Sakarya University
Journal of Computer and Information Sciences

e-ISSN 2636-8129
VOLUME 8 ISSUE 2
JUNE 2025

VOLUME: 8 ISSUE: 2
E-ISSN 2636-8129

JUNE 2025
http://saucis.sakarya.edu.tr/tr/

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES

SAKARYA
ÜNİVERSİTESİ

The Owner on Behalf of Sakarya University

Editor in Chief

Managing Editors

Editorial Board

Language Editors

Layout Editor

Indexing & Abstracting & Archiving

Contents

Research Article

Reviews

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
http://saucis.sakarya.edu.tr/
e-ISSN: 2636-8129
Publisher: Sakarya University
Vol. 8, No. 2, 171-183, 2025
DOI: 10.35377/saucis...1592312
RESEARCH ARTICLE

Performance Analysis and Optimization of Enterprise Wireless Networks Based on 802.11ax Technology

Berkay Dağtaş[1,*], İsmail Hakkı Cedimoğlu[2]

[1]Sakarya University, Institute Of Science, Sakarya, Türkiye, ror.org/04ttnw109
[2]Sakarya University, Sakarya, Türkiye, ror.org/04ttnw109

Corresponding author:
Berkay Dağtaş, Sakarya University, Faculty of Computer and Information Sciences, Sakarya, Türkiye
berkay.dagtas1@ogr.sakarya.edu.tr

Article History:
Received: 27.11.2024
Revised: 26.02.2025
Accepted: 18.04.2025
Published Online: 13.06.2025

ABSTRACT
This study focuses on performance analysis and optimization of enterprise wireless networks. Fundamental performance parameters of Wi-Fi networks such as signal strength, signal-to-noise ratio (SNR), data rate, and channel interference were evaluated in detail in the study. The analysis process was carried out using Ekahau AI Pro software and Ekahau Sidekick device in a corporate facility, consisting of three main buildings. The obtained data revealed that signal strength dropped to -85 dBm levels in certain areas, negatively affecting the network's coverage area. Particularly in the ground floor of building B-C, secondary signal levels were found to be insufficient for roaming. Across the campus, SNR levels were observed to be 30 dB and above, and these values were found to provide ideal connectivity. During the analysis, it was discovered that in some areas, the number of access points broadcasting signals on the same channel increased up to 6. It has been assessed that this situation may negatively affect network performance in areas where interference is intense. Data rates varied between 1 - 300 Mbps in the 2.4 GHz frequency band and 1 - 585 Mbps in the 5 GHz band. The study provides significant data for performance analysis and optimization of enterprise wireless networks.

Keywords: Wi-Fi performance, Signal strength, SNR, Channel interference, Ekahau AI Pro

1. Introduction

With the rapid development of information and communication technology, the demand for wireless network technology has increased significantly. Wi-Fi is easy to implement in workplaces and educational environments, enabling users to access the internet anytime and anywhere. While this technology provides data communication through radio waves, it typically operates in the 2.4 GHz and 5 GHz radio frequency bands that do not require licensing [1]. These bands are reserved for unlicensed radio services that can be used without obtaining a radio station license.

Wi-Fi is a technology based on the IEEE 802.11 standard that provides wireless data transfer [2]. While Wi-Fi modules were initially used only in tablets, laptops, and smartphones, nowadays, it is possible to find them in many electronic devices, including cameras, printers, 3D printers and multi-cookers. Access points are used to establish wireless network connections. These devices provide remote access to users by broadcasting wireless network signals [3].

Enterprise wireless network systems are typically managed through an access point controller. Different wireless network infrastructures, systems, and services are distributed by this controller and delivered to users through access points[4]. This structure enables effective and secure management of wireless connections in large-scale networks.

This study aims to examine, analyze, and optimize the performance of an enterprise wireless network in a real-world scenario based on the evaluation results. To achieve these goals, tests were conducted using spectrum analysis tools and software that help analyze the results. These tests aim to identify the network's strengths and weaknesses and reveal the main factors affecting user behavior in resource usage. A professional solution including advanced network analysis software and a set of sensors was chosen to analyze wireless network performance. Tests using Ekahau AI Pro Version 2.0 allowed performance measurements and analysis to be performed on various parts of the network.

This software has the capacity to examine many factors affecting network performance and evaluates parameters such as spectrum analysis, signal strength measurements, data transfer rates, and network coverage areas in detail. This analysis process was conducted based on metrics like capacity management, bandwidth usage, signal quality, and channel interference to understand the overall health and performance of the network. Based on the findings, improvement steps were implemented

Cite as: B. Dağtaş and İ. Hakkı Cedimoğlu. (2025). Performance analysis and optimization of enterprise wireless networks based on 802.11ax technology. *Sakarya University Journal of Computer and Information Sciences*. vol. 8, no. 2, pp. 171-183, 2025 doi: 10.35377/saucis...1592312

to increase wireless network performance, and user experience was improved. This study conducts a detailed analysis of wireless network performance in a corporate scenario, comprehensively presents the current state of the network, implements optimization processes based on the findings, and makes a practical contribution to the literature in this field.

2. Literature Review

2.1 Wi-Fi Standards

Wi-Fi is a modern standard for data transmission and reception between devices. These devices must be equipped with radio modules. Wi-Fi standards are defined by the Institute of Electrical and Electronics Engineers (IEEE). The standards differ in terms of frequency bands, data transfer rates, and other features [5]. Wi-Fi standards are presented in Table 1.

Table 1. Wi-Fi Versions and Technical Specifications

Wi-Fi Version	IEEE Standard	Frequency Band	Data Rate	Modulation Technique	Year
Wi-Fi 0	IEEE 802.11	2.4 GHz	2 Mbit/s	FHSS, DSSS	1997
Wi-Fi 1	IEEE 802.11b	2.4 GHz	11 Mbit/s	DSSS, CCK	1999
Wi-Fi 2	IEEE 802.11a	5 GHz	54 Mbit/s	OFDM	1999
Wi-Fi 3	IEEE 802.11g	2.4 GHz	54 Mbit/s	OFDM	2003
Wi-Fi 4	IEEE 802.11n	2.4 GHz, 5 GHz	600 Mbit/s	OFDM, MIMO	2009
Wi-Fi 5	IEEE 802.11ac	5 GHz	3.5 Gbit/s	OFDM, MIMO	2013
Wi-Fi 6	IEEE 802.11ax	2.4 GHz, 5 GHz	9.6 Gbit/s	OFDMA, MU-MIMO, TWT	2021
Wi-Fi 7	IEEE P802.11be	2.4 GHz, 5 GHz, 6 GHz	~40 Gbit/s	OFDMA, MU-MIMO, 320 MHz channels	2024

2.2. Factors Affecting Wireless Network Performance

This section discusses the key factors that affect the performance of wireless networks. Analysis of these factors affecting network performance helps understand existing problems and contributes to improving network efficiency.

2.3. Effects of Obstacles and Building Materials

Penetration loss refers to the reduction in power of a radio signal as it passes through a medium in wireless communication. This loss occurs when the signal weakens while passing through different materials (such as walls, doors, glass, metal surfaces) [6]. Wireless network performance varies significantly depending on how much radio frequency signals are weakened by obstacles and building materials. Table 2 shows how much common building materials weaken Wi-Fi signals, measured in dB [7].

Table 2. Effects of Building Materials on Wi-Fi Signals

Materials	Attenuation (dB)
Drywall	3 dB
Bookshelf	2 dB
Exterior Glass	3 dB
Solid Wood Door	6 dB
Marble	6 dB
Brick	10 dB
Concrete	12 dB
Elevator Shaft	30 dB

When radio signals encounter an obstacle, some of them are absorbed, reflected, or refracted, causing the signal to reach its target with less power. Penetration loss can significantly affect wireless network performance and is a common problem typically encountered in indoor environments.

These losses vary depending on the frequency used, the material of the medium, and the distance the signal needs to travel through. For example, high-frequency signals (like 5 GHz) usually experience more penetration loss compared to low-frequency signals (like 2.4 GHz) [8]. Therefore, penetration loss is a critical parameter in terms of Wi-Fi coverage planning and performance optimization.

2.4. Signal Strength (RSSI - Received Signal Strength Indicator)

The Received Signal Strength Indicator (RSSI) is a critical measurement in Wi-Fi networks that represents the power level of the signal a wireless device receives from an access point or router. The RSSI value is measured in decibels (dBm) and provides an indication of the signal strength at the receiver. The closer an end device is to the network, the higher the measured RSSI value will be [9]. RSSI values are presented in Table 3.

Table 3. RSSI Values

Signal Strength (dBm)	Category
-30 dBm	Excellent
-67 dBm	Very Good
-70 dBm	Acceptable
-80 dBm	Weak
-90 dBm	Insufficient

2.5. Signal to Noise Ratio (SNR)

Signal-to-Noise Ratio (SNR) is a parameter that measures the strength of a data signal relative to the background noise level. This ratio is critically important for evaluating signal quality, and the SNR calculation method is shown in Equation 1.

$$\boldsymbol{SNR(dB) = 10.log_{10}\left(\frac{P_{signal}}{P_{noise}}\right)} \quad (1)$$

Here;

P_{Signal} : Signal power (in Watts)

P_{noise} : Noise power (in Watts)

SNR is measured in dB instead of dBm. In cases where the measurements are already in decibels (dB), the difference between the signal power and the noise power directly gives the signal-to-noise ratio (SNR). This is shown in Equation 2.

$$\boldsymbol{SNR = S - N} \quad (2)$$

Here;

S : Signal level (in dB)

N : Noise level (in dB)

Even if a client receives an excellent signal in close proximity to an access point, strong signals originating from neighboring wireless networks or radio frequency (RF) devices may disrupt connection continuity. Such signals are interpreted as noise by the client device [10]. The relationship between SNR levels and signal quality is presented in Table 4 [11].

Table 4. Relationship between SNR Levels and Signal Quality

SNR Level	Description
> 40 dB	Excellent quality
25-40 dB	Very Good quality
15-25 dB	Good quality
10-15 dB	Moderate quality
0-10 dB	Low or no signal

2.6. Channel Interference

Channel interference in wireless communication is one of the most important factors affecting wireless network performance. This situation, which is frequently encountered especially in enterprise wireless networks, is examined under two headings.

2.6.1. Co-Channel Interference

Co-Channel Interference (CCI) occurs when two or more Access Points (APs) configured on the same channel are physically positioned close to each other, regardless of the frequency band (2.4GHz / 5GHz). This situation causes signals to interfere with each other and creates interference [12]. The 802.11 standard is based on "Carrier Sense Multiple Access" (CSMA) technique [13]. This requires a device to transmit data only when the channel is free. CCI can confuse the carrier sensing mechanism by giving the impression that the channel is busy, which may prevent the channel from being used.

2.6.2. Adjacent Channel Interference

Adjacent Channel Interference (ACI) is a type of interference where signals from transmitters on adjacent frequency channels "leak" and cause interference on another channel [14]. Adjacent channel interference is caused by overlapping channels in the 2.4 GHz band. In this band, there are only three non-overlapping channels 1, 6 and 11 [15]. Figure 1 shows the channel planning in 2.4 GHz and 5 GHz frequency bands respectively [16].

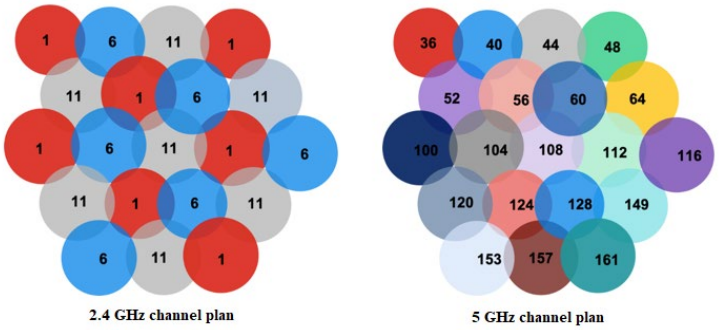


Figure 1. Channel allocation scheme for 2.4 GHz and 5 GHz Wi-Fi bands

3. Materials and Methods

Figure 2. Wi-Fi Performance Analysis and Optimization Process

3.1 Using Ekahau and Alternative Tools in Wireless Network Performance Analysis

Evaluating wireless network performance is critically important for reliable data communication and user experience. Therefore, various software tools are used to optimize the performance of wireless networks during and after installation. These tools aim to improve network efficiency by analyzing network capacity, signal strength, signal-to-noise ratio (SNR), data rates, and other performance metrics. The comparative features of Wi-Fi performance testing tools are presented in Table 5 [17], [18], [19], [20].

Table 5. Wi-Fi Features and Comparisons of Wi-Fi Performance Testing Tools

Feature	Ekahau AI Pro	Acrylic Wi-Fi	AirMagnet Survey Pro	NetSpot
Number of Heatmaps	26	14	11	15
Supported Frequencies	2.4 GHz, 5 GHz, 6 GHz	2.4 GHz, 5 GHz, 6 GHz	2.4 GHz, 5 GHz, 6 GHz	2.4 GHz, 5 GHz, 6 GHz
Wi-Fi Standards	802.11a/b/g/n/ac/ax	802.11a/b/g/n/ac/ax	802.11a/b/g/n/ac/ax	802.11a/b/g/n/ac/ax
User Level	Professional IT experts	From beginners to professionals	Professional IT experts	From beginners to professionals
Reporting Formats	PDF, CSV, DOCX	PDF, CSV, DOCX	PDF, CSV, XML	PDF, CSV
Spectrum Analysis	Yes	Yes	Yes	Yes

3.2. Wireless Network Performance Testing

In this section, it is explained how enterprise Wi-Fi network design and performance tests were conducted using Ekahau AI Pro software and Sidekick device. The study was carried out in a facility consisting of blocks A, B, and C. The area information of these blocks included in the study is presented in Table 6, and the architectural plans are shown in Figure 3.

