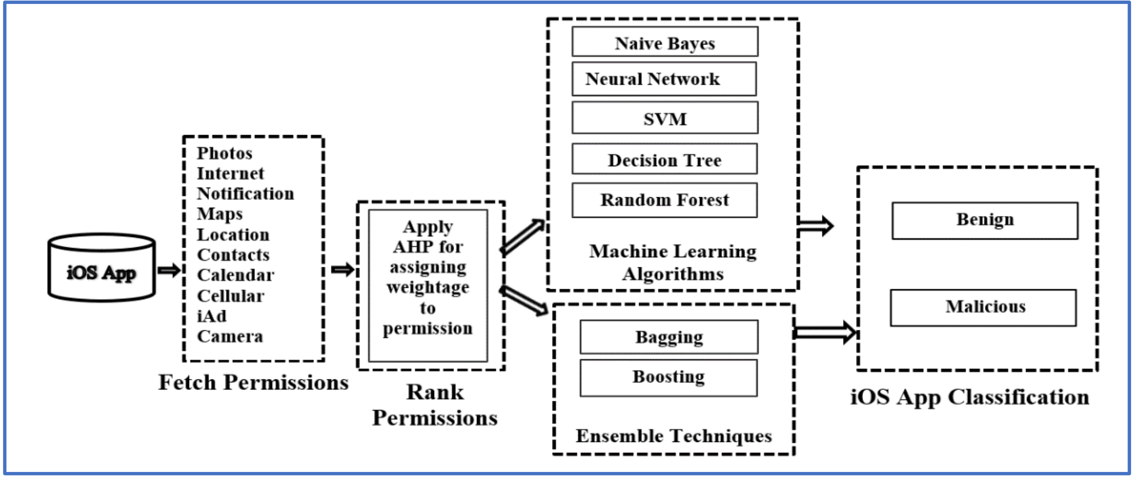
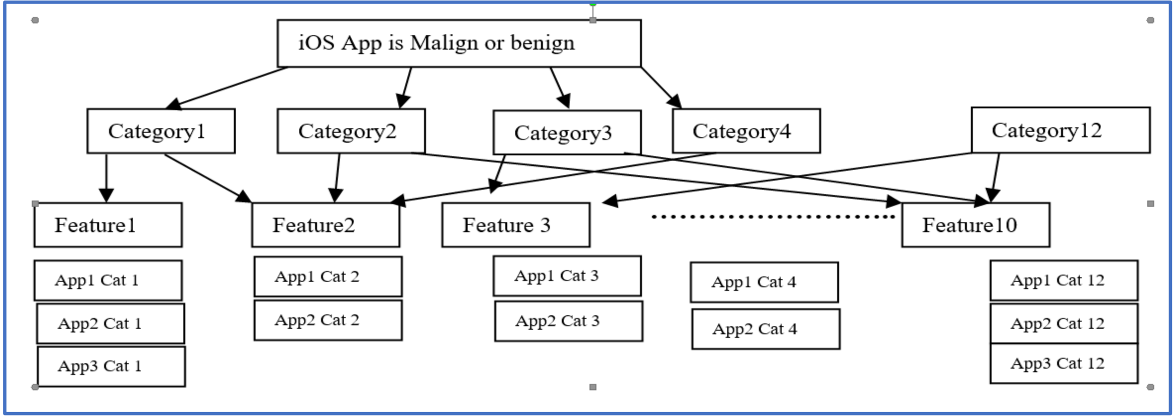


Figure 2: The Hierarchy of iOS Apps for AHP



Twelve mobile app categories have been considered. The categories are Books, Navigation, Games, Education, Health & Fitness, Lifestyle, Music, Utilities, Photo & Video, Sports, Social Networking, and Finance(Bhatt et al.). Third, the sub-criterion is defined if present. Here, the sub-criterion is the feature set of each category app for which permissions are present.AHP has been used to identify important features of each category app. The bottom level consists of the various alternatives. In our problem, 1150 apps belonging to twelve categories are the varied alternatives. The AHP hierarchy for ranking permissions of each category app is shown in Figure 1.

In step 2, pair-wise comparisons among features are performed to create a judgement matrix. We consider the ten feature sets of all apps belonging to each category. The features set consists of photos, internet, notification, map, location, contacts, calendar, cellular data, iAd, and camera. The Pearson correlation coefficient is utilized to find the feature importance for a particular category app. Then we used the Pearson correlation coefficient to perform a pair-wise comparison among features to find the relative importance of each pair and are given values in the range of 1 to 9. Table 2 has been used for the creation of a judgement matrix that provides the relative ranking that signifies the intensity of importance among the considered feature pair. The order of the judgement matrix depends upon the number of elements that the level of comparison. Since we are comparing the 10 feature pairs so for each category, the matrix of dimension 10*10 is formed. As the judgement matrices are formed, the eigenvectors and maximum eigenvalue (λmax) for each matrix are computed. Later consistency index (CI) and consistency ratio (CR) are calculated as shown in Algorithm 1. RI is a random consistency index given by Saaty for n varying from 1 to 10 (Refer Table 3). The acceptable value of CR is less than 0.1. If the CR value exceeds 0.1, it represents inconsistencies and the result is meaningless. Thus, the entire process requires re-evaluations(Saaty).

Table 2: Criteria for comparison

Intensity of Importance	Definition
1	Equal importance
3	Weak importance of C_i over C_j
5	Essential or strong importance
7	Demonstrated importance
9	Absolute importance
2,4,6,8	Intermediate
Reciprocals	If C_i has one of the above judgements assigned to it when compared with C_j has the reciprocal value when compared with C_i

Table 3: RI values

Size	1	2	3	4	5	6	7	8	9	10
RI	0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

Step1: Define AHP hierarchy. (Refer Figure 2)

1. Goal: To identify an iOS app is benign or malicious
2. Criteria: Twelve iOS App Categories
3. SubCriteria: Feature vectors of each category app.
4. Alternatives: 1150 apps belonging to different categories.

Step 2: Pairwise comparison is performed to find relative ranking among features of each category apps. (Refer Table 1)

Step 3: Compute Judgement Matrix (M) of dimension 10*10 for each category apps. (Illustrated in Table 5)

Step 4: Calculate M_n, normalized judgement matrix which can be obtained by dividing each element with the column sum.

Step 5: Find the average of all the row elements of $M_n$to get eigenvectors W^T having dimension 10*1 that are considered as the weights of each feature if the matrix is consistent.

Step 6: Check the consistency of the matrix, M.

a. The maximum eigen value (λ_{max}) for each matrix is calculated.

$$\lambda max = \frac{1}{n}\sum_{i=1}^{n\cdot}\frac{\text{ith entry in } MW^T}{\text{ith entry in } W^T}$$

b. The consistency index (CI) of each matrix of order 10 is calculated

$$CI = \frac{(\lambda max - n)}{(n-1)}\text{Here n =10 (number of app features)}$$

c. Compute consistency ratio (CR)

$$CR = \frac{CI}{RI}\ Here, RI = 1.49$$

Algorithm 1 : Ranking permission of varied category Apps using AHP

The judgement matrix is found to be consistent if the value of CR is less than 0.1. The weights of features are fetched.

Apply classification algorithms to identify the benign and malign iOS apps

The feature weights computed by applying AHP are used for each category of apps and are used to train the classifiers. Five machine learning and two ensemble learning techniques are applied for the identification of malign and benign apps. The machine learning techniques that are considered are Naïve Bayes, Neural Network, Random Forest, Decision Tree, and SVM.