Table 6. Building Floor Areas and Measurements

Building	Floor	Area (Square Meters)
A	Ground Floor	16695
B-C	First Floor	5110
	Ground Floor	6199

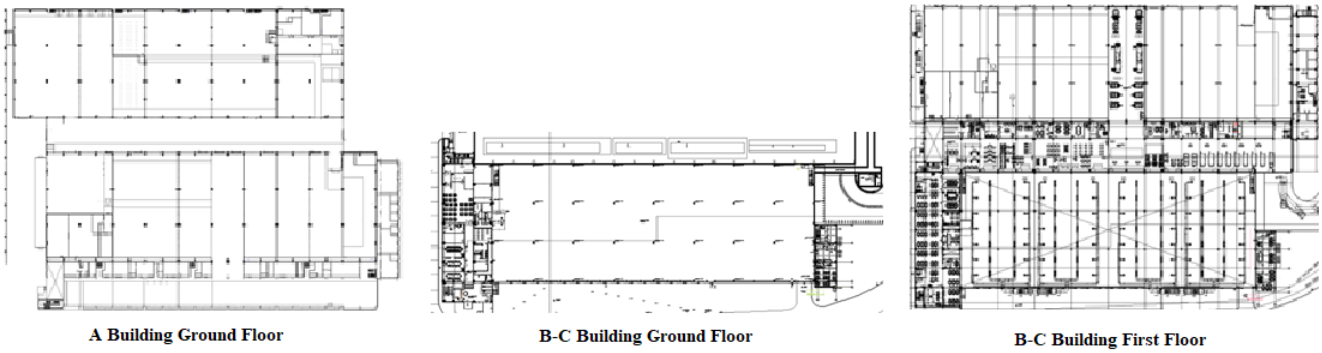


Figure 3. Floor Plans of A, B, and C Buildings

Measurements were first carried out in an office environment using the automatic planner feature of Ekahau AI Pro software, then conducted on-site using Ekahau AI Pro software installed on an iPad and the Ekahau Sidekick 2 device.

The Ekahau AI Automatic Planner was used to provide a fast and effective simulation of the wireless network design. After uploading the facility's architectural plans in DWG format into the Ekahau AI Pro software, user-dense areas were identified, and the Huawei 5776-26 access points, which support the 802.11ax standard, were automatically positioned throughout the facility. The AI Automatic Planner placed the access points based on the coverage requirements defined by Ekahau Best Practices, aiming to maximize signal coverage for both the 2.4 GHz and 5 GHz frequency bands while minimizing the risk of channel overlap. The Ekahau automatic planner results are presented in Figure 4.

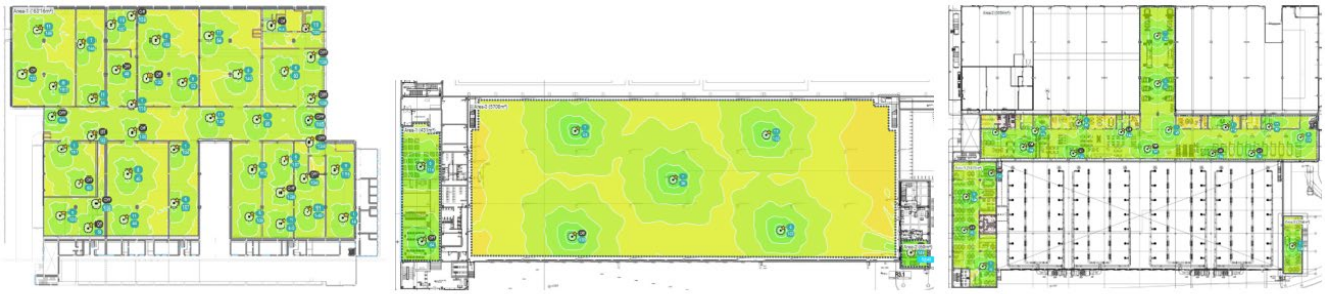


Figure 4. Ekahau Automatic Planner Result for A, B, and C Buildings

A comprehensive on-site test was conducted using the Ekahau Sidekick device to evaluate the performance of the wireless network designed using the automatic planner under real field conditions. On-site tests were planned and carried out to cover personnel-intensive areas where the wireless network would be actively used.

Ekahau Sidekick 2 is one of the fastest and most precise measuring devices used for analyzing wireless network performance. The device can perform highly accurate measurements in 2.4 GHz, 5 GHz, and 6 GHz frequency bands, and with its 9 integrated broadband 3D antennas, it ensures accurate capture of signals from all directions. Additionally, environmental interference was quickly detected with its spectrum analysis feature that can perform 50 scans per second. This allowed parameters such as signal strengths, SNR (Signal-to-Noise Ratio), and channel interference to be analyzed accurately and reliably. The routes followed during measurements in blocks A, B, and C are shown in Figure 5.

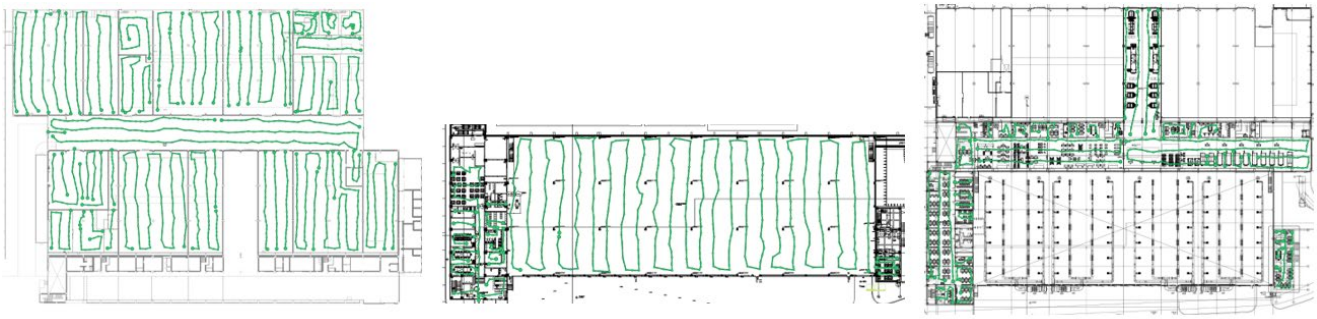


Figure 5. Measurement Routes Conducted with Sidekick in A, B, and C Buildings

On-site tests followed Ekahau Best Practices guidelines, and performance data for both 2.4 GHz and 5 GHz frequency bands were analyzed in detail. These are presented in Table 7.

Table 7. Ekahau Best Practices

Criteria	2.4GHz	5GHz
Signal Strength Min	-67 dBm	-67 dBm
Secondary Signal Strength Min	-67 dBm	-67 dBm
Tertiary Signal Strength	OFF	OFF
Signal-to-Noise Ratio Min	20 dB	25 dB
Data Rate Min	24 Mbps	24 Mbps
Channel Interference Max	2 at min. -85.0 dBm	1 at min. -85.0 dBm

4. Results

In this study, wireless network performance was assessed based on key parameters including signal strength, signal-to-noise ratio (SNR), throughput, and channel interference. Additionally, the effectiveness of applied optimization strategies was analyzed. Signal strength measurements were conducted across blocks A, B, and C in both 2.4 GHz and 5 GHz frequency bands. The results, illustrated in Figures 6 and 7, demonstrate the signal distribution in the evaluated areas. Green regions correspond to optimal signal levels, yellow indicates areas near threshold limits, while gray represents locations with insufficient or no coverage.

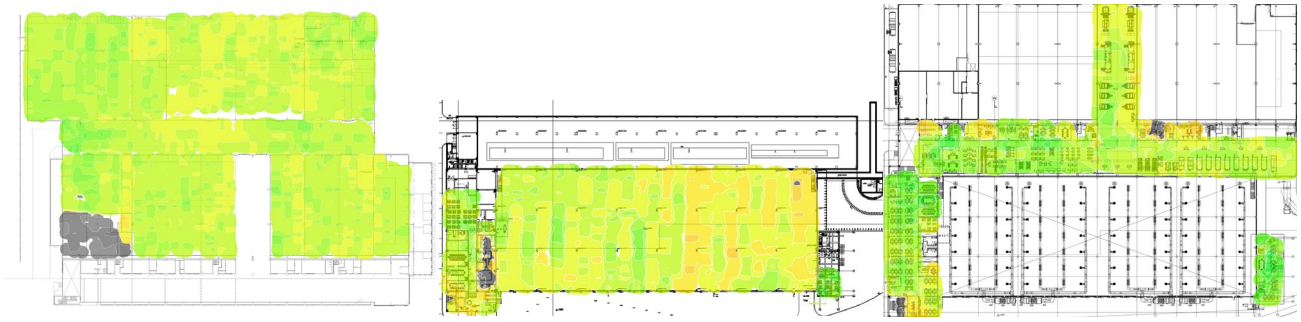


Figure 6. 2.4 GHz Signal Strength Heatmaps

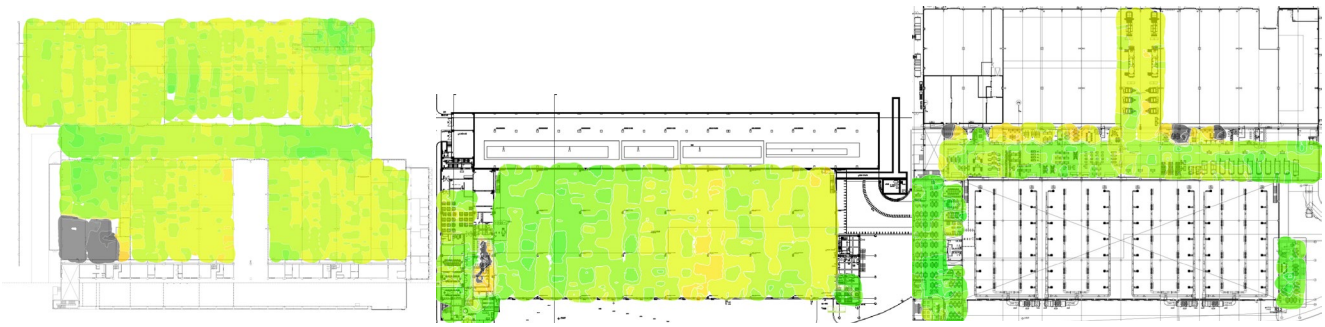


Figure 7. 5 GHz Signal Strength Heatmaps

Secondary signal strength is a critical metric in evaluating wireless network performance, particularly in maintaining seamless handover between access points (APs) and ensuring redundancy in the event of AP failures. Analysis results indicate that in areas where secondary signal strength is equal to or greater than -67 dBm in both the 2.4 GHz and 5 GHz frequency bands,

these performance criteria are successfully met. Conversely, gray zones with signal levels equal to or below -85 dBm reveal insufficient support for client roaming, as illustrated in Figures 8 and 9.

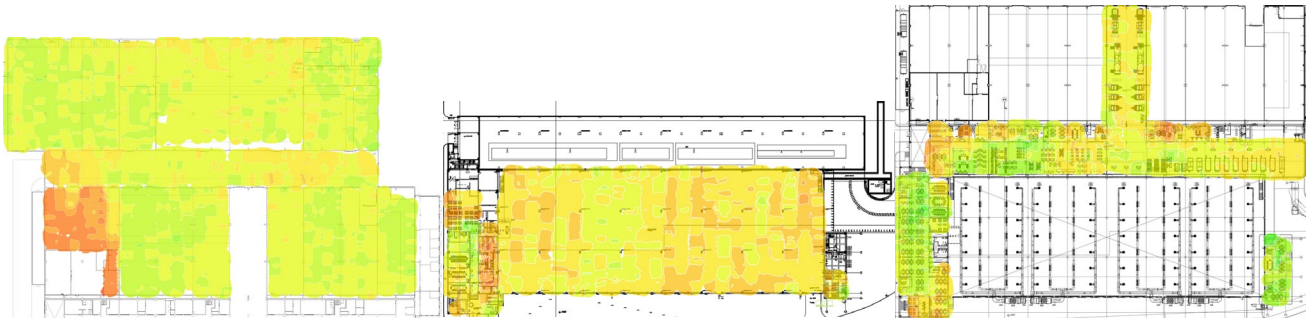


Figure 8. 2.4 GHz Secondary Signal Strength Heatmaps

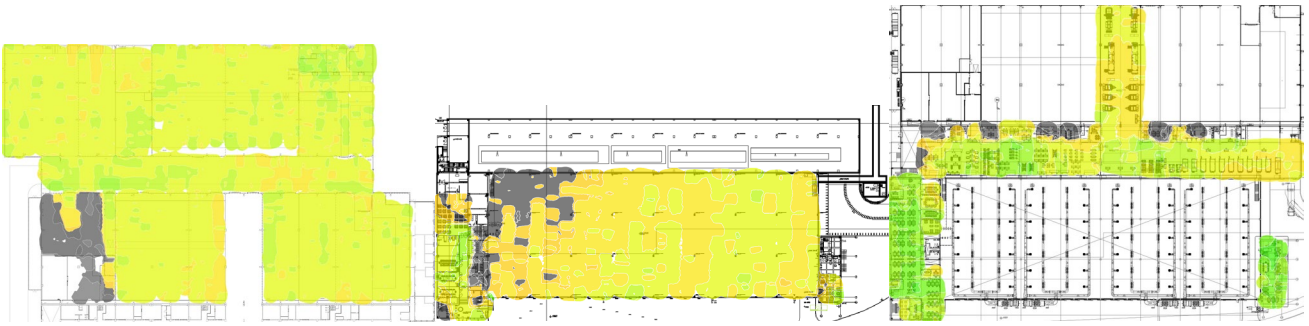


Figure 9. 5 GHz Secondary Signal Strength Heatmaps

For successful data transmission, the signal strength must exceed the level of background noise. Figures 10 and 11 present the SNR (Signal-to-Noise Ratio) heatmaps obtained for the ground floor and first floor of blocks A and B-C. In these maps, green areas indicate high SNR values (≥ 30 dB), which are associated with excellent signal quality and reliable communication. Conversely, yellow and gray areas represent zones with lower SNR levels, potentially leading to degraded performance or connection instability.

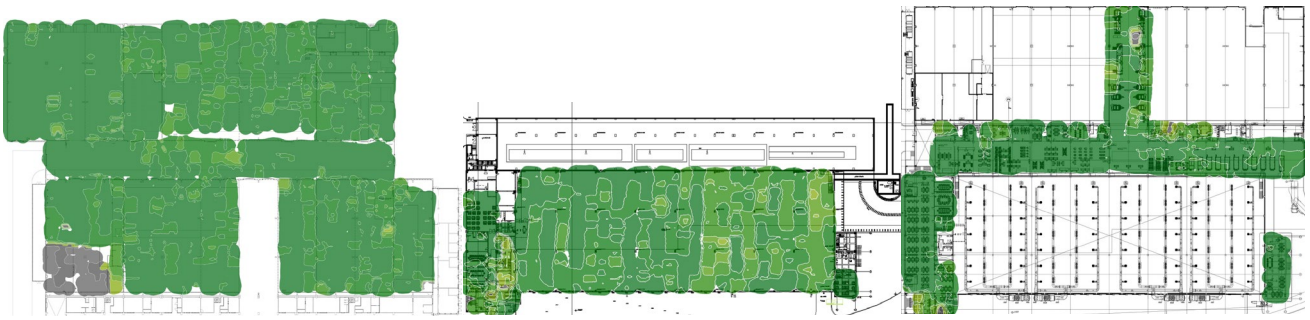


Figure 10. 2.4 GHz SNR Heatmaps

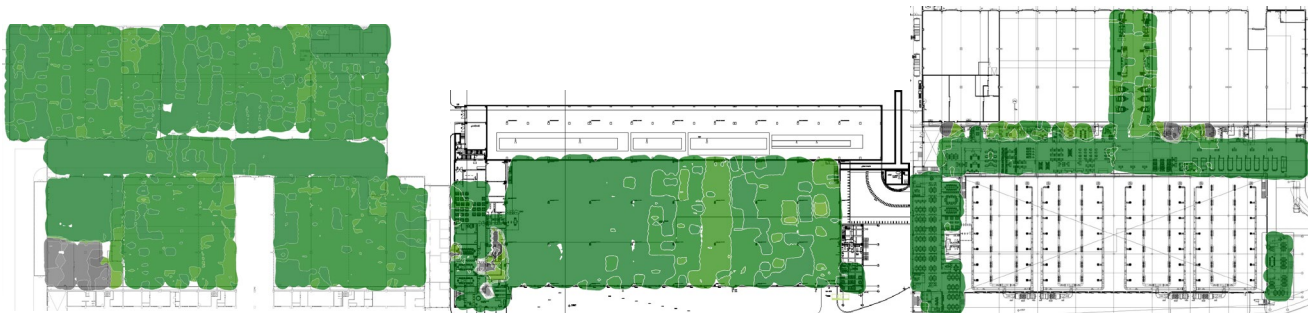


Figure 11. 5 GHz SNR Heatmaps

The data rate analyses conducted in the 2.4 GHz and 5 GHz frequency bands for blocks A and B-C are presented in Figures 12 and 13. The data rate refers to the maximum speed at which wireless devices can transmit or receive data, typically expressed in megabits per second (Mbps). In the 2.4 GHz frequency band, data rates ranged between 1 Mbps and 300 Mbps,

whereas in the 5 GHz band, they varied between 1 Mbps and 585 Mbps. In the heatmaps, green-colored areas denote regions with higher data rates, indicating efficient data transmission capabilities, while yellow and orange areas represent locations where the data transmission rates are relatively lower.

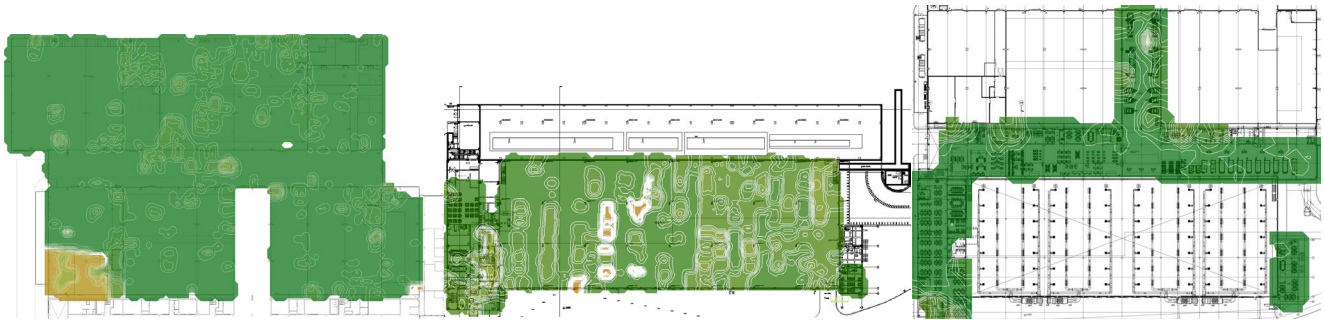


Figure 12. 2.4 GHz Data Rate Heatmaps

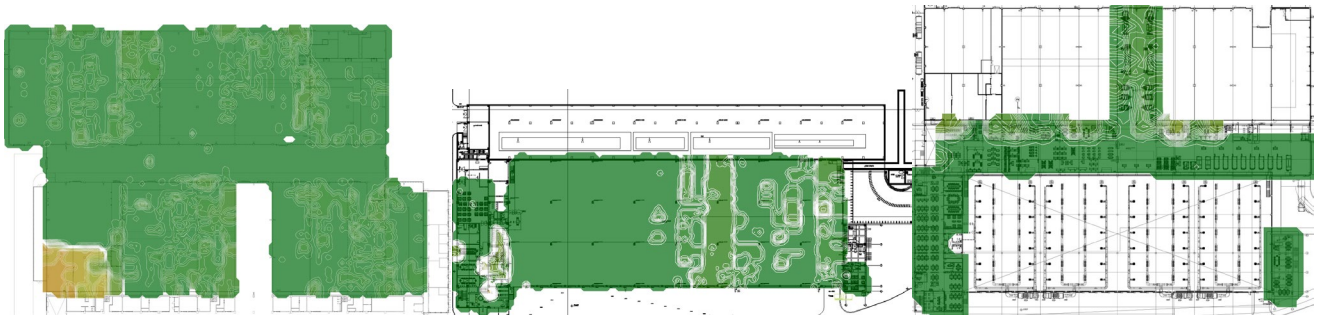


Figure 13. 5 GHz Data Rate Heatmaps

The throughput analyses for the 2.4 GHz and 5 GHz frequency bands in blocks A and B-C are illustrated in Figures 14 and 15. Throughput refers to the actual rate at which data is successfully transmitted over a network channel and is a critical metric that reflects end-user experience. While the data rate provides insight into the theoretical capacity of the network, throughput represents the practical, real-world performance. In the 2.4 GHz band, throughput values ranged from 1 Mbps to 240 Mbps, whereas in the 5 GHz band, values varied between 1 Mbps and 420 Mbps. As shown in the heatmaps, green areas correspond to zones with higher throughput performance, while yellow and orange areas indicate relatively lower throughput levels.

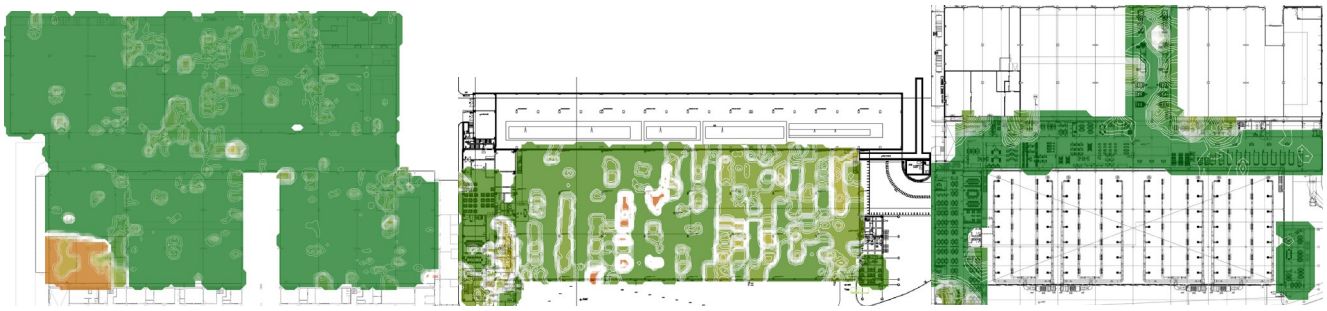


Figure 14. 2.4 GHz Throughput Heatmaps

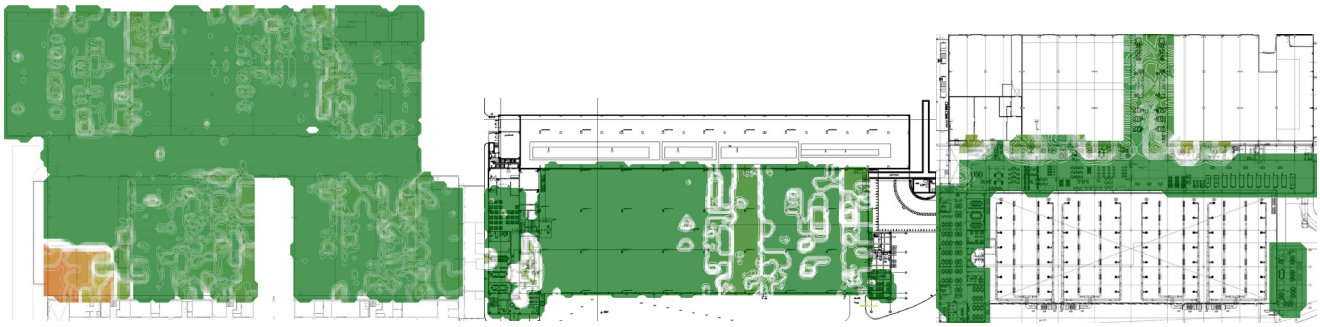


Figure 15. 5 GHz Throughput Heatmaps

The channel interference analysis is presented in Figures 16 and 17. In the heatmaps, green areas represent zones where interference is minimal or entirely absent, indicating optimal conditions for network performance. Conversely, the gray-shaded regions denote areas where multiple access points operate on the same channel, leading to co-channel interference. Such interference can significantly degrade wireless network efficiency and stability.

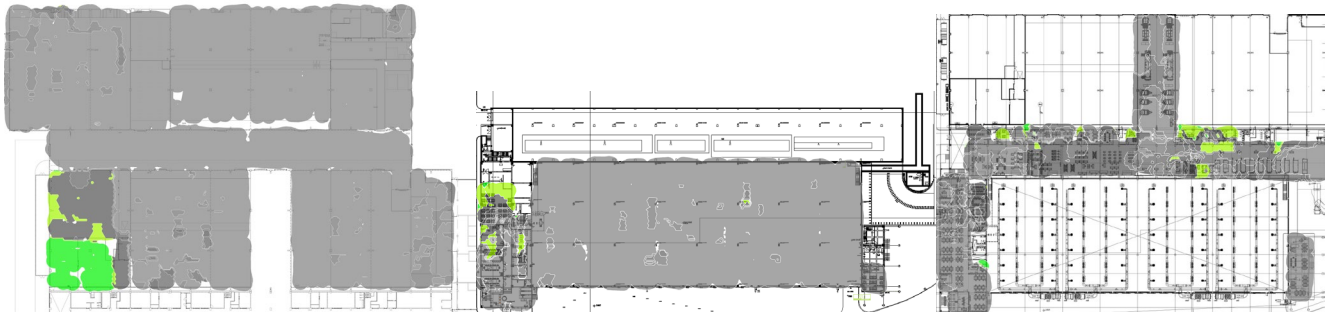


Figure 16. 2.4 GHz Channel Interference Heatmaps

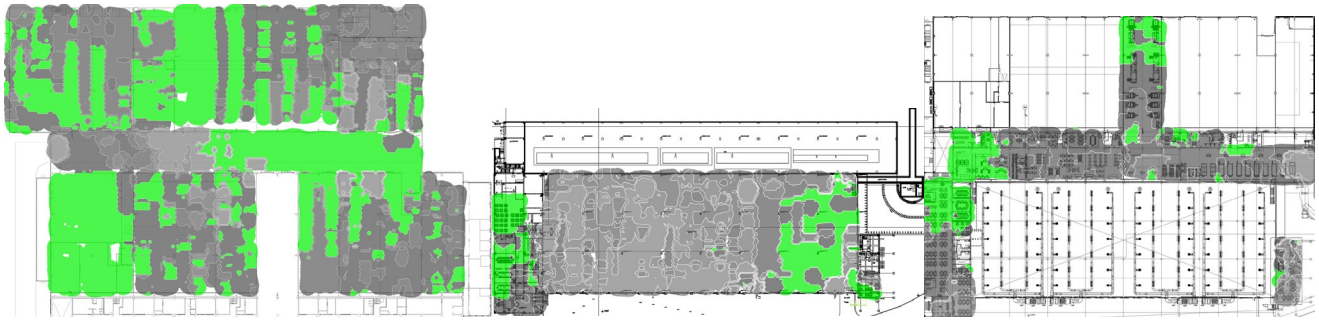


Figure 17. 5 GHz Channel Interference Heatmaps

The overall network performance heatmaps for the 5 GHz frequency band are presented in Figure 18. The 5 GHz band is commonly preferred in enterprise networks due to its lower channel congestion and greater bandwidth capacity. Accordingly, the analysis and optimization efforts in this study primarily focused on this band, where more effective improvements could be implemented. These heatmaps provide a comprehensive visual representation of performance bottlenecks and areas with optimization potential. In the performance maps, red and yellow tones indicate weak signal strength, whereas purple regions denote high levels of channel congestion.

The optimization efforts aimed at enhancing network performance in the 5 GHz frequency band and their results have been thoroughly presented. Based on the analyses obtained from performance tests, various optimization measures were implemented, and subsequent on-site assessments were carried out within the facility. The overall impact of these optimizations on network performance was comprehensively evaluated.

As illustrated in Figures 19 to 21, the improper placement of access points was identified as one of the main causes of weak Wi-Fi signal strength. In addition, signal degradation due to distance and the attenuating effect of interior building materials also contributed to reduced performance. To address these issues, the positions and orientations of access points were restructured, and additional access points were deployed in signal-poor areas. Moreover, transmit power levels were adjusted to improve signal coverage and distribution. These improvements not only resolved signal degradation but also significantly

enhanced low signal-to-noise ratios (SNR) and secondary signal strength, ultimately contributing to overall network performance.

Figure 18. 5 GHz Network Performance Heatmaps

Figure 19. Access Point Orientation and Placement Optimization.