The ensemble learning techniques are applied as the feature set in some of the categories is found to be skewed. The techniques used are bagging and boosting where J48 is used as a baseline technique.

4 Experimental results and analysis

This section presents the experimental setup and results of the proposed model on 12 categories of iOS 1150 apps.

4.1 Experimental setup

The proposed permission weighting approach is evaluated on twelve categories of iOS apps. The categories are Books (Bks), Education (Edu), Finance (Fin), Games (Gam), Health & Fitness(H&F), Lifestyle (LS), Music (Mus), Navigation (Nav), Photo&Video(P&V), Social Networking (SN), Sports (Sp) and Utilities (Util). There are a total of 1150 iOS apps that have been considered. The apps are fetched from the App Store. Table 5 shows the distribution of apps in each category that has been considered.

Table 4: Category wise apps distribution

Category	**Apps Count**
Books	126
Education	141
Finance	91
Games	201
Health & Fitness	81
Lifestyle	101
Music	91
Navigation	46
Photo & Video	61
Social Networking	91
Sports	46
Utilities	86
Total	**1150**

Each category app defines the important features that it serves to the users. The features set consists of photos, internet, notification, map, location, contacts, calendar, cellular, iAd, and camera. The permission log of these features has been extracted. Each app has its own set of feature vectors, for example, App1 of the Books category has feature vector fv= (1,0,0,1,0,1,0,0,0,0) which depict that this app has three features: photos, map, and contacts. The different features of apps are generally given equal importance. Here photos, map, and contacts are given equal importance. It has been noticed that certain features of apps are more important than other features. Based on this belief, this paper prioritizes the features of an app using the AHP, MCDM technique to classify the app as malicious or benign.

The proposed method undergoes two phases to detect malicious iOS apps: In the first phase the features of each category of iOS apps are ranked using AHP and in the second phase machine learning as well as ensemble learning methods are applied to ranked features. Finally, the performance of both approaches is evaluated using precision defined by equation 1. Here, precision is a metric used to evaluate the model's positive classifications which are actually positive. It is defined as a ratio of true positive (TP) predictions and total number of predicted positives which includes both false positives (FP) as well as true positives. Precision improves when false positives decrease.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (1)$$

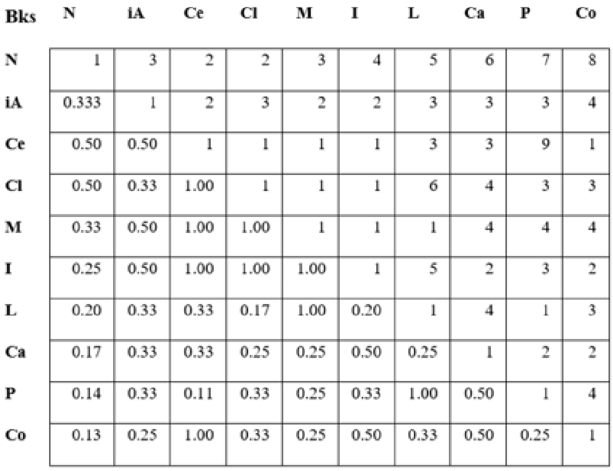
4.2 Experimental results

This section presents the results obtained from our experimental study. During the first phase, the features of varied category apps were ranked using AHP. The feature's importance has been computed using the correlation coefficient. The results of which are given in Table 4 and have been taken from our previous work on the detection of malicious iOS apps using static and dynamic analysis approaches (Bhatt et al.). The work is an extension of previous work by including ensembling and multi-criteria decision-making approach. The correlation values are used for pair-wise feature comparison. The judgement matrix has been computed for each category app using pairwise feature comparison. Figures 3a-3l represent the Judgement Matrix corresponding to each category of apps.

Table 5: Features Importance in each category apps

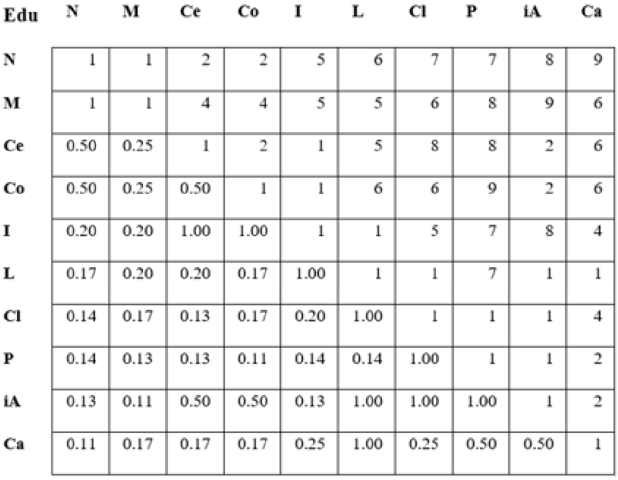
Bks	**Edu**	**Fin**	**Gam**
iAd	Notifi.	Notifi.	iAd
Photos	MapKit	Location	Notifi.
Internet	Cellular	MapKit	Cellular
Notifi.	Location	Camera	Calendar
Contacts	Camera	Contacts	Photos
MapKit	Calendar	Cellular	Location
Calendar	Contacts	Photos	Internet
Cellular	Internet	Calendar	Camera
Location	Photos	iAd	MapKit
Camera	iAd	Internet	Contacts
H&F	**LS**	**P&V**	**SN**
MapKit	Notifi.	Notifi.	Notifi.
Notifi.	MapKit	Camera	Cellular
Contacts	Cellular	iAd	Camera
iAd	Contacts	MapKit	Calendar
Calendar	Location	Calendar	Location
Camera	iAd	Cellular	iAd
Photos	Calendar	Internet	Contacts
Cellular	Photos	Location	MapKit
Location	Camera	Photos	Photos
Internet	Internet	Contacts	Internet

Spo	Util	Mus	Nav
Cellular	MapKit	Cellular	Cellular
Calendar	Cellular	Notifi.	Calendar
MapKit	Calendar	MapKit	Notifi.
Notifi.	Camera	Calendar	Internet
Camera	Location	Internet	Photos
Photos	Internet	Location	iAd
Contacts	Photos	Photos	MapKit
Location	Contacts	Camera	Camera
Internet	Notifi.	iAd	Contacts
iAd	iAd	Contacts	Location



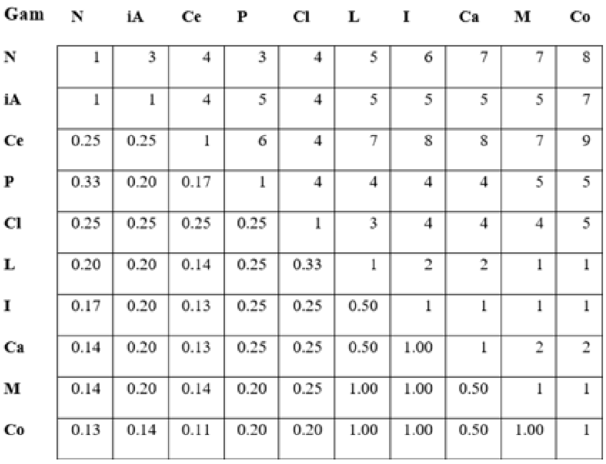
Bks	N	iA	Ce	Cl	M	I	L	Ca	P	Co
N	1	3	2	2	3	4	5	6	7	8
iA	0.333	1	2	3	2	2	3	3	3	4
Ce	0.50	0.50	1	1	1	1	3	3	9	1
Cl	0.50	0.33	1.00	1	1	1	6	4	3	3
M	0.33	0.50	1.00	1.00	1	1	1	4	4	4
I	0.25	0.50	1.00	1.00	1.00	1	5	2	3	2
L	0.20	0.33	0.33	0.17	1.00	0.20	1	4	1	3
Ca	0.17	0.33	0.33	0.25	0.25	0.50	0.25	1	2	2
P	0.14	0.33	0.11	0.33	0.25	0.33	1.00	0.50	1	4
Co	0.13	0.25	1.00	0.33	0.25	0.50	0.33	0.50	0.25	1

Figure 3 a: Judgement Matrix for Apps from (Books (Bks) category



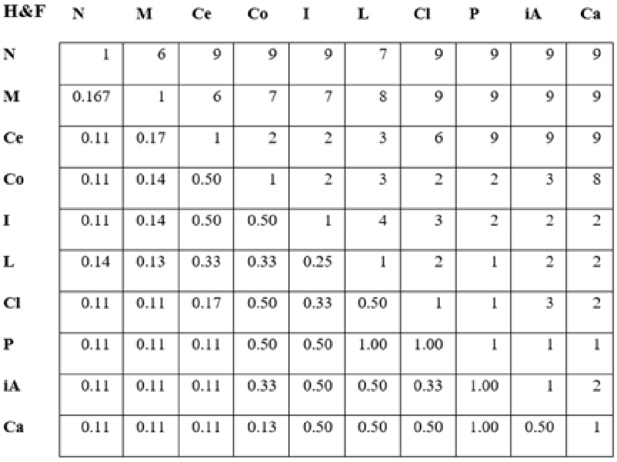
Edu	N	M	Ce	Co	I	L	Cl	P	iA	Ca
N	1	1	2	2	5	6	7	7	8	9
M	1	1	4	4	5	5	6	8	9	6
Ce	0.50	0.25	1	2	1	5	8	8	2	6
Co	0.50	0.25	0.50	1	1	6	6	9	2	6
I	0.20	0.20	1.00	1.00	1	1	5	7	8	4
L	0.17	0.20	0.20	0.17	1.00	1	1	7	1	1
Cl	0.14	0.17	0.13	0.17	0.20	1.00	1	1	1	4
P	0.14	0.13	0.13	0.11	0.14	0.14	1.00	1	1	2
iA	0.13	0.11	0.50	0.50	0.13	1.00	1.00	1.00	1	2
Ca	0.11	0.17	0.17	0.17	0.25	1.00	0.25	0.50	0.50	1

Figure 3 b: Judgement Matrix for Apps from Education (Edu) category



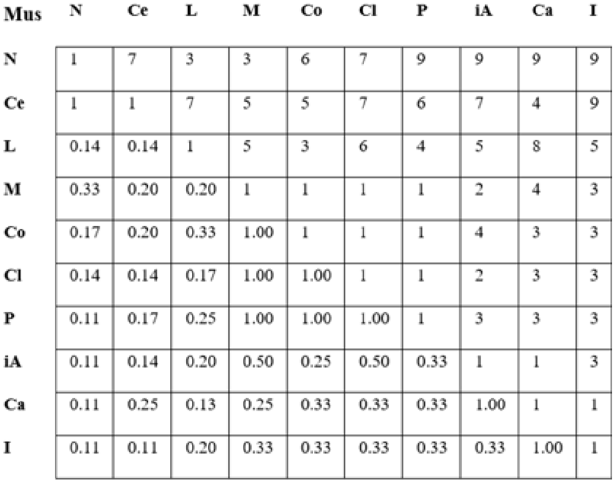
Gam	N	iA	Ce	P	Cl	L	I	Ca	M	Co
N	1	3	4	3	4	5	6	7	7	8
iA	1	1	4	5	4	5	5	5	5	7
Ce	0.25	0.25	1	6	4	7	8	8	7	9
P	0.33	0.20	0.17	1	4	4	4	4	5	5
Cl	0.25	0.25	0.25	0.25	1	3	4	4	4	5
L	0.20	0.20	0.14	0.25	0.33	1	2	2	1	1
I	0.17	0.20	0.13	0.25	0.25	0.50	1	1	1	1
Ca	0.14	0.20	0.13	0.25	0.25	0.50	1.00	1	2	2
M	0.14	0.20	0.14	0.20	0.25	1.00	1.00	0.50	1	1
Co	0.13	0.14	0.11	0.20	0.20	1.00	1.00	0.50	1.00	1

Figure 3 c: Judgement Matrix for Apps from Games (Gam) category



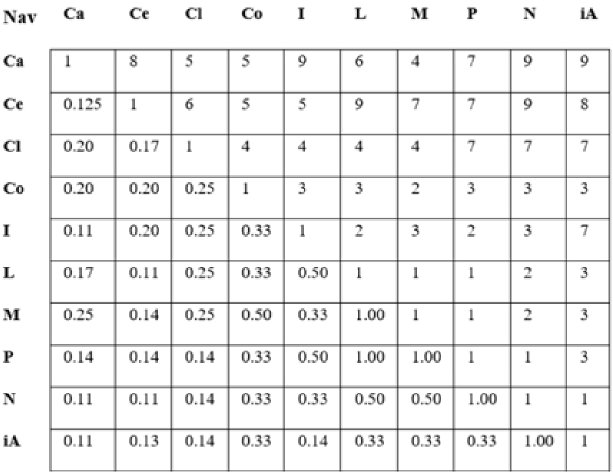
H&F	N	M	Ce	Co	I	L	Cl	P	iA	Ca
N	1	6	9	9	9	7	9	9	9	9
M	0.167	1	6	7	7	8	9	9	9	9
Ce	0.11	0.17	1	2	2	3	6	9	9	9
Co	0.11	0.14	0.50	1	2	3	2	2	3	8
I	0.11	0.14	0.50	0.50	1	4	3	2	2	2
L	0.14	0.13	0.33	0.33	0.25	1	2	1	2	2
Cl	0.11	0.11	0.17	0.50	0.33	0.50	1	1	3	2
P	0.11	0.11	0.11	0.50	0.50	1.00	1.00	1	1	1
iA	0.11	0.11	0.11	0.33	0.50	0.50	0.33	1.00	1	2
Ca	0.11	0.11	0.11	0.13	0.50	0.50	0.50	1.00	0.50	1

Figure 3 d: Judgement Matrix for Apps from Health & Fitness (H&F) category



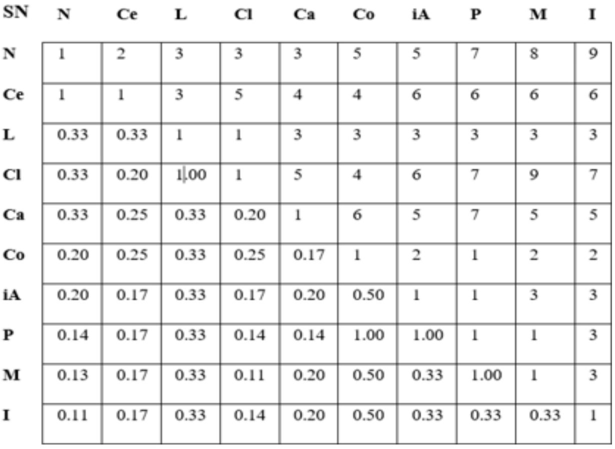
Mus	N	Ce	L	M	Co	Cl	P	iA	Ca	I
N	1	7	3	3	6	7	9	9	9	9
Ce	1	1	7	5	5	7	6	7	4	9
L	0.14	0.14	1	5	3	6	4	5	8	5
M	0.33	0.20	0.20	1	1	1	1	2	4	3
Co	0.17	0.20	0.33	1.00	1	1	1	4	3	3
Cl	0.14	0.14	0.17	1.00	1.00	1	1	2	3	3
P	0.11	0.17	0.25	1.00	1.00	1.00	1	3	3	3
iA	0.11	0.14	0.20	0.50	0.25	0.50	0.33	1	1	3
Ca	0.11	0.25	0.13	0.25	0.33	0.33	0.33	1.00	1	1
I	0.11	0.11	0.20	0.33	0.33	0.33	0.33	0.33	1.00	1