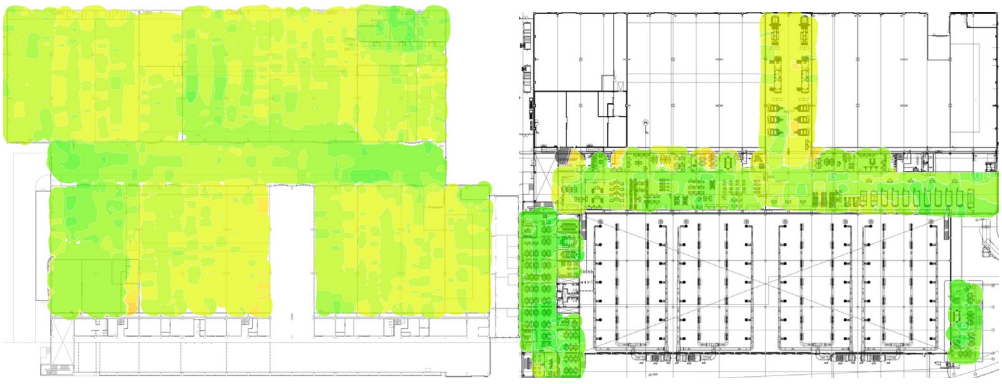


Figure 20. Optimized Signal Strength Heatmaps

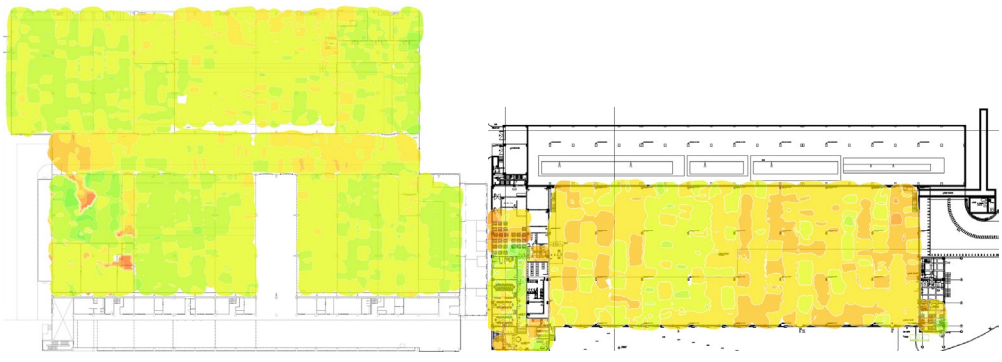


Figure 21. Optimized Secondary Signal Strength Heatmaps

To minimize channel interference and improve overall network performance, frequency and channel configurations of the access points were carefully optimized. In regions where interference was identified in the 2.4 GHz frequency band, the 2.4 GHz transmission was disabled due to the limited number of non-overlapping channels in this band, thereby preventing further signal congestion.

For the 5 GHz frequency band, which offers a broader spectrum, potential interference caused by wide channel usage was mitigated by restricting the channel width to 20 MHz. Furthermore, each access point was statically assigned to independent and non-overlapping channels. These configurations effectively reduced channel interference to ≤ 1 at the minimum signal strength threshold, thereby significantly improving the stability and performance of the wireless network across the entire facility, as illustrated in Figure 22.

The findings obtained from this study demonstrate the significant impact of the applied optimization strategies on wireless network performance. Prior to the implementation of these strategies, notable weaknesses in signal strength and severe interference issues were recorded, especially in zones with high user density. Following the repositioning of access points, reconfiguration of channels, and adjustment of power levels, a marked improvement in both network stability and performance was achieved. A comparative evaluation of performance metrics before and after the optimization efforts is presented in Table 8.

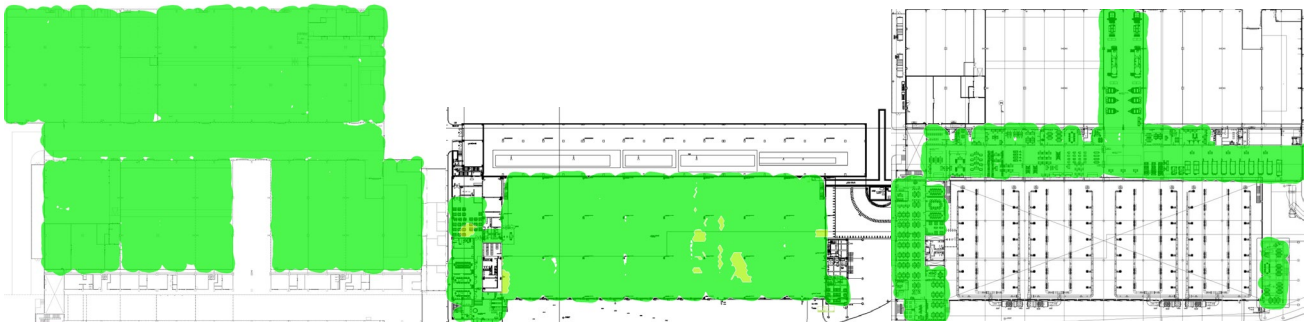


Figure 22. Optimized Channel Interference Heatmaps

Table 8. Comparison of Wireless Network Performance Metrics Before and After Optimization

Measurement Criteria	Before Optimization	After Optimization
Signal Strength	≤ -85 dBm	≥ -67 dBm
Secondary Signal Strength	≤ -67 dBm	≥ -67 dBm
Signal-to-Noise Ratio (SNR)	≤ 5 dB	≥ 30 dB
Data Rate	1 Mbps - 585 Mbps	1 Mbps - 585 Mbps
Throughput	1 Mbps - 420 Mbps	1 Mbps - 420 Mbps
Channel Interference at Minimum Signal Strength	-85 dBm ≥ 6	-85 dBm ≤ 1

5. Discussion

In the evaluation of wireless network performance, theoretical planning and software-based simulations alone are often insufficient. In this context, the importance of field tests and professional analysis tools becomes even more evident. Professional Wi-Fi analysis tools such as Ekahau AI Pro and Sidekick 2 not only enable signal strength measurements, but also allow for highly precise analysis of critical metrics such as spectrum activity, channel overlap, and signal-to-noise ratio (SNR). The results obtained in this study clearly demonstrate how effective and reliable these tools are in enterprise network environments.

In particular, the automatic planner feature offered by Ekahau AI Pro provides a significant advantage in optimizing access point (AP) placements by taking into account the internal building architecture. However, measurements conducted in the field have shown that relying solely on simulation results is insufficient. Real-time tests revealed significant differences between the planned AP placements and the actual signal distribution, especially with regard to secondary signal strength and roaming continuity. The success achieved through reorienting access points to enhance secondary signal strength contributes new insights to the literature. The findings indicate that secondary signal strength can be significantly improved not only by increasing the number of APs but also through proper orientation and strategic placement. Given the limited

number of studies in the literature that demonstrate field-based improvements in secondary signal coverage through directional AP optimization, this contribution is particularly valuable.

In parallel with similar studies in the literature, this study also employed common mitigation strategies for channel interference. Specifically, the 2.4 GHz band was disabled due to its limited channel structure, and the 5 GHz band was constrained to 20 MHz channels to minimize overlap. Additionally, the assignment of static and non-overlapping channels to each AP positively impacted network stability. As a result of these optimizations, measurable improvements were observed in signal strength, SNR, and client handover continuity.

In conclusion, the use of advanced analysis tools is of great importance in the performance analysis of enterprise wireless networks. Especially in corporate environments, issues such as signal leakage, increased interference, or roaming failures are considered unacceptable. The findings presented in this study prove that comprehensive, field-supported optimization strategies are essential for building a high-performance and secure Wi-Fi infrastructure.

6. Conclusions

The aim of this study was to conduct performance tests of a corporate company's wireless network and implement optimizations based on the results. Performance analysis methods were carried out through measurements. Throughout the study, Ekahau AI Pro software and Ekahau Sidekick 2 device were used on all building floors. During the measurements, areas where access points (APs) were located were examined, and parameters such as signal strength, SNR (Signal-to-Noise Ratio), data rate, network capacity, and channel interference were accurately and reliably analyzed. These measurements and analyses revealed that the network was not implemented as initially designed. Placing APs in locations other than those determined by the automatic planner caused AP insufficiency in some areas, while many APs broadcast signals beyond the targeted coverage zones. The best signal strength was measured as -30 dBm in targeted areas with high personnel density, whereas in more distant regions, signal strength dropped between -67 dBm and -85 dBm. At signal levels around -85 dBm, connection continuity issues and disconnections were likely to occur. SNR levels were observed to be around 30 dB and above, indicating ideal connectivity. Channel interference emerged as the most significant issue in the existing network. This situation was particularly common in the 2.4 GHz band due to its limited number of channels, causing channel overlaps that negatively affected network performance. Despite the use of wide channels in the 5 GHz frequency band, heavy channel interference was also detected. In areas with channel interference, performance problems such as decreased network capacity, reduced SNR, packet loss, and increased latency were identified.

Based on the results, AP placements were reorganized, and access points were added to regions with signal strength between -67 dBm and -85 dBm. This arrangement brought the signal strength to the desired levels across all building floors and significantly reduced connection interruptions. Signal strength optimizations not only resolved weak signal issues but also improved low SNR and secondary signal strength values, enhancing overall network performance. In regions with severe channel interference, the 2.4 GHz band was disabled due to its limited channel structure to eliminate interference. In the 5 GHz band, the channel width was limited to 20 MHz to prevent potential interference caused by wide channel usage, and channel assignments for each access point were statically configured to be independent and non-overlapping. Each floor was upgraded to serve an average capacity of 200 end users.

Future studies may focus on the advantages of the 320 MHz wide channel structure offered by Wi-Fi 7 technology and the 6 GHz band in high-density environments. Additionally, AI-powered network management systems could be explored in areas such as channel optimization and dynamic resource allocation. The findings of this study demonstrate that proper AP placement and channel configuration have a direct impact on network performance. Strategies aimed at optimizing weak signal areas and reducing interference have increased network stability and ensured continuous connectivity. From a security perspective, a stable network design minimizes risks such as unauthorized access and data loss, contributing to the establishment of secure communication environments.

References

[1] H. Soy et al., "Kablosuz yerel alan ağlarında güncel gelişmeler: IEEE 802.11ac ile yeni nesil gigabit Wi-Fi," *J. Faculty Eng. Archit. Gazi Univ.*, vol. 28, no. 4, pp. 675-687, 2013.

[2] K. Pahlavan and P. Krishnamurthy, "Evolution and impact of Wi-Fi technology and applications: A historical perspective," *Int. J. Wirel. Inf. Netw.*, vol. 28, no. 1, pp. 3–19, Mar. 2021, doi: 10.1007/s10776-020-00501-8.

[3] R. Nazir, A. A. Laghari, K. Kumar, S. David, and M. Ali, "Survey on Wireless Network Security," *Arch. Comput. Methods Eng.*, vol. 29, no. 3, pp. 1591–1610, May 2022, doi: 10.1007/s11831-021-09631-5.

[4] A. Yılmaz and Z. Aslan, "802.11ax teknolojisinin kablosuz ağ altyapılarındaki rolü ve geleceği: Performans, güvenlik ve yenilik," *J. Anadolu Bil Vocat. School Higher Educ.*, vol. 19, no. 69, pp. 1–29, Jul. 2024, doi: 10.17932/IAU.ABMYOD.2006.005/abmyod_v19i69001.

[5] IEEE, "The evolution of Wi-Fi technology and standards," IEEE Standards Association, [Online]. Available: https://standards.ieee.org/beyond-standards/the-evolution-of-wi-fi-technology-and-standards/. Accessed: Oct. 23, 2024.

[6] S. Bytyqi and B. Jashari, "Experimental assessment of the effects of building materials on Wi-Fi signal 2.4 GHz and 5 GHz," *J. Comput. Commun.*, vol. 12, no. 05, pp. 1–10, 2024, doi: 10.4236/JCC.2024.125001.

[7] Ekahau, "How to accurately measure wall attenuation for the best Wi-Fi designs," [Online]. Available: https://www.ekahau.com/blog/how-to-measure-wall-attenuation-for-spotless-wi-fi-network-designs/#Why-You-Should-Measure-Wall-Attenuation. Accessed: Oct. 13, 2024.

[8] C. M. Own, J. Hou, and W. Tao, "Signal fuse learning method with dual bands WiFi signal measurements in indoor Positioning," *IEEE Access*, vol. 7, pp. 131805–131817, 2019, doi: 10.1109/ACCESS.2019.2940054.

[9] C. Aneke, H. Akpan Jacob, K. Constance, and A. Chikezie Samuel, "Application of Ekahau real time location software for the calibration of test beds for wireless network-base asset location management system," *J. Multidiscip. Eng. Sci. Stud.*, vol. 5, no. 8, pp. 2769-2776, Aug. 2019.

[10] N. E. Kaljahi, "Performance evaluation of Wi-Fi networks," M.S. thesis, Dept. Inf. Security Commun. Technol., Norwegian Univ. Sci. Technol., Trondheim, Norway, 2021.

[11] Ekahau, "Ekahau Site Survey & Heatmap Visualizations," Ekahau, Helsinki, Finland, Tech. Rep., 2016. [Online]. Available: https://www.ekahau.com/wp-content/uploads/2020/06/Ekahau-Site-Survey-Heatmap-Visualizations.pdf

[12] A. A. Rabbany et al., "Analisis pengaruh co-channel interference terhadap kualitas Wi-Fi pada frekuensi 2,4 GHz," *J. Komputer, Inf. Teknologi, dan Elektro*, vol. 6, no. 2, pp. 31–35, Aug. 2021.

[13] D. H. Kang, K. W. Sung, and J. Zander, "Attainable user throughput by dense Wi-Fi deployment at 5 GHz," in *Proc. IEEE Int. Symp. Personal, Indoor and Mobile Radio Commun. (PIMRC)*, London, UK, 2013, pp. 3418–3422, doi: 10.1109/PIMRC.2013.6666739.

[14] A. Zubow and R. Sombrutzki, "Adjacent channel interference in IEEE 802.11n," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Paris, France, 2012, pp. 1163–1168, doi: 10.1109/WCNC.2012.6213952.

[15] I. Dolinska, M. Jakubowski, and A. Masiukiewicz, "Interference comparison in Wi-Fi 2.4 GHz and 5 GHz bands," in *Proc. Int. Conf. Inf. Digital Technol. (IDT)*, Zilina, Slovakia, Jul. 2017, pp. 106–112, doi: 10.1109/DT.2017.8024280.