Figure 3 e:Judgement Matrix for Apps from Music (Mus) category



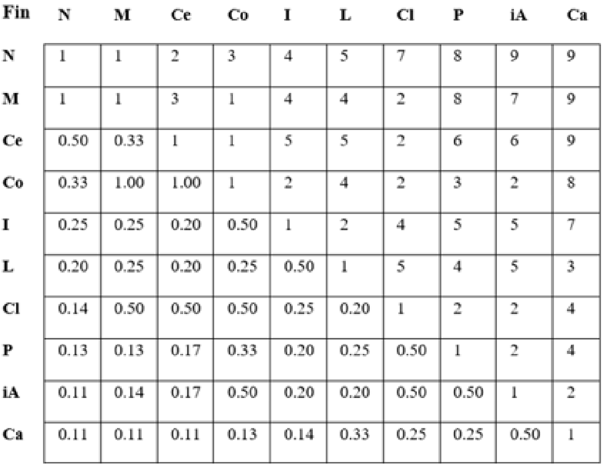
Nav	Ca	Ce	Cl	Co	I	L	M	P	N	iA
Ca	1	8	5	5	9	6	4	7	9	9
Ce	0.125	1	6	5	5	9	7	7	9	8
Cl	0.20	0.17	1	4	4	4	4	7	7	7
Co	0.20	0.20	0.25	1	3	3	2	3	3	3
I	0.11	0.20	0.25	0.33	1	2	3	2	3	7
L	0.17	0.11	0.25	0.33	0.50	1	1	1	2	3
M	0.25	0.14	0.25	0.50	0.33	1.00	1	1	2	3
P	0.14	0.14	0.14	0.33	0.50	1.00	1.00	1	1	3
N	0.11	0.11	0.14	0.33	0.33	0.50	0.50	1.00	1	1
iA	0.11	0.13	0.14	0.33	0.14	0.33	0.33	0.33	1.00	1

Figure 3 f:Judgement Matrix for Apps from Navigation (Nav) category



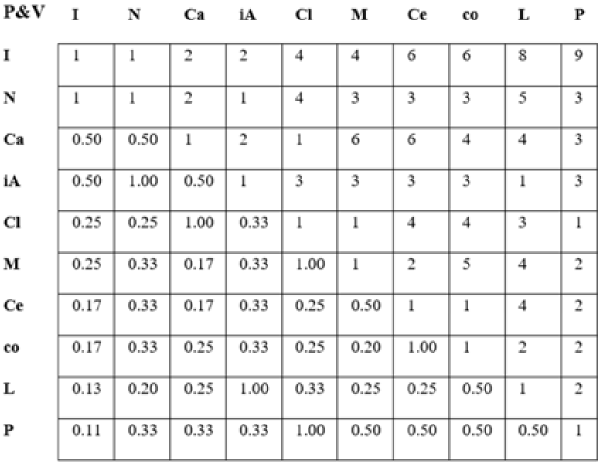
SN	N	Ce	L	Cl	Ca	Co	iA	P	M	I
N	1	2	3	3	3	5	5	7	8	9
Ce	1	1	3	5	4	4	6	6	6	6
L	0.33	0.33	1	1	3	3	3	3	3	3
Cl	0.33	0.20	1.00	1	5	4	6	7	9	7
Ca	0.33	0.25	0.33	0.20	1	6	5	7	5	5
Co	0.20	0.25	0.33	0.25	0.17	1	2	1	2	2
iA	0.20	0.17	0.33	0.17	0.20	0.50	1	1	3	3
P	0.14	0.17	0.33	0.14	0.14	1.00	1.00	1	1	3
M	0.13	0.17	0.33	0.11	0.20	0.50	0.33	1.00	1	3
I	0.11	0.17	0.33	0.14	0.20	0.50	0.33	0.33	0.33	1

Figure 3 g: Judgement Matrix for Apps from Social Networking (SN) category



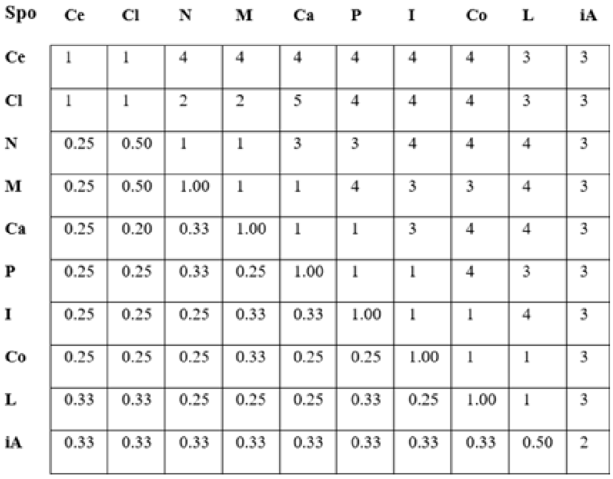
Fin	N	M	Ce	Co	I	L	Cl	P	iA	Ca
N	1	1	2	3	4	5	7	8	9	9
M	1	1	3	1	4	4	2	8	7	9
Ce	0.50	0.33	1	1	5	5	2	6	6	9
Co	0.33	1.00	1.00	1	2	4	2	3	2	8
I	0.25	0.25	0.20	0.50	1	2	4	5	5	7
L	0.20	0.25	0.20	0.25	0.50	1	5	4	5	3
Cl	0.14	0.50	0.50	0.50	0.25	0.20	1	2	2	4
P	0.13	0.13	0.17	0.33	0.20	0.25	0.50	1	2	4
iA	0.11	0.14	0.17	0.50	0.20	0.20	0.50	0.50	1	2
Ca	0.11	0.11	0.11	0.13	0.14	0.33	0.25	0.25	0.50	1

Figure 3 h: Judgement Matrix for Apps from Finance (Fin)category



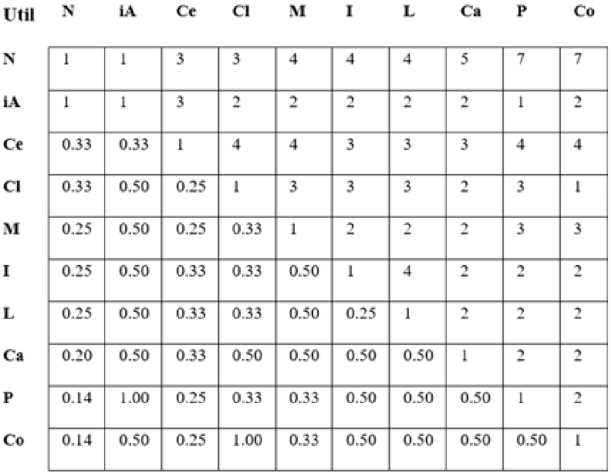
P&V	I	N	Ca	iA	Cl	M	Ce	co	L	P
I	1	1	2	2	4	4	6	6	8	9
N	1	1	2	1	4	3	3	3	5	3
Ca	0.50	0.50	1	2	1	6	6	4	4	3
iA	0.50	1.00	0.50	1	3	3	3	3	1	3
Cl	0.25	0.25	1.00	0.33	1	1	4	4	3	1
M	0.25	0.33	0.17	0.33	1.00	1	2	5	4	2
Ce	0.17	0.33	0.17	0.33	0.25	0.50	1	1	4	2
co	0.17	0.33	0.25	0.33	0.25	0.20	1.00	1	2	2
L	0.13	0.20	0.25	1.00	0.33	0.25	0.25	0.50	1	2
P	0.11	0.33	0.33	0.33	1.00	0.50	0.50	0.50	0.50	1

Figure 3 i: Judgement Matrix for Apps from Photo & Video (P&V) category



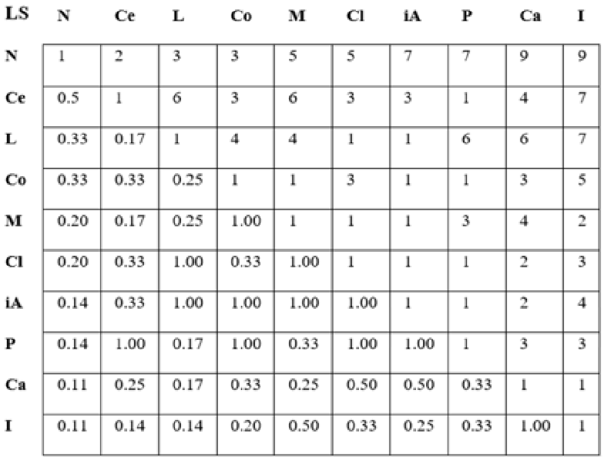
Spo	Ce	Cl	N	M	Ca	P	I	Co	L	iA
Ce	1	1	4	4	4	4	4	4	3	3
Cl	1	1	2	2	5	4	4	4	3	3
N	0.25	0.50	1	1	3	3	4	4	4	3
M	0.25	0.50	1.00	1	1	4	3	3	4	3
Ca	0.25	0.20	0.33	1.00	1	1	3	4	4	3
P	0.25	0.25	0.33	0.25	1.00	1	1	4	3	3
I	0.25	0.25	0.25	0.33	0.33	1.00	1	1	4	3
Co	0.25	0.25	0.25	0.33	0.25	0.25	1.00	1	1	3
L	0.33	0.33	0.25	0.25	0.25	0.33	0.25	1.00	1	3
iA	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.50	2

Figure 3 j: Judgement Matrix for Apps from Sports (Spo)category



Util	N	iA	Ce	Cl	M	I	L	Ca	P	Co
N	1	1	3	3	4	4	4	5	7	7
iA	1	1	3	2	2	2	2	2	1	2
Ce	0.33	0.33	1	4	4	3	3	3	4	4
Cl	0.33	0.50	0.25	1	3	3	3	2	3	1
M	0.25	0.50	0.25	0.33	1	2	2	2	3	3
I	0.25	0.50	0.33	0.33	0.50	1	4	2	2	2
L	0.25	0.50	0.33	0.33	0.50	0.25	1	2	2	2
Ca	0.20	0.50	0.33	0.50	0.50	0.50	0.50	1	2	2
P	0.14	1.00	0.25	0.33	0.33	0.50	0.50	0.50	1	2
Co	0.14	0.50	0.25	1.00	0.33	0.50	0.50	0.50	0.50	1

Figure 3 k: Judgement Matrix for Apps from Utilities (Util)category



LS	N	Ce	L	Co	M	Cl	iA	P	Ca	I
N	1	2	3	3	5	5	7	7	9	9
Ce	0.5	1	6	3	6	3	3	1	4	7
L	0.33	0.17	1	4	4	1	1	6	6	7
Co	0.33	0.33	0.25	1	1	3	1	1	3	5
M	0.20	0.17	0.25	1.00	1	1	1	3	4	2
Cl	0.20	0.33	1.00	0.33	1.00	1	1	1	2	3
iA	0.14	0.33	1.00	1.00	1.00	1.00	1	1	2	4
P	0.14	1.00	0.17	1.00	0.33	1.00	1.00	1	3	3
Ca	0.11	0.25	0.17	0.33	0.25	0.50	0.50	0.33	1	1
I	0.11	0.14	0.14	0.20	0.50	0.33	0.25	0.33	1.00	1

Figure 3 l: Judgement Matrix for Apps from Lifestyle (LS) category

The procedure given in Algorithm 1 has been followed and the final weights of the ten features for all the twelve category Apps are shown in Table 6. Table 7 shows the Consistency Index (CI) and inconsistency ratio (CR) values obtained for each iOS category app. A CR value below 0.1 suggests that the comparisons made are consistent and that the judgments used in the AHP process are reliable. In our analysis, it can be clearly observed from the tables 6 & 7 that the CR values obtained were consistently below this threshold for all category iOS apps, which means that the pairwise comparisons of the permissions were logical and coherent.

Table 6: Final weights of all features for twelve category apps

apps**Features**	**Weights (Bks)**	**Features**	**Weights (Edu)**	**Features**	**Weights (Gam)**	**Features**	**Weights (H&F)**
N	0.260416	N	0.22869837	N	0.258617391	N	0.373378406
iA	0.1536197	M	0.25874097	iA	0.217896213	M	0.239753234
Ce	0.1118894	Ce	0.13251323	Ce	0.196327424	Ce	0.127111058
Cl	0.1130392	Co	0.11850109	P	0.108141669	Co	0.070634637
M	0.0972874	I	0.1023565	Cl	0.080053713	I	0.054553629
I	0.0941756	L	0.0464333	L	0.034984294	L	0.03462572
L	0.0570132	Cl	0.03384061	I	0.025581562	Cl	0.030962473
Ca	0.0387621	P	0.02371545	Ca	0.030363325	P	0.025841841
P	0.0395886	iA	0.03329195	M	0.025065278	iA	0.023556455
Co	0.0342088	Ca	0.02190853	Co	0.02296913	Ca	0.019582547
Features	**Weights (Mus)**	**Features**	**Weights (Nav)**	**Features**	**Weights (SN)**	**Features**	**Weights (Spo)**
N	0.3164822	Ca	0.32905485	N	0.243144501	Ce	0.226412065
Ce	0.2583536	Ce	0.2360261	Ce	0.230117625	Cl	0.193037193
L	0.1466667	Cl	0.14566162	L	0.098318076	N	0.130405534
M	0.0571527	Co	0.07414116	Cl	0.158169247	M	0.114890071
Co	0.0561844	I	0.06437649	Ca	0.110922948	Ca	0.089492707
Cl	0.0476753	L	0.03690112	Co	0.040755572	P	0.070530109
P	0.0505276	M	0.04092982	iA	0.03808108	I	0.058589306
iA	0.0271474	P	0.03279818	P	0.032758226	Co	0.043199994
Ca	0.0210871	N	0.0221352	M	0.028127397	L	0.039473318
I	0.018723	iA	0.01797546	I	0.019605329	iA	0.033969703
Features	**Weights (Fin)**	**Features**	**Weights (Util)**	**Features**	**Weights (P&V)**	**Features**	**Weights (LS)**
N	0.2430282	N	0.24600855	I	0.239110134	N	0.286591226
M	0.200416	iA	0.14601401	N	0.169841704	Ce	0.198462154
Ce	0.151552	Ce	0.15944265	Ca	0.154772531	L	0.141693949
Co	0.1195261	Cl	0.10371867	iA	0.119366316	Co	0.079178536
I	0.0901974	M	0.08086139	Cl	0.081263199	M	0.064995197
L	0.0727182	I	0.07471276	M	0.074714065	Cl	0.056637209
Cl	0.0504631	L	0.055601	Ce	0.047334716	iA	0.061643922
P	0.0312343	Ca	0.04967195	co	0.040725409	P	0.064021004
iA	0.0256206	P	0.04642525	L	0.037987177	Ca	0.025483038
Ca	0.015244	Co	0.03791838	P	0.03488475	I	0.021293766