[16] Fortinet, "Channels and channel planning," Fortinet Document Library, [Online]. Available: https://docs.fortinet.com/document/fortilan-cloud/24.2.0/fortilan-cloud-wi-fi-concept-guide/15862/channels-and-channel-planning. Accessed: Oct. 16, 2024.

[17] NetAlly, "AirMagnet® Survey Pro wireless design & site survey analysis software," NetAlly, Colorado Springs, CO, USA, Tech. Rep., 2023. [Online]. Available: https://www.netally.com/wp-content/uploads/AirMagnet-Survey-Pro-Datasheet.pdf

[18] Ekahau, "Ekahau AI Pro: The most trusted solution for business-critical Wi-Fi design create fast and reliable networks with powerful AI tools and expert-level customization options," Ekahau, Helsinki, Finland, Tech. Rep., 2024. [Online]. Available: www.ekahau.com

[19] NetSpot, "Plan, survey, visualize & improve WiFi network with NetSpot," [Online]. Available: https://www.netspotapp.com/features.html. Accessed: Oct. 30, 2024.

[20] Acrylic WiFi, "Wi-Fi site survey tool site survey," Tarlogic Security, Tech. Rep., 2024. [Online]. Available: www.acrylicwifi.com

Article Information Form

Author Contributions: The article was prepared by İsmail Hakkı Cedimoğlu and Berkay Dağtaş. İsmail Hakkı Cedimoğlu contributed to the literature review and provided guidance particularly in the methodology and results sections. Berkay Dağtaş was responsible for field studies, data collection, analysis, and the drafting of the manuscript.

Artificial Intelligence Statement: ChatGPT was used solely for grammar and spelling corrections during the preparation of the article. The scientific content, analysis, and interpretation were produced entirely by the authors.

Plagiarism Statement: This article has been scanned by iThenticate.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
http://saucis.sakarya.edu.tr/
e-ISSN: 2636-8129
Publisher: Sakarya University
Vol. 8, No. 2, 184-197, 2025
DOI: 10.35377/saucis..1665478
RESEARCH ARTICLE

LungDxNet: AI-Powered Low-Dose CT Analysis for Early Lung Cancer Detection

Premananda Sahu[1], Ashwani Kumar[2], Mahesh Singh[3], Rituraj Jain[4], Kamal Upreti [5,*], Jyoti Parashar[6]

[1]Lovely Professional University, School of CSE, Jalandhar, Punjab, India, ror.org/00et6q107
[2]Bennett University, School of Computer Science Engineering & Technology, Greater Noida, UP India, ror.org/00an5hx75
[3]Aditya University, Department of ECE, Surampalem, India, ror.org/03pztks36
[4]Marwadi University, Department of Information Technology, Rajkot, Gujarat, India, ror.org/030dn1812
[5]Christ University, Department of Computer Science, Delhi NCR Campus, Ghaziabad, India, ror.org/022tv9y30
[6]Bharati Vidyapeeth's Institute of Computer Applications & Management, Department of Computer Application, Delhi, India

Corresponding author:
Kamal Upreti, School of Sciences, Department of Computer Science, Christ University, Delhi NCR Campus, Ghaziabad, India
kamalupreti1989@gmail.com

Article History:
Received: 25.03.2025
Revised: 20.04.2025
Accepted: 21.04.2025
Published Online: 13.06.2025

ABSTRACT

Early and accurate diagnosis, however, is still lacking for the most common form of lung cancer, and this remains one of the leading cancers leading to mortality. CT scans are widely used for lung cancer screening; however, their manual interpretation is time-consuming and prone to variability. This study introduces LungDxNet, a deep learning-based framework that integrates transfer learning to enhance diagnostic accuracy and efficiency. Using a large dataset of Low Dose CT (LDCT) scans, the system is built with fine-tuned pre-trained Convolutional Neural Networks (CNNs) such that feature extraction is reliable though minimal reducing radiation exposure. Consequently, LungDxNet involves the integration of component segmentation techniques that have been used to isolate the lung regions and discriminate the cancerous nodules from the malignant and benign cases. Very rigorous evaluations were performed on the model against both conventional machine learning and state of the art deep learning architectures. Results show that there is a substantial reduction of false positive and false negative resulting in a superior accuracy (98.88), sensitivity, and specificity. This design is to be scaled, robust and clinically applicable, making it a potential real world lung cancer diagnosis tool. Deep learning and transfer learning has excellent power to transform lung cancer detection, and this research brings awareness of how far we can optimise and integrate into clinical workflow. The model is enhanced for future work and adapted for real time diagnostic applications.

Keywords: Lung Cancer, Deep Learning, Machine Learning, Low Dose CT, Artificial Intelligence

1. Introduction

Lung cancer is a significant health issue and remains the primary cause of a large proportion of deaths that occur with this illness. Concerning survival, identification is among the most excellent tools because early interventions may even change the course of the disease. It is generally agreed that CT scan visioning is the best technique for lung deviations; however, there are some time-consuming with subjectively dependent manual diagnoses by radiologists. It reflects the requirement for automatic, fixed, and scalable resolutions for clinical experts as they proceed to perform their investigative activities. AI has been introduced in medicinal visioning and is proving divergent for many different domains. Deep learning, which automatically learns features from large complex datasets by generalization, has caught much attention in the current AI community. Among the CNNs belonging to the more prominent family of DL, they are capable of impressive performance on image categorization and decomposition tasks [1].

On the other hand, the system needs large amounts of explained data to train from scratch, and in much relevance, including medical, such annotated datasets are complex to come by. Transfer learning is one of the most potent strategies against these challenges above. Transfer learning assists in making use of pre-trained models, which have cultured the general aspects of gigantic data like ImageNet. Fine-tuning such models on context-specified data enables the attainment of elevated accomplishment with significantly much smaller datasets.

This research explores the possibility of utilizing DL coupled with transfer learning in the creation of a self-regulated lung cancer recognition system. The idea to be introduced here is fine-tuning pre-trained CNNs towards classifying lung low-dose

Cite as: P. Sahu, A. Kumar, M. Singh, R. Jain, K. Upreti, and J. Parashar. (2025). Lungdxnet: Ai-powered low-dose ct analysis for early lung cancer detection. *Sakarya University Journal of Computer and Information Sciences*. vol. 8, no. 2, pp. 184-197, 2025 doi: 10.35377/saucis...1665478

CT scans into cancerous and non-cancerous cases. The identification of lung cancer using deep learning relies on several key techniques to enhance accuracy and robustness. The process begins with lung region segmentation, which isolates the lung area from medical images, removing irrelevant background structures. This supports the fact that the model concentrates only on lung tissues and rises within feature extraction. Then, data augmentation deals with the problem of class imbalance by artificially growing the diversity of training instances. In this, techniques like rotation, flipping, and contrast adjustments create variations of lung images, so the model is not geared toward dominant classes, and its ability to generalize, among other cases, improves. For feature extraction, VGGNet, ResNet, and InceptionNet are used in advanced deep learning architecture. The patterns captured by these networks are very complex in medical images. However, given the deep convolutional layers of VGGNet, hierarchical features are extracted. ResNet, utilizing residual connections, ensures deeper network training without vanishing gradients, making it highly effective for complex image analysis [2]. InceptionNet, with its multi-scale feature extraction, captures fine details and structural variations in lung nodules, which is essential for differentiating between benign and malignant cases. The main aim would be to produce effective, automatically enhanced clinical workflows, thereby avoiding potential diagnostic errors and accelerating efforts made towards its detection from the earliest stages.

1.1.Motivation

Though medical imaging has seen numerous improvements, conventional methods of lung cancer diagnosis particularly those reliant on manual CT scan assessment are still largely encumbered by subjectivity, time consumption, and inter-observer variability. These constraints create serious problems in the early detection of lung cancer, where a successful diagnosis can make a difference between living and dying. Deep learning (DL) models have been shown to automate these tasks, but a lot depends on the availability of large, well-annotated datasets, which are very scarce in the medical field owing to privacy issues, labelling complexity, and expert dependency. In order to overcome these limitations, this study proposed a LungDxNet, a novel deep learning based deep learning framework that coalesces model fine tuning along with explainable AI on lung regions. This directly lowers the false positives/negatives on LungDxNet, improves model interpretability and allows the real time deployment in clinical settings while overcome from the gaps indicated in table 1.

The primary contributions of this work are as follows:

1. This study developed LungDxNet, an AI-driven system utilizing fine-tuned CNN models optimized for Low-Dose CT (LDCT) scan analysis to detect lung cancer at an early stage.
2. This study introduced a robust lung segmentation technique to isolate the lung region and enhance feature extraction, improving classification between malignant and benign nodules.
3. This study applied explainable AI techniques to improve model transparency and clinical trustworthiness.
4. This study extensively evaluated proposed model on a large LDCT dataset, demonstrating superior accuracy (98.88%) compared to traditional machine learning and recent deep learning approaches.
5. This study provides a scalable, real-time compatible solution that can be adapted for clinical lung cancer screening workflows.

This paper will be presented as follows: the review of the research gives a summary overview of the present methodologies used and highlights the areas this work seeks to bridge. Then, the methodology section extends with a presentation of the data processing pipeline, model architectures, and training strategy adopted in this work. Finally, results and discussions provide an in-depth analysis of the model's performance in relation to its clinical relevance and integration into the real-world setting.

2. Literature Review

Thus, low dose computed tomography (LDCT) has become a promising tool for early lung cancer detection with the potential to reduce mortalities [3]. Nevertheless, despite this, high false positive rates and dependence on the expertise of radiologists have limited its widespread adoption [4-6]. Artificial intelligence (AI) and machine learning (ML) techniques have integrated promise to overcome some of these challenges and improve lung cancer screening accuracy and efficiency [7-13].

AI could bring a lot to improving patient outcomes in lung cancer screening workflows [14]. With the help of AI-powered reconstruction techniques, less radiation can be given, and yet the image quality remains optimal, addressing concerns about the radiation exposure of CT scans [15]. One of the interesting uses of the application of AI algorithms, mainly deep learning (DL) models, has shown very high accuracy in the detection of lung nodules and classifies them as benign or malignant. For example, the accuracy of a 3x3 kernel convolutional neural network in lung nodule detection and classification was 97.56%, and the specificity was 98.4% [16].

Such integration of AI-driven analysis onto LDCT has multiple advantages compared to existing screening methods, including imaged and clinical data analysis and risk stratification. Computer-aided detection (CAD) systems have enhanced the automatic detection of potential lung nodules with high sensitivity and reduced the reading time for a concurrent or second reader. Furthermore, AI-based approaches facilitate the automatic segmentation and assessment of lesion size, volume, and

densitometric features, as well as radiomic feature extraction for comprehensive nodule characterisation [14].

At the same time that early-stage lung cancer detection has been increasingly popular using AI leveraging deep learning models, the use of such models for this purpose has become commonplace. P. Sathe et al. [17] used TNM (Tumor, Lymph Nodes, metastasis) classification to develop a fully automated solution for the screening of lung cancer with a performance of 96.4%. In the same vein, N. Gautam et al. [18] developed a hybrid model based on ResNet, DenseNet, and EfficientNet (LIDC-IDRI validation dataset), of which the model achieves 97.23% accuracy with 98.6% sensitivity and decreases false negatives.

Additionally, HA Ewaidat and YE Brag [19] used a CNN-based model, YOLOV5, to detect lung nodules in CT images. The model was trained on 280 annotated scans of the LIDC-IDRI dataset and achieved a mean average precision of 92.27%. PG Mikhael et al. introduced a deep learning-based lung cancer risk prediction model using LDCT imaging [20], and they achieved an AUC of 0.92 at one year, which expands on personalised screening.

Spatial Pyramid Pooling and 3D Convolution Deep Screener (Deep learning method) beat previous state-of-the-art algorithms [21] with the area under the curve (AUC) equal to 0.892. Another study introduced a Computer-Aided Detection (CADe) system utilising deep learning features and genetic algorithm optimisation, achieving a detection accuracy of 96.25%, sensitivity of 97.5%, and specificity of 95% [22]. Additionally, a 3D interpretable hierarchical semantic convolutional neural network (HSNet) demonstrated superior diagnostic performance in detecting various aspects of lung nodules, including malignancy [23].

In further studies, DL models were compared to radiologists. C. Jacobs et al. [24] compared 11 radiologists with high-performing deep learning systems on LDCT for lung cancer detection and found AUC values of 0.9 for deep learning and comparable to the radiologists. Y. Wang et al. [25] developed a radiomics-aided reinforcement learning model for early lung cancer analysis among serial LDCT scans with an AUC of 0.88, which is better than that of the model here.

Furthermore, A. Saha et al. [26] used three of the three transfer learning models to classify lung cancer-based lung cancer CT images using a multiclass (Categorisation) and achieved an accuracy of 91%. L. In LDCT imaging to detect lung nodules, Song et al. [27] used CapsNet as a feature extractor and integrated it with a 3D CNN, getting a detection rate of 95.19%. J. Shao et al. [28] performed the deep learning models' implementation across 12,360 people in screening by LDCT in China, which reached an accuracy of 86.96% in lung cancer risk assessment through using AI.

Additionally, the concept of transfer learning has been explored to improve lung cancer detection models. R. Anand et al. [29], VGG 16, and Inception V3 architecture were applied over the IQ _OThnccd lung cancer dataset, and their classification accuracy outcome was 96%. While this is, their study also showed key limitations: the necessity of a lot of labelled datasets, biases in the database, and capabilities to adapt pre-trained networks to medical problems.

Despite a great deal of progress in using AI to analyse LDCT, challenges exist. Refining AI models for lung cancer screening is necessary to mitigate high false positive rates. Furthermore, for healthcare centres, DL-based detection models are computationally demanding and present difficulties in implementing these technologies. In addition, radiologists' presence is essential to compensate for algorithmic bias and clinical validation and improve diagnostic accuracy [4].

Moving forward, it will be necessary to continue research and clinical validation of AI-powered LDCT analysis so that it can be brought into routine clinical practice. Gaps in data quality, algorithmic bias, and computational efficiency will have to be resolved in order to optimise AI for lung cancer screening. Collaboration between AI technology and human expertise could be the solution to refining the processes of lung cancer screening and improving patient outcomes [4, 30].