Table 7: Consistency Index (CI) and inconsistency ratio (CR) values

Sno	Category	CI	CR
1	Books	0.14527	0.097497
2	Education	0.146455	0.098292
3	Finance	0.136704	0.091747
4	Games	0.145712	0.097794
5	Health & Fitness	0.135315	0.090816
6	Lifestyle	0.144084	0.096701
7	Music	0.14871	0.099806
8	Navigation	0.143824	0.096526
9	Photo & Video	0.1444	0.096913
10	Social Networking	0.142957	0.095944
11	Sports	0.135897	0.091206
12	Utilities	0.147348	0.098891

A detailed illustration of AHP steps has been omitted for the sake of the length of the paper. The results of this evaluation were analyzed to find whether the inclusion of AHP in prioritizing and determining the weights of features improves the accuracy of iOS app classification or not. We have used the cross-validation technique in the Weka toolkit to measure the efficiency of the models. Generally, whenever an inadequate amount of data instances is available, cross-validation method is preferred to accomplish an unbiased approximation of the model performance. In the k-fold cross-validation technique, the dataset is divided into k subsets, each of equal size. The model is constructed 'k' times, each time using (k−1) sets of data instances for training the classifier and leaving out one subset as a 'test set' for predictions. We considered five machine-learning techniques and two ensemble-based techniques for classification. The machine learning classifiers that are used for evaluating the proposed method are Decision Tree (DT), Random Forest (RF), Naïve Bayes (NB), Neural Network (NN), and Support Vector Machine (SVM) and the considered ensemble approaches are bagging using J48(Bg) and Boosting (Bo). We compared the proposed AHP-based weighing approach with actual permission-based classification approaches.

The summary of the precision values for different classifiers has been depicted in Table 8 and Table 9. The tables also depict the comparison of precision values before/after applying the AHP technique for various classifiers. The results depicted in Table 8 and Table 9 demonstrate that the proposed AHP-based approach achieved an improved average accuracy in all the category apps. The improved average accuracy attained for classification algorithms Random Forest, Support Vector Machine, Naïve Bayes, Neural Network, and Decision tree is 77.83%, 78.61%, 77.99%, 75.21%, and 78.21% respectively. It has been observed that Random Forest and SVM-based AHP classifier (SVM_{AHP}), performs better in 8 categories out of 12 categories apps, and Naïve Bayes-based AHP (NB_{AHP}) and Neural Network-based AHP (NB_{AHP}) classifier performs better in 9 categories out of 12 categories apps. Integration of machine learning with AHP has shown the best performance for Health & fitness category apps as the improvement can be clearly observed in the case of three classifiers SVM_{AHP}, NB_{AHP}, and NN_{AHP} as 9.8%, 9.1%, and 8.3%. The proposed hybrid model has also shown good results for the apps belonging to the categories: Navigation and Photo & Video. The results reveal the improvement of 4.4%, 2.2%, and 2.9% in SVM_{AHP}, NB_{AHP}, and NN_{AHP} classifiers for the Navigation category and 1.9%, 3.3% and 4.9% in RFAHP, SVM_{AHP} and NB_{AHP} for Photo & Video category. The average accuracy attained in ensemble techniques, Boosting and Bagging using J48 is 80.01% and 77.22%. The ensemble learning technique, boosting integrated with AHP performed the best as it has shown better accuracy in all the 12 categories of apps. The highest improvement attained is 14% for Health and Fitness Apps. Figure 4 shows the improved precision scores for 12 iOS apps categories using the proposed hybrid approach.

Table 8: Summary of Results Precision Values (in Percentage)

Category	RF	RF_{AHP}	SVM	SVM_{AHP}	NB	NB_{AHP}	NN	NN_{AHP}	DT	DT_{AHP}
Books	76.1	76	75.3	76.3	76.8	75.9	71.8	75.3	79.9	79.9
Education	76.5	75.7	76.5	69	75.7	74.3	78.6	70.7	78.6	77.9
Finance	62.2	61.6	63.9	63.9	67.9	73.3	66	61.3	71	67.2
Games	85.1	86	86.2	85.7	84.1	84.6	84.2	84.5	83.5	82.5
Health & Fitness	83.7	82.5	74.9	84.7	71.7	80.8	77.9	86.2	84.1	78.5
Lifestyle	79.1	80	82.9	76	74.9	74.9	75.8	74.8	82.9	81.9
Music	97.8	98.9	98.9	98.9	95.6	95.6	95.6	95.6	98.6	98.6

Navigation	67.2	67.2	71.6	76	76	78.7	60.4	63.3	67.2	67.2
Photo & Video	75.9	77.8	68	71.3	67.3	72.2	77.4	77.4	74.1	74.1
Social Networking	79.7	81	82	81.9	75.2	74.2	76.5	76.5	78.5	81.3
Sports	66.1	67.9	74.7	74.7	69.9	69.9	58.4	58.4	70.8	74.8
Utilities	77.6	79.4	83.1	84.9	80.8	81.5	78.5	78.5	71.1	77.1
Average	77.25	**77.83**	78.17	**78.61**	76.33	**77.99**	75.09	**75.21**	78.36	**78.42**

Table9: Evaluation of weighing based approach using machine learning classification algorithm

S No	Category	Bo	BoAHP	Bg	BgAHP
1	Books(B)	78.5	79.3	79.2	76
2	Education(E)	70	73	76.4	78.6
3	Finance(F)	73.5	73.6	67.2	66.9
4	Games(G)	84	84	84.7	83.5
5	Health & Fitness (HF)	70	83.9	80	84.3
6	Lifestyle(L)	72.9	72.9	78.9	76.8
7	Music(M)	97.8	98.9	98.9	98.9
8	Navigation(N)	78.7	78.7	55.6	53.8
9	Photo & Video (PV)	77.8	79.6	70.4	70.4
10	Social Networking (SN)	79.6	79.7	82	82
11	Sports(S)	72.2	72.2	75.2	75.2
12	Utilities(U)	85.4	85.4	80.8	80.2
	Average	**78.37**	**80.1**	**77.44**	**77.22**

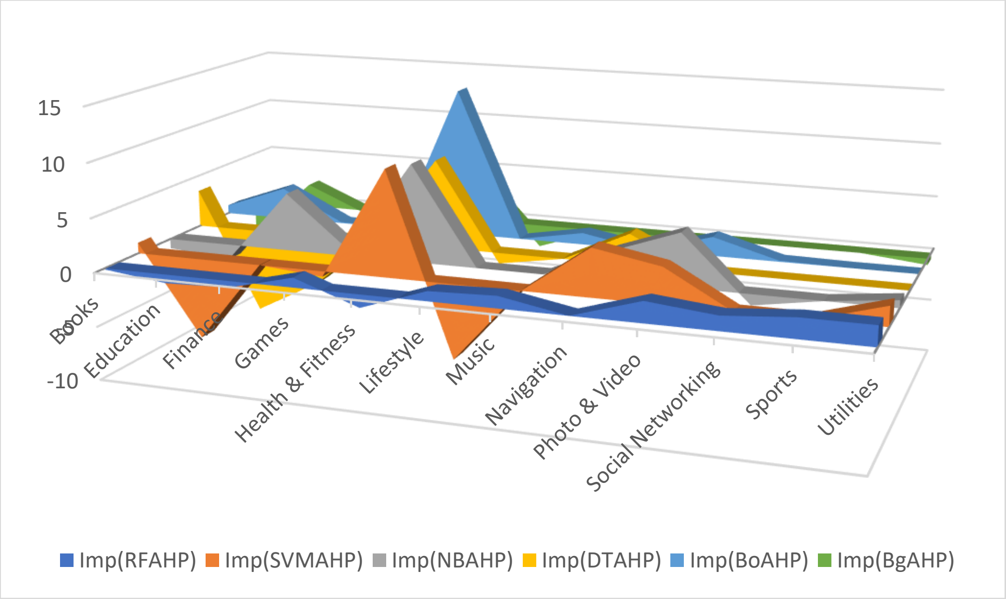


Figure 4: Improved precision using proposed hybrid approach for 12 iOS category apps

Based on the above results from Table 8 and Table 9 it can be concluded that the proposed approach of using a multi-criteria decision-making approach using AHP improves the detection rate of malicious apps. Up to 9.8% in machine learning techniques and 14% in ensemble learning techniques.

5 Conclusion

The paper proposes an AHP-based weighting approach integrated with machine learning and ensemble learning techniques to detect iOS malicious apps. The proposed method initially extracts the app permissions using static analysis for 12 categories to compute a permission matrix comprising the number of apps and presence/absence of features. Then correlation of permissions for every category is computed using Pearson Correlation. Later, the AHP technique is applied to determine the weights of all permissions based on their correlation with respect to the category and in order to compute a weighted permission matrix. The proposed method has been compared with traditional permission-based classification methods. Empirical results depict that the proposed approach improves the detection rate for all 12 categories of iOS apps. In the future, we plan to conduct a sensitivity analysis to test the robustness of the AHP-derived weights. We will also explore different privacy settings for iOS apps namely track, link, and not-link, and investigate which privacy settings are better predictors for determining malicious or benign apps based on app permissions.

References

[1] Abaker, Ali A., and Fakhreldeen A. Saeed. “A Comparative Analysis of Machine Learning Algorithms to Build a Predictive Model for Detecting Diabetes Complications.” Informatica (Slovenia), 45(1), 117–25, 2021. doi:10.31449/inf.v45i1.3111.

[2] Adhikari, Rajindra, et al. “Security and Privacy Issues Related to the Use of Mobile Health Apps.” 25th Australasian Conference on Information Systems (ACIS 2014), 2014.

[3] Apple Inc. App Store Downloads on iTunes. https://apps.apple.com/in/genre/ios/id36. Accessed 6 Apr. 2021.

[4] Bhatt, Arpita Jadhav, et al. “iABC: Towards a Hybrid Framework for Analyzing and Classifying Behaviour of iOS Applications Using Static and Dynamic Analysis.” Journal of Information Security and Applications, 41, 144–58, 2018. doi:10.1016/j.jisa.2018.07.005.

[5] Chehal, Dimple, et al. “Predicting the Usefulness of E-Commerce Products’ Reviews Using Machine Learning Techniques.” Informatica (Slovenia),47(2),275–84, 2023. doi:10.31449/inf.v47i2.4155.

[6] Congyi, Deng, and Shi Guangshun. “Method for Detecting Android Malware Based on Ensemble Learning.” ACM International Conference Proceeding Series, 8–31, 2020. doi:10.1145/3409073.3409084.

[7] Coronado-De-Alba, Lilian D., et al. “Feature Selection and Ensemble of Classifiers for Android Malware Detection.” 2016 8th IEEE Latin-American Conference on Communications, LATINCOM 2016, 128, 2–7, 2016. doi:10.1109/LATINCOM.2016.7811605.

[8] Erickson, Jeremy, et al. AndroidLeaks: Detecting Privacy Leaks In Android Applications., 1–17, 2011. http://www.cs.ucdavis.edu/research/tech-reports/2011/CSE-2011-10.pdf.

[9] Harahsheh, Khawlah, and Chung Hao Chen. “A Survey of Using Machine Learning in IoT Security and the Challenges Faced by Researchers.” Informatica (Slovenia), 47(6), 1–54, 2023. doi:10.31449/inf.v47i6.4635.

[10] Huang, Chun Ying, et al. “Performance Evaluation on Permission-Based Detection for Android Malware.” Advances in Intelligent Systems and Applications,2, 111–20, 2013. doi:10.1007/978-3-642-35473-1_12.

[11] Idrees, Fauzia, et al. “PIndroid: A Novel Android Malware Detection System Using Ensemble Learning Methods.” Computers and Security, 68, 36–46, 2017. doi:10.1016/j.cose.2017.03.011.

[12] Jing, Yiming, et al. “RiskMon : Continuous and Automated Risk Assessment of Mobile Applications.” Proceedings of the 4th ACM Conference on Data and Application Security and Privacy - CODASPY ’14, 99–110, 2014. doi:10.1145/2557547.2557549.

[13] Kang, Hyunjae, et al. “Detecting and Classifying Android Malware Using Static Analysis along with Creator Information.” International Journal of Distributed Sensor Networks, 11(6), Hindawi Publishing Corporation, 479174,2015.

[14] Khan, Jalaluddin, et al. “Survey on Mobile User’s Data Privacy Threats and Defense Mechanisms.” Procedia Computer Science, 56(1), Elsevier Masson SAS, 376–83,2015. doi:10.1016/j.procs.2015.07.223.

[15] Krupp, Brian. “Enhancing Security And Privacy For Mobile Systems.” Doctoral Dissertation, Department of Electrical and Computer Engineering, Cleveland State University, 148, 2015.

[16] Liu, Xing, and Jiqiang Liu. “A Two-Layered Permission-Based Android Malware Detection Scheme.” Proceedings - 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, MobileCloud 2014, 128, IEEE, 142–48, 2014. doi:10.1109/MobileCloud.2014.22.

[17] Mesevage, Tobias Geisler. Machine Learning Classifiers - The Algorithms & How They Work. https://monkeylearn.com/blog/what-is-a-classifier/. Accessed 28 May 2021.

[18] Millman, Rene. Oxford Researchers Expose Personal Data Harvesting in Third-Party Facebook and Google Apps. https://www.itpro.co.uk/privacy/32190/oxford-researchers-expose-personal-data-harvesting-in-third-party-facebook-and-google. Accessed 7 Apr. 2021.

[19] Ptsecurity.com. “Vulnerability and Threats in Mobile Applications.” Ptsecurity.Com, 2019.

[20] Saaty, Thomas L. “Decision Making with the Analytic Hierarchy Process.” Int. J. Services Sciences, 1(1), 2008.

[21] Sahal, Abdirashid Ahmed, et al. “Mining and Detection of Android Malware Based on Permissions.” 3rd International Conference on Computer Science and Engineering (UBMK 2018), IEEE, 264–68, 2018. doi:10.1109/UBMK.2018.8566510.