Additionally, the existing literature shows how much has been achieved in AI powered LDCT analysis for lung cancer detection early. Although deep learning models show notable progress, such as false positive and false negative rates, lack of dataset, being interpretable, and yet deploying in a real-time clinical environment, there remain still challenges. Addressing such gaps herein is described LungDxNet, a deep learning-based framework that incorporates transfer learning fine-tuned for pre-trained CNN architectures in enhanced diagnostic efficiency and accuracy. By leveraging advanced segmentation techniques, optimised feature extraction, and explainable AI methods, this research aims to bridge the identified gaps and provide a scalable, robust, and clinically applicable lung cancer detection solution. To systematically bridge the gaps identified in prior studies, this research formulates key research questions and addresses them through the proposed LungDxNet framework. Table 1 presents an overview of the significant research gaps, corresponding research questions, and how they are fulfilled in this study.

Having established the research gaps and their solutions, the following section details the methodology adopted to develop and evaluate the LungDxNet model, including data preprocessing, feature extraction, and classification techniques.

Table 1. Research Gaps, Corresponding Research Questions, and How They Are Addressed in This Study

Research Questions	Research Gaps	How These Questions Were Fulfilled
How can AI-powered lung cancer detection models be improved to minimise false-positive and false-negative rates?	High false-positive and false-negative rates in existing AI-powered lung cancer detection models.	LungDxNet utilises a fine-tuned CNN model and transfer learning to improve diagnostic accuracy, reducing false-positive and false-negative rates.
What strategies can be used to enhance the availability and quality of LDCT datasets for training deep learning models?	Limited availability of large, high-quality annotated LDCT datasets for deep learning model training.	The study leveraged a large LDCT dataset, applying augmentation techniques to enhance data quality and diversity.
How can AI algorithms be optimised to improve the differentiation between malignant and benign lung nodules?	Challenges in distinguishing between malignant and benign lung nodules with high accuracy.	A combination of segmentation techniques and deep learning-based classification improves differentiation between malignant and benign nodules.
What methodologies can facilitate the real-time deployment of AI-based lung cancer detection models in clinical settings?	Limited real-time deployment of AI-powered lung cancer detection models in clinical settings.	The model was designed with scalability and robustness in mind, enabling real-time diagnostic applications in clinical settings.
How can the interpretability and transparency of deep learning-based lung cancer detection models be enhanced?	Lack of interpretability and transparency in deep learning-based lung cancer detection models.	The study integrated explainable AI techniques to improve the transparency and interpretability of deep learning-based decisions.
What optimisations can be made to CNN-based feature extraction techniques to improve accuracy and efficiency?	Need for optimisation of CNN-based feature extraction techniques for better accuracy and efficiency.	Feature extraction was optimised using VGGNet, ResNet, and InceptionNet, ensuring improved accuracy y and efficiency.
How can AI models be effectively integrated with radiologists' expertise to enhance lung cancer diagnosis?	Challenges in integrating AI models with radiologists' expertise to enhance diagnostic performance.	The model's decision-making process was designed to work in conjunction with radiologists, enhancing human-AI collaboration.
What modifications can be made to deep learning models to improve their applicability in resource-constrained healthcare environments?	Limited adaptation of deep learning models for resource-constrained healthcare environments.	The proposed methodology reduces computational costs by using pre-trained models and optimising network architectures for efficiency.
How can the computational cost and processing time of deep learning-based lung cancer detection be minimised?	High computational costs and processing time are required for deep learning-based detection approaches.	By optimising network architectures and leveraging efficient training strategies, the computational demand and processing time were minimised.
What advancements in transfer learning can help enhance the performance of deep learning models trained on smaller datasets?	Need for improved transfer learning techniques to enhance the performance of deep learning models with smaller datasets.	Transfer learning techniques were fine-tuned, enabling the model to achieve high performance even with smaller datasets.

3. Methodology

This section consists of the proposed architecture in Fig.1, which describes the details of the model designed and the detailed steps expressed in a consequent manner:

1. A high-ranking medical radiologist usually conducts this test, and the input of microscopic cell images helps to detect and identify features of cancer in the primary phase.
2. The preprocessing phase describes discolouration and normalisation, which focus on the quality and consistency of the images. Noise Reduction eliminates artefacts and distortions unrelated to the actual contents of the image that may interfere with model performance, and normalisation equalises the pixel intensity values across images to ensure uniformity in the dataset.

3. Feature extraction, in this work, provides the procedure that employs deep convolutional neural networks (CNNs) to auto-extract important patterns and features present in the preprocessed images. The three well-established architectures are as follows:
    - VGGNet
    - ResNet
    - InceptionNet
4. The extracted features are fed into classification models to ascertain their likelihood of containing cancer. Three classifiers are attempted; these models map the deep features to specific classes primarily based on learned decision boundaries, and these are:
    - Fully Connected Neural Network (FCNN)
    - SoftMax Classifier
    - Support Vector Machine
5. Finally, the prediction result was produced in terms of normal, cancerous, and non-cancerous.

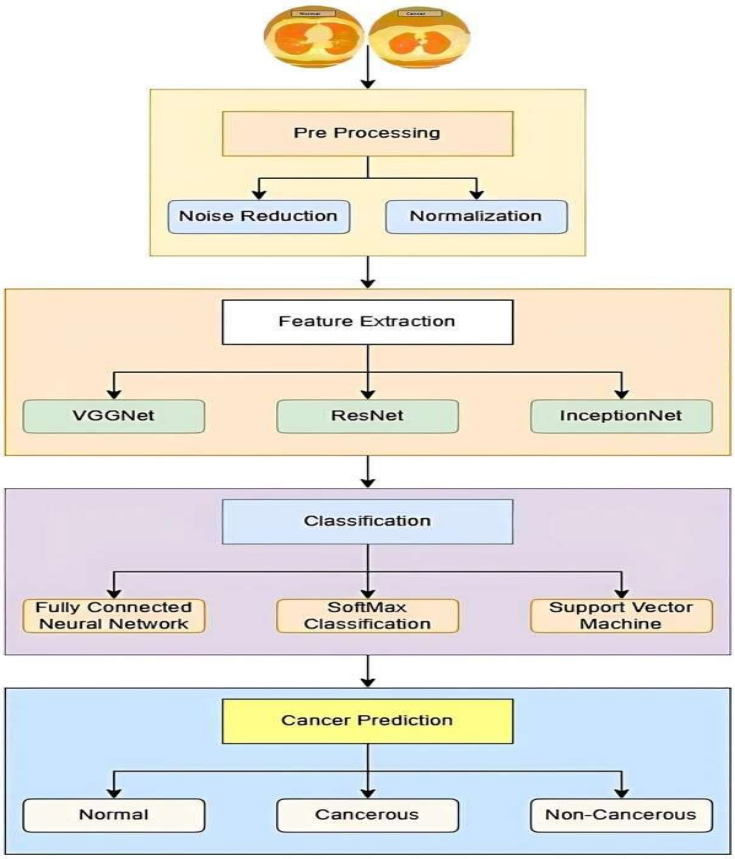


Figure 1: Complete architecture of the proposed LungDxNet framework for automatic classification of lung cancer using low-dose CT images.

3.1 Data Processing

The elimination of any disturbances and normalisation are crucial measures to progress picture excellence and model performance in the preprocessing stage of LDCT image lung cancer diagnosis. These types of images frequently cause significant levels of disturbances, which compromises image intelligibility. By regulating pixel concentration values throughout scans, normalisation improves the toughness of deep learning models. It has been done so that pixel values will be scaled onto a standard range; usually, it falls between 0 and 1 or -1 and 1, which makes the deep learning models better in terms of their convergence rate and performance. By focusing on important lung cancer attributes by means of noise lessening and normalisation, the deep learning model helps to improve nodule classification correctness and primary lung cancer diagnosis. Here, the Keras image data generator function was used.

3.2 Feature Extraction

Progressive CNN architecture is quite essential for feature removal from LDCT image-based lung cancer identification. Precise classification is made possible by these networks' instinctive learning of outlines and nodular traits at several intellectual levels. Here are some of the most predominantly utilised CNN architectures for feature removal and these are:

VGGNet (Visual Geometry Group Network)-: Applies several convolutional layers with insignificant kernel size, i.e. 3x3. Beginning from boundaries and surfaces, it hierarchically removes features till increasingly intricate lung cancer outlines. The innovative layers of VGGNet for identifying spiculated boundaries and compactness disparities play a critical role in distinguishing between normal, cancerous, and non-cancerous lungs.

ResNet—To address the issue of vanishing gradients in deep CNNs, ResNet (Residual Network) introduces skip connections, which allow for bypassing certain layers. These connections help information flow within the network without loss, ensuring that fine-grained lung nodule characteristics are well captured. Such improvement is due to Resnet's ability to extract texture, shape irregularities, and changes of contrast.

InceptionNet-: To obtain features at different scales at once, InceptionNet employs parallel convolutional layers with varying kernel sizes (1×1, 3×3, 5×5). This multi-scale approach enables the network to recognise nodules of differing sizes and densities in the LDCT images. These multi-scale approaches guarantee that both small isolated nodules and larger irregular tumours are well-defined, ensuring that all cases are effectively classified [31].

By mixing the above processes, the model proficiently learns discriminatory lung cancer-related data features, which are important to initial and precise lung cancer identification from LDCT images.

3.3 Classification

Fully Connected Neural Network (FCNN): An FCNN has individual neurons in a layer associated with each neuron in the subsequent layer. This way, a dense network of interconnections is formed. This architecture, therefore, allows the network to acquire complex patterns and relations within the data through the operation of weights throughout the training process [32]. Moreover, it shows excessive promise in supervised learning contexts such as image organisation, text classification, or in any submission where the goal is to allocate input data into one amongst a list of predefined groups. It comprises 3 main types, i.e. input, hidden and output layer.

The input layer gathers unprocessed fresh data. Every neuron in this layer relates to an input characteristic. For example, in image classification, the input may be a compressed vector of picture element strengths.

Hidden layers learn hierarchical, class-specific structures from LDCT data: The initial layers of the network learn to identify basic characteristics, including edges and textures, as well as noise patterns within low-dose medical images. The deeper layers integrate these basic features to form complex representations that capture details like nodule information. Now, the non-linear activation function ReLU enables the network to capture the lung nodules, which can be expressed as:

$$f(x)=\max(0,x) \quad (1)$$

Where f is a function, x is the input variable, and max returns the maximum value.

The output layer comprises a single neuron for each class, and the Softmax activation function translates the grooves into regularised probabilities.

For training FCNN, LDCT images with corresponding ground truth labels utilising the cross-entropy function that calculates loss function and are expressed as:

$$L = -\frac{1}{S}\sum_{j=1}^{S}\sum_{c=1}^{C} x_{j,c}\log\left(\hat{x}_{j,c}\right) \quad (2)$$

Where L is the loss function, S is the total number of samples, x_j,c is the ground truth for specified input with j sample, and class c and xˆis predicted probability.

Softmax Classifier: Let's familiarise the softmax classifier collected formerly to precisely classify lung conditions from a low-dose CT (LDCT) dataset into three classes: We will inspect how the classifier scrutinises LDCT data to dispersed normal conditions from cancerous and non-cancerous cases for lung cancer screening submissions. The softmax function stimulates the output layer to produce probability distributions over all three classes. The classification result is made by selecting the class that displays the highest probability [34]. The softmax function normalises output layer scores into probabilities during classification tasks. The FCNN establishes the neural network structure while the softmax function generates the output probabilities in this common combination. It is expressed as:

$$\text{Softmax}\,(q_c) = \frac{e^{q_c}}{\sum_k e^{q_k}} \quad (3)$$

Where q_c is the input score for class c, and k is another class.

The interior neural network framework of FCNN, where the relationship among laers is fully associated, is depicted in Fig.2.

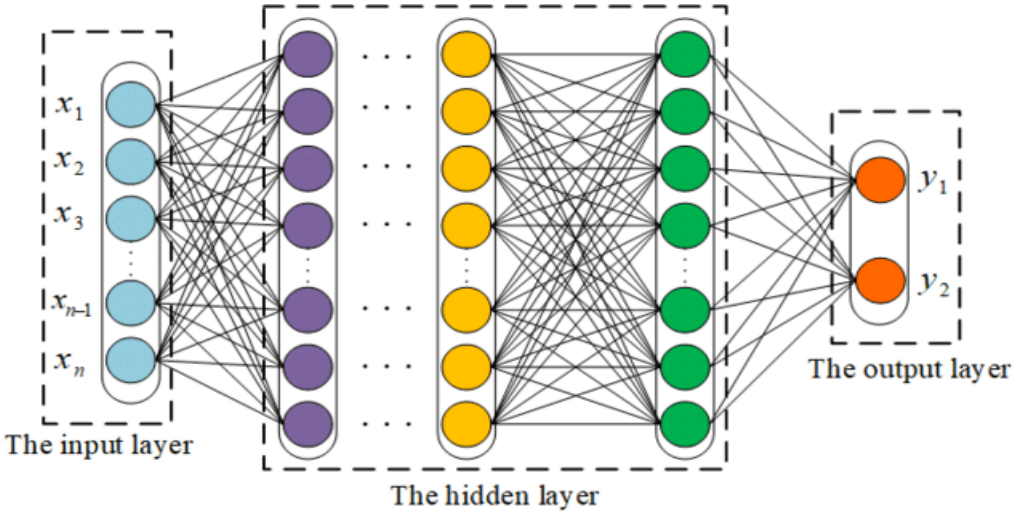


Figure 2: Architecture of the Fully Connected Neural Network (FCNN) model used for feature-based lung cancer classification in LDCT images [33]

Fig. 3 depicts the detailed softmax transformation and the FCNN pipeline for the visual flowchart.

Support Vector Machine: The system demonstrates proficiency in classification tasks by identifying the best boundary to divide distinct class data points within high-dimensional environments. In lung cancer detection, SVM uses LDCT image-derived features to accurately classify lung conditions as normal, cancerous, or non-cancerous. While neural networks train classified attributes, SVM influences predefined attributes and accomplishes well with a lesser dataset [35].

For the novel LDCT sample x, every SVM calculates the decision function expressed as:

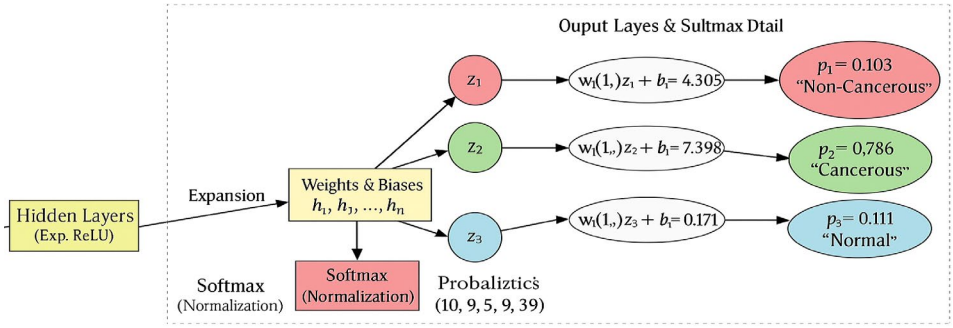


Figure 3: Complete architecture of the Fully Connected Neural Network (FCNN) pipeline used for Categorizing LDCT Lung Images into 3 Dissimilar Groups: Cancerous, Non-Cancerous, and Normal

$$f_c(x) = \sum_{j \in SV} m_j y_j k(x_j, x) + b \qquad (4)$$

Where fc (x) is the decision function with c as class and x is input, x_j is the support vector, m_j is the Lagrange multiplier with j^{th} support vector, y_j is the actual output, k is the kernel function, and b is bias.

The SVM decision-making process clearly classifies the above 3 classes based on the visual flowchart depicted in Fig.4.

It simplifies the classification of lung cancer in LDCT examinations by projecting image characteristics into an exalted dimensional space, identifying optimal hyperplanes to discriminate between normal, cancerous, and non-cancerous groups. Its benefit is in dealing with the noise inherent to LDCT and proficiently handling minor datasets, providing a dependable and explainable option associated with neural networks for the above groups.

LDCT and proficiently handling minor datasets, providing a dependable and explainable option associated with neural networks for the above groups.

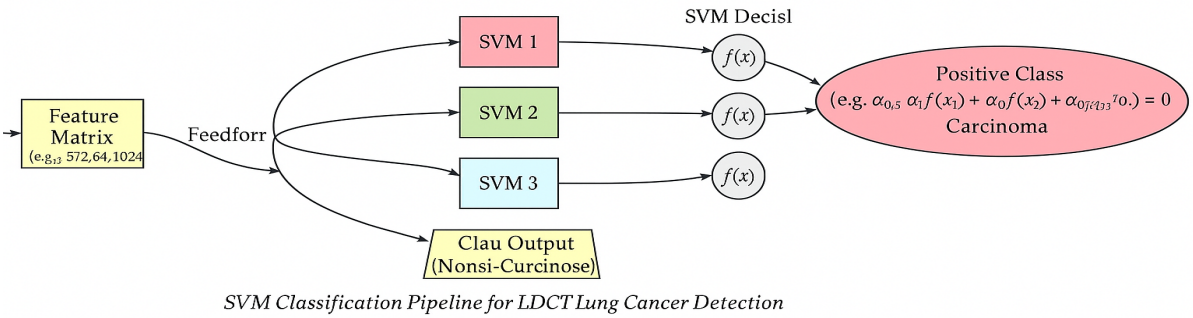


Figure 4: Support Vector Machine (SVM) pipeline for multi-class classification of lung cancer in LDCT images.

4. Result and Discussion

This section represents the implementation steps for the complete methodology for lung cancer classification, which is described below. The projected model for lung cancer classification was employed on a computer running Windows 11 OS, 64 bits as processing speed with 16GB of RAM, Google col-ab with tensor flow, and the Keras library.

4.1 Dataset

The detailed image dataset utilised to classify lung cancer using LDCT, i.e. Low Dose Computed Tomography and Projection dataset, are provided in this section [36]. By using existing lung cancer statistics and LDCT broadcast trends tailored to the Indian background. Indian lung cancer occurrence data and broadcast studies will learn these percentages. The total dataset contains 5378 LDCT images. The first class consists of non-cancerous lung cancer images (13%), comprising 592 scanned images. The second class includes 2661 cancerous lung cancer images (49%). The third class contains 2231 normal lung images (40%), all with dimensions of 512x512 pixels. In this study, 70% of the data is used for training, 20% for testing, and 10% for validation.

The assessment is done on the basis of performance matrices such as accuracy, precision, recall and f1 score. Accuracy is the description of accurately classified instances out of all cases. Recall measures the true positive occurrences classified out of actual positive occurrences [37]. Precision measures the true positive instances out of all positive instances, and they are expressed as:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \quad (5)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (6)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad (7)$$

After training from algorithms like FCNN, SoftMax classifier, and SVM, the model's training accuracy and loss, along with its validation accuracy and loss, have been determined. The performance metrics table for the above model is depicted in Table 2.

$$\text{F1 Score} = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision+Recall}} \quad (8)$$

Table 2. Accomplishment Metrices for Individual Models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
SVM	94.54	93.52	93.85	94.01
SoftMax Classifier	97.42	96.86	97.03	96.78
FCNN	**98.88**	**97.84**	**98.81**	**97.62**

The above graphical presentation and table clearly compare each other, and the results are clearly achieved. Here, the authors categorised the images into normal, cancerous, and non-cancerous, and after training, the sample image is expressed in Fig.5. The following diagram shows examples of low-dose computed tomography scans serving as screening modalities in which

both positive (likely malignant) and negative (likely benign or normal) cases are observed. Each case must include a series of axial LDCT slices, all with regions of interest highlighted and the malignancy probability score corresponding. Each row contains multiple CT slices of the same case to portray the 3D perspective of either lesion or normal anatomy. All suspicious areas are marked with red circles, while white boxes identify the region analyzed for the possibility of malignancy.

Various performance metrics for the above 3 models are described separately in Table 3, and the corresponding ROC plot is illustrated in Fig.6.

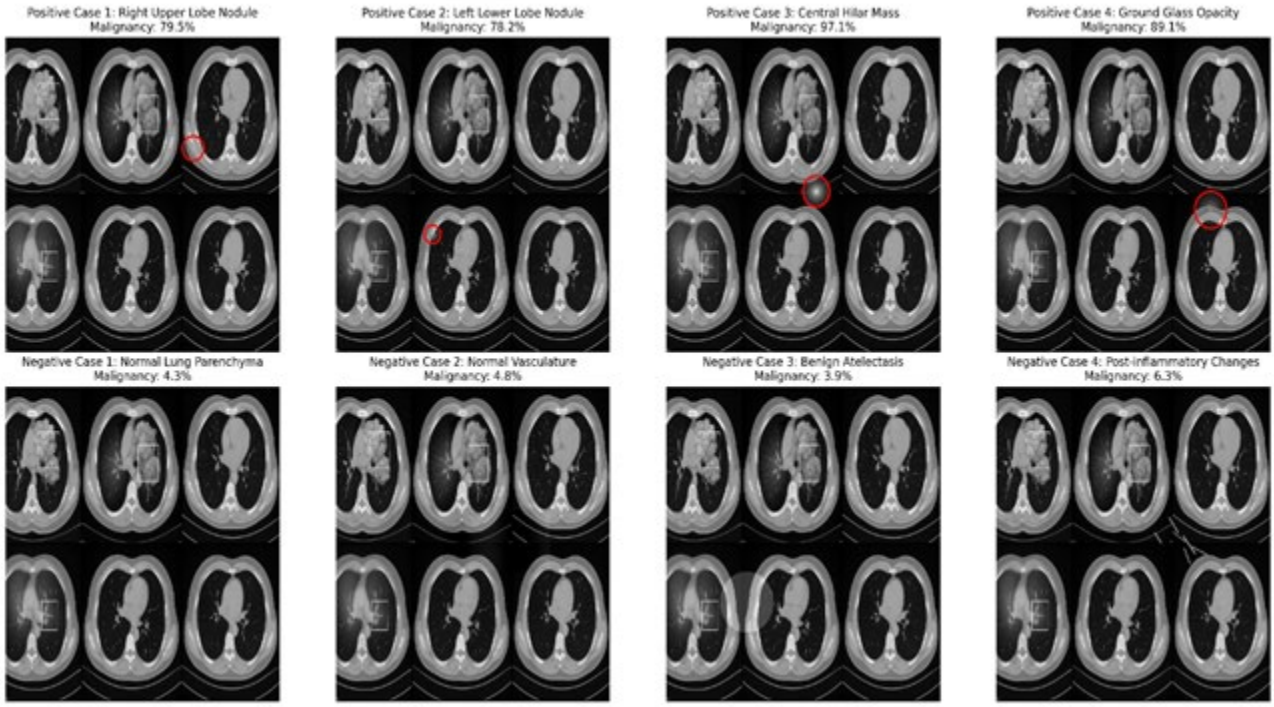


Figure 5: Visual examples of lung tumor detection using the proposed LungDxNet model, distinguishing among cancerous, non-cancerous, and normal lung CT scans.

Table 3. Multiclass Classification of All the Classifiers

Model	Class	Accuracy	Precision	Recall	F1 Score
FCNN	Non-cancerous	0.9888	1.0000	0.980	0.987
	Cancerous	0.9888	1.0000	0.990	0.984
	Normal	0.9888	0.9400	0.990	0.957
Softmax Classifier	Non-cancerous	0.9742	0.9737	0.965	0.9694
	Cancerous	0.9742	0.9286	0.946	0.9469
	Normal	0.9742	0.9615	0.952	0.9569
SVM	Non-cancerous	0.9454	0.9424	0.937	0.9396
	Cancerous	0.9454	0.9288	0.922	0.9320
	Normal	0.9454	0.9402	0.933	0.9391

Receiver Operating Characteristic curves for lung cancer detection based on the table the authors have provided need to extract the performance metrics, i.e. Accuracy, Precision, Recall, F1 Score) for each model, such as FCNN, SoftMax Classifier, and SVM, relative to the three classes: "Non-cancerous," "Cancerous," and "Normal." But ROC curves typically need the true positive rates (TPR, also known as Recall or Sensitivity) and the false positive rates (FPR) at multiple thresholds [38], [39], Neither of which the authors can obtain directly from the table.

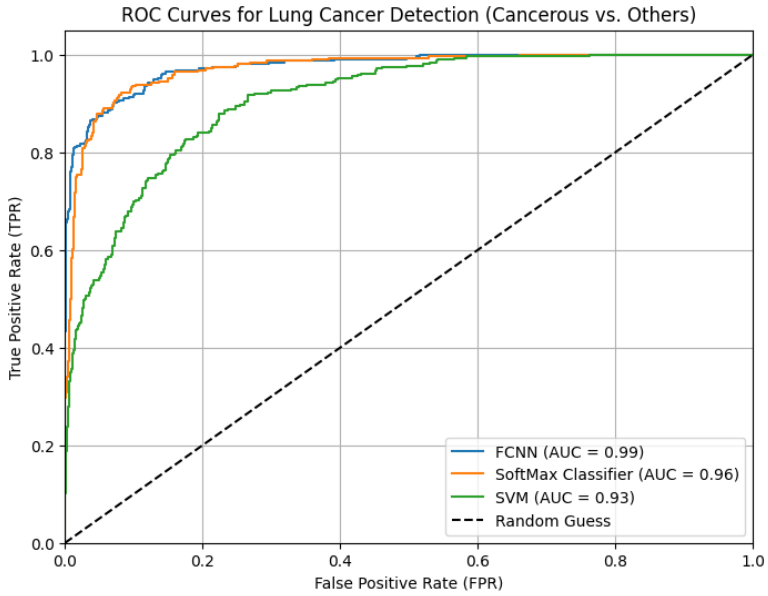


Figure 6. ROC Curve for the Assessed Classification Models, Demonstrating the Trade-off among True positive Rate (Sensitivity) and False Positive Rate (Specificity) over Diverse Threshold Values

After a training activity with the neural networks, the model would yield the values shown below on training and validation accuracies and losses, as has been proved above. Interpretation of training and validation accuracies refers to how the model has been able to perform on training databases relative to how the same model performs on validation databases [40], [41]. On the other note, training and validation losses indicate the fitting quality of the model in correlation with the training set and the validation set, respectively. Now, the training loss for all the models has been estimated for 100 epochs [42], as expressed in Fig.7.

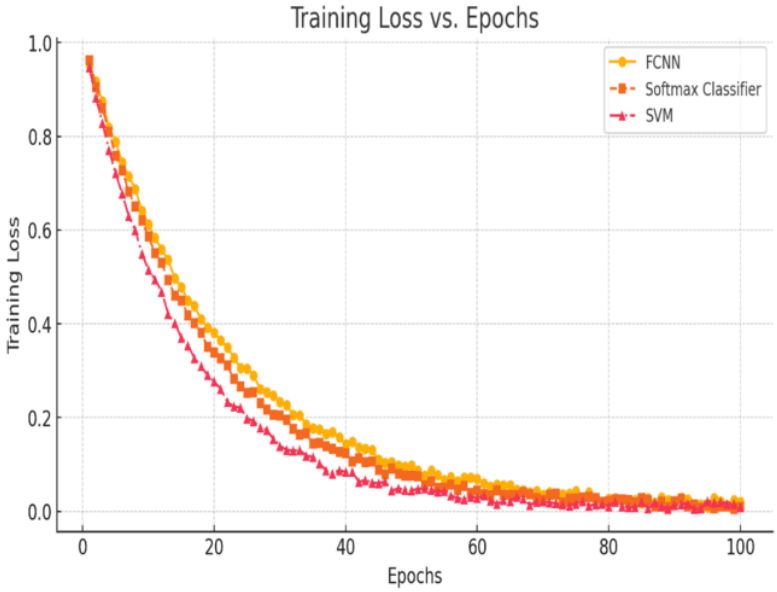


Figure 7. Training loss curves for FCNN, SoftMax, and SVM models over 100 epochs, showing model convergence behavior during lung cancer classification

Table 4 summarises the methodologies, datasets, accuracy, and key metrics from relevant literature to provide a comprehensive comparison of previous studies and the proposed LungDxNet model. This comparison highlights the strengths and limitations of existing methods and demonstrates the improvements made by the proposed approach.

Table 4. Comparison of Deep Learning Approaches for Lung Nodule Detection and Risk Assessment

Study Reference	Methodology Used	Dataset Used	Accuracy (%)	Additional Metrics
[16]	CNN with 3x3 kernel for lung nodule detection	CT scan dataset	97.56	98.4% specificity in classification
[17]	TNM classification using CNN	CT scan dataset	96.4	High specificity
[18]	ResNet, DenseNet, EfficientNet ensemble	LIDC-IDRI dataset	97.23	98.6% sensitivity, reduced false negatives
[19]	YOLOV5-based lung nodule detection	LIDC-IDRI dataset (280 scans)	92.27 (mAP)	Nodule detection precision
[20]	Deep learning-based risk prediction	LDCT-based risk assessment dataset	92.0 (AUC)	Future risk prediction capability
[21]	DeepScreener with 3D convolution	LDCT scans	89.2 (AUC)	Advanced feature extraction with 3D CNN
[22]	CADe system with genetic algorithm	LDCT scans with CADe annotations	96.25	High sensitivity (97.5%) and specificity (95%)
[24]	Comparison of DL models with radiologists	LDCT dataset (radiologist comparison)	90.2 (AUC)	Comparable performance to radiologists
[25]	Radiomics-based reinforcement learning	Serial LDCT scans	88.0 (AUC)	Superior early diagnosis prediction
[26]	Hybrid transfer learning with multiclass classification	CT scan dataset	91.0	Multiclass categorisation
[27]	3D CNN with CapsNet	Low-dose CT imaging dataset	95.19	High cancer identification rate
[28]	Mobile LDCT-based deep learning model	Mobile-based LDCT dataset	86.96	Optimised for resource-constrained settings
[29]	Transfer learning using VGG-16 and Inception V3	IQ-OTHNCCD dataset	96.0	Challenges in dataset bias and fine-tuning
Proposed LungDxNet Model	Fine-tuned CNN with transfer learning, segmentation, and optimised feature extraction	Large LDCT dataset (5378 images)	98.88	High sensitivity, specificity, and reduced false positives/negatives

Based on the comparative analysis in Table 4, it is evident that while prior studies have made significant strides in AI-driven LDCT analysis, challenges such as high false-positive rates, dataset biases, and limited real-time applicability persist. The following section details the methodology of the proposed LungDxNet model, designed to address these gaps through advanced deep learning techniques and optimised feature extraction. Figure 9 presents a comparative analysis of the accuracy achieved by various deep learning-based lung cancer detection models from the literature alongside the proposed LungDxNet model. The results indicate that while prior studies have reported notable performance improvements, the proposed method achieves the highest accuracy of 98.88%, surpassing existing approaches. This enhancement is attributed to the integration of optimised CNN architectures, transfer learning techniques, and advanced feature extraction methods. The figure visually reinforces the effectiveness of LungDxNet in minimising false positives and negatives, making it a promising tool for real-world lung cancer screening applications.

5. Conclusion

Investigation on lung cancer recognition using advanced machine learning and deep learning techniques like fully connected neural networks (FCNNs), SoftMax classifiers, and support vector machines has shown substantial advancement in classifying low-dose CT (LDCT) images into the classes of "normal," "cancerous," and "non-cancerous." The research, directed both image and text data mimicking the features of Indian LDCT, discloses excellent performance for all of the models, with an FCNN attaining the uppermost accuracy of 98.88%, precision, recall, and F1 scores, trailed by SoftMax and SVM. These consequences highlight the potential of AI-based models to advance primary detection, which is critical to improving existence in India, where growth in lung cancer occurrence can be credited to tobacco use, air pollution, and tuberculosis-related findings. Mixing these models with LDCT screening challenges the issue of late-stage analysis. It

provides a mechanism through which humanity may be condensed and the effective distribution of resources may be optimised in a resource-poor setting.

The new and unexplored area in contemporary and future scope is the application of quantum computing with AI in the area of lung cancer exposure. With quantum algorithms, feature extraction from LDCT images may be exponentially faster, allowing real-time access and analysis of vast datasets regarding genetic and environmental factors peculiar to India. This will disrupt personalised risk assessment by surpassing the current deep learning limitations dealing with noisy, low-dose CT data, probably enabling large-scale screening programs at a reasonable cost.

References

[1] B. Ozdemir, E. Aslan, and I. Pacal, "Attention enhanced inceptionNext-based hybrid deep learning model for lung cancer detection," *IEEE Access*, vol. 13, pp. 27050–27069, 2025, doi: 10.1109/ACCESS.2025.3539122.

[2] H. Rajaguru and K. Shanmugam, "Enhanced superpixel guided ResNet framework with optimized deep weighted averaging based feature fusion for lung cancer detection in histopathological images," Preprints, Feb. 2025, doi: 10.20944/preprints202502.0736.v1.

[3] M. Reck, S. Dettmer, H.-U. Kauczor, R. Kaaks, N. Reinmuth, and J. Vogel-Claussen, "Lung cancer screening with low-dose computed tomography: Current status in Germany," *Dtsch. Arztebl. Int.*, Jun. 2023, doi: 10.3238/arztebl.m2023.0099.

[4] A. Schreuder, E. T. Scholten, B. van Ginneken, and C. Jacobs, "Artificial intelligence for detection and characterisation of pulmonary nodules in lung cancer CT screening: Ready for practice?," *Transl. Lung Cancer Res.*, vol. 10, no. 5, pp. 2378–2388, May 2021, doi: 10.21037/tlcr-2020-lcs-06.

[5] A. K. Esim, H. Kaya, and V. Alcan, "Determination of malignant melanoma by analysis of variation values," *Turkish J. Eng.*, vol. 3, no. 3, pp. 120–126, Jul. 2019, doi: 10.31127/tuje.472328.

[6] M. Dirik, "Machine learning-based lung cancer diagnosis," *Turkish J. Eng.*, vol. 7, no. 4, pp. 322–330, Oct. 2023, doi: 10.31127/tuje.1180931.

[7] S. N. Polater and O. Sevli, "Deep learning based classification for alzheimer's disease detection using MRI images," *Turkish J. Eng.*, vol. 8, no. 4, pp. 729–740, Oct. 2024, doi: 10.31127/tuje.1434866.

[8] D. Maza, J. O. Ojo, and G. O. Akinlade, "A predictive machine learning framework for diabetes," *Turkish J. Eng.*, vol. 8, no. 3, pp. 583–592, Jul. 2024, doi: 10.31127/tuje.1434305.

[9] P. Kaur et al., "DELM: Deep ensemble learning model for multiclass classification of super-resolution leaf disease images," *Turkish J. Agric. For.*, vol. 47, no. 5, pp. 727–745, Oct. 2023, doi: 10.55730/1300-011X.3123.

[10] K. Meghraoui, I. Sebari, S. Bensiali, and K. A. El Kadi, "On behalf of an intelligent approach based on 3D CNN and multimodal remote sensing data for precise crop yield estimation: Case study of wheat in Morocco," *Adv. Eng. Sci.*, vol. 2, pp. 118–126, 2022.

[11] H. Ghayoomi and M. Partohaghighi, "Investigating lake drought prevention using a DRL-based method," *Eng. Appl.*, vol. 2, no. 1, pp. 49–59, 2023.

[12] H. F. Kayıran, "The function of artificial intelligence and its sub-branches in the field of health," *Eng. Appl.*, vol. 1, no. 2, pp. 99–107, 2022.

[13] E. O. Nwafor and F. O. Akintayo, "Predicting trip purposes of households in Makurdi using machine learning: A comparative analysis of decision tree, CatBoost, and XGBoost algorithms," *Eng. Appl.*, vol. 3, no. 3, pp. 260–274, 2024.

[14] M. Cellina et al., "Artificial intelligence in lung cancer screening: The future is now," *Cancers*, vol. 15, no. 17, p. 4344, Aug. 2023, doi: 10.3390/cancers15174344.

[15] R. T. Sadia, J. Chen, and J. Zhang, "CT image denoising methods for image quality improvement and radiation dose reduction," *J. Appl. Clin. Med. Phys.*, vol. 25, no. 2, Feb. 2024, doi: 10.1002/acm2.14270.

[16] R. R. Shivwanshi and N. Nirala, "Hyperparameter optimisation and development of an advanced CNN-based technique for lung nodule assessment," *Phys. Med. Biol.*, vol. 68, no. 17, p. 175038, Sep. 2023, doi: 10.1088/1361-6560/acef8c.

[17] P. Sathe, A. Mahajan, D. Patkar, and M. Verma, "End-to-end fully automated lung cancer screening system," *IEEE Access*, vol. 12, pp. 108515–108532, 2024, doi: 10.1109/ACCESS.2024.3435774.

[18] N. Gautam, A. Basu, and R. Sarkar, "Lung cancer detection from thoracic CT scans using an ensemble of deep learning models," *Neural Comput. Appl.*, vol. 36, no. 5, pp. 2459–2477, Feb. 2024, doi: 10.1007/s00521-023-09130-7.

[19] H. Al Ewaidat and Y. El Brag, "Identification of lung nodules CT scan using YOLOv5 based on convolution neural network," 2022.

[20] P. G. Mikhael et al., "Sybil: A validated deep learning model to predict future lung cancer risk from a single low-dose chest computed tomography," *J. Clin. Oncol.*, vol. 41, no. 12, pp. 2191–2200, Apr. 2023, doi: 10.1200/JCO.22.01345.

[21] J. L. Causey et al., "Spatial pyramid pooling with 3D convolution improves lung cancer detection," *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 19, no. 2, pp. 1165–1172, Mar. 2022, doi: 10.1109/TCBB.2020.3027744.

[22] A. Elnakib, H. M. Amer, and F. E. Z. Abou-Chadi, "Early lung cancer detection using deep learning optimisation," *Int. J. Online Biomed. Eng.*, vol. 16, no. 06, pp. 82–94, May 2020, doi: 10.3991/ijoe.v16i06.13657.

[23] S.-C. Hung, Y.-T. Wang, and M.-H. Tseng, "An interpretable three-dimensional artificial intelligence model for computer-aided diagnosis of lung nodules in computed tomography images," *Cancers*, vol. 15, no. 18, p. 4655, Sep. 2023, doi: 10.3390/cancers15184655.

[24] C. Jacobs et al., "Deep learning for lung cancer detection on screening CT scans: Results of a large-scale public competition and an observer study with 11 radiologists," *Radiol. Artif. Intell.*, vol. 3, no. 6, Nov. 2021, doi: 10.1148/ryai.2021210027.

[25] Y. Wang et al., "Leveraging serial low-dose CT scans in radiomics-based reinforcement learning to improve early diagnosis of lung cancer at baseline screening," *Radiol. Cardiothorac. Imaging*, vol. 6, no. 3, Jun. 2024, doi: 10.1148/ryct.230196.

[26] A. Saha, S. M. Ganie, P. K. D. Pramanik, R. K. Yadav, S. Mallik, and Z. Zhao, "VER-Net: A hybrid transfer learning model for lung cancer detection using CT scan images," *BMC Med. Imaging*, vol. 24, no. 1, p. 120, May 2024, doi: 10.1186/s12880-024-01238-z.

[27] L. Song, M. Zhang, and L. Wu, "Detection of low dose CT pulmonary nodules based on 3D CNN-CapsNet," Jun. 2023, doi: 10.22541/au.168576934.49766817/v1.

[28] J. Shao et al., "Deep learning empowers lung cancer screening based on mobile low-dose computed tomography in resource-constrained sites," *Front. Biosci. Landmark*, vol. 27, no. 7, Jul. 2022, doi: 10.31083/j.fbl2707212.

[29] R. Anand, "Lung cancer detection and prediction using deep learning," *Int. J. Eng. Appl. Sci. Technol.*, vol. 7, no. 1, pp. 313–320, May 2022, doi: 10.33564/IJEAST.2022.v07i01.048.

[30] A. R. Wahab Sait, "Lung cancer detection model using deep learning technique," *Appl. Sci.*, vol. 13, no. 22, p. 12510, Nov. 2023, doi: 10.3390/app132212510.

[31] K. Ahmed, S. S. Ahmed, A. Talukdar, and D. Chakrabarty, "An empirical study on lung cancer detection and classification using machine learning and image processing techniques," in *Adv. Intell. Syst. Comput.*, 2024, pp. 165–176, doi: 10.1007/978-3-031-75771-6_11.

[32] B. Lee et al., "Breath analysis system with convolutional neural network (CNN) for early detection of lung cancer," *Sens. Actuators B Chem.*, vol. 409, p. 135578, Jun. 2024, doi: 10.1016/j.snb.2024.135578.

[33] X. Yang, T. Ye, Q. Wang, and Z. Tao, "Diagnosis of blade icing using multiple intelligent algorithms," *Energies*, vol. 13, no. 11, p. 2975, Jun. 2020, doi: 10.3390/en13112975.

[34] U. Prasad, S. Chakravarty, and G. Mahto, "Lung cancer detection and classification using deep neural network based on hybrid metaheuristic algorithm," *Soft Comput.*, vol. 28, no. 15–16, pp. 8579–8602, Aug. 2024, doi: 10.1007/s00500-023-08845-y.

[35] N. Venkatesan, S. Pasupathy, and B. Gobinathan, "An efficient lung cancer detection using optimal SVM and improved weight based beetle swarm optimisation," *Biomed. Signal Process. Control*, vol. 88, p. 105373, Feb. 2024, doi: 10.1016/j.bspc.2023.105373.

[36] H. Liz-López, Á. A. de Sojo-Hernández, S. D'Antonio-Maceiras, M. A. Díaz-Martínez, and D. Camacho, "Deep learning innovations in the detection of lung cancer: Advances, trends, and open challenges," *Cognit. Comput.*, vol. 17, no. 2, p. 67, Apr. 2025, doi: 10.1007/s12559-025-10408-2.

[37] P. Sahu, B. Kumar Sahoo, S. Kumar Mohapatra, and P. Kumar Sarangi, "Segmentation of encephalon tumor by applying soft computing methodologies from magnetic resonance images," *Mater. Today Proc.*, vol. 80, pp. 3371–3375, 2023, doi: 10.1016/j.matpr.2021.07.255.

[38] P. Sahu, P. K. Sarangi, S. K. Mohapatra, and B. K. Sahoo, "Detection and classification of encephalon tumor using extreme learning machine learning algorithm based on deep learning method," in *Smart Innov