[22] Sun, Lichao, et al. “Significant Permission Identification for Machine-Learning-Based Android Malware Detection.” IEEE Transactions on Industrial Informatics, 14(7), 3216–25, 2018. doi:10.1109/tii.2017.2789219.

[23] Wang, Wei, et al. “Exploring Permission-Induced Risk in Android Applications for Malicious Application Detection.” IEEE Transactions on Information Forensics and Security, 9(11),1869–82, 2014. doi:10.1109/TIFS.2014.2353996.

[24] Wikipedia. iOS. https://en.wikipedia.org/wiki/IOS. Accessed 6 Apr. 2021.

[25] WIRED. Thousands of Android and iOS Apps Leak Data From the Cloud. https://www.wired.com/story/ios-android-leaky-apps-cloud/. Accessed 7 Apr. 2021.

[26] Xiong, Ping, et al. "Android Malware Detection with Contrasting Permission Patterns." China Communications, 11(8), China Institute of Communications, 1–14, 2014. doi:10.1109/CC.2014.6911083.

[27] Zhou, Victor. Machine Learning for Beginners: An Introduction to Neural Networks . https://towardsdatascience.com/machine-learning-for-beginners-an-introduction-to-neural-networks-d49f22d238f9. Accessed 28 May 2021.

JOŽEF STEFAN INSTITUTE

Jožef Stefan (1835-1893) was one of the most prominent physicists of the 19th century. Born to Slovene parents, he obtained his Ph.D. at Vienna University, where he was later Director of the Physics Institute, Vice-President of the Vienna Academy of Sciences and a member of several sci- entific institutions in Europe. Stefan explored many areas in hydrodynamics, optics, acoustics, electricity, magnetism and the kinetic theory of gases. Among other things, he originated the law that the total radiation from a black body is proportional to the 4th power of its absolute tem- perature, known as the Stefan–Boltzmann law.

The Jožef Stefan Institute (JSI) is the leading independent scientific research institution in Slovenia, covering a broad spectrum of fundamental and applied research in the fields of physics, chemistry and biochemistry, electronics and information science, nuclear science technology, en- ergy research and environmental science.

The Jožef Stefan Institute (JSI) is a research organisation for pure and applied research in the natural sciences and technology. Both are closely interconnected in research de- partments composed of different task teams. Emphasis in basic research is given to the development and education of young scientists, while applied research and development serve for the transfer of advanced knowledge, contributing to the development of the national economy and society in general.

At present the Institute, with a total of about 900 staff, has 700 researchers, about 250 of whom are postgraduates, around 500 of whom have doctorates (Ph.D.), and around 200 of whom have permanent professorships or temporary teaching assignments at the Universities.

In view of its activities and status, the JSI plays the role of a national institute, complementing the role of the uni- versities and bridging the gap between basic science and applications.

Research at the JSI includes the following major fields: physics; chemistry; electronics, informatics and computer sciences; biochemistry; ecology; reactor technology; ap- plied mathematics. Most of the activities are more or less closely connected to information sciences, in particu- lar computer sciences, artificial intelligence, language and speech technologies, computer-aided design, computer architectures, biocybernetics and robotics, computer automa- tion and control, professional electronics, digital communications and networks, and applied mathematics.

The Institute is located in Ljubljana, the capital of the in dependent state of **Slove**nia (or S♡nia). The capital today isconsidered a crossroad bet between East, West and Mediter-ranean Europe, offering excellent productive capabilities and solid business opportunities, with strong international connections. Ljubljana is connected to important centers such as Prague, Budapest, Vienna, Zagreb, Milan, Rome, Monaco, Nice, Bern and Munich, all within a radius of 600 km.

From the Jožef Stefan Institute, the Technology Park “Ljubljana” has been proposed as part of the national strat- egy for technological development to foster synergies be- tween research and industry, to promote joint ventures be- tween university bodies, research institutes and innovative industry, to act as an incubator for high-tech initiatives and to accelerate the development cycle of innovative products.

Part of the Institute was reorganized into several high-tech units supported by and connected within the Technology park at the Jožef Stefan Institute, established as the beginning of a regional Technology Park "Ljubljana". The project was developed at a particularly historical moment, characterized by the process of state reorganisation, privatisation and private initiative. The national Technology Park is a shareholding company hosting an independent venture-capital institution.

The promoters and operational entities of the project are the Republic of Slovenia, Ministry of Higher Education, Science and Technology and the Jožef Stefan Institute. The framework of the operation also includes the University of Ljubljana, the National Institute of Chemistry, the Institute for Electronics and Vacuum Technology and the Institute for Materials and Construction Research among others. In addition, the project is supported by the Ministry of the Economy, the National Chamber of Economy and the City of Ljubljana.

Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Tel.: +386 1 4773 900, Fax.: +386 1 251 93 85
WWW: http://www.ijs.si
E-mail: matjaz.gams@ijs.si
Public relations: Polona Strnad

Informatica

An International Journal of Computing and Informatics

Web edition of Informatica may be accessed at: http://www.informatica.si.

Subscription Information Informatica (ISSN 0350-5596) is published four times a year in Spring, Summer, Autumn, and Winter (4 issues per year) by the Slovene Society Informatika, Litostrojska cesta 54, 1000 Ljubljana, Slovenia.
The subscription rate for 2022 (Volume 46) is
– 60 EUR for institutions,
– 30 EUR for individuals, and
– 15 EUR for students
Claims for missing issues will be honored free of charge within six months after the publication date of the issue.

Typesetting: Blaž Mahnič, Gašper Slapničar; gasper.slapnicar@ijs.si
Printing: ABO grafika d.o.o., Ob železnici 16, 1000 Ljubljana.

Orders may be placed by email (drago.torkar@ijs.si), telephone (+386 1 477 3900) or fax (+386 1 251 93 85). The payment should be made to our bank account no.: 02083-0013014662 at NLB d.d., 1520 Ljubljana, Trg republike 2, Slovenija, IBAN no.: SI56020830013014662, SWIFT Code: LJBASI2X.

Informatica is published by Slovene Society Informatika (president Niko Schlamberger) in cooperation with the following societies (and contact persons):
Slovene Society for Pattern Recognition (Vitomir Štruc)
Slovenian Artificial Intelligence Society (Sašo Džeroski)
Cognitive Science Society (Olga Markič)
Slovenian Society of Mathematicians, Physicists and Astronomers (Dragan Mihailović)
Automatic Control Society of Slovenia (Giovanni Godena)
Slovenian Association of Technical and Natural Sciences / Engineering Academy of Slovenia (Mark Pleško)
ACM Slovenia (Ljupčo Todorovski)

Informatica is financially supported by the Slovenian research agency from the Call for co-financing of scientific periodical publications.

Informatica is surveyed by: ACM Digital Library, Citeseer, COBISS, Compendex, Computer & Information Systems Abstracts, Computer Database, Computer Science Index, Current Mathematical Publications, DBLP Computer Science Bibliography, Directory of Open Access Journals, InfoTrac OneFile, Inspec, Linguistic and Language Behaviour Abstracts, Mathematical Reviews, MatSciNet, MatSci on SilverPlatter, Scopus, Zentralblatt Math

Volume 49 Number 1 March 2025 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics

Start of Special Issue

End of Special Issue / Start of Normal Papers

Informatica **49** (2025) Number 1, pp. 1–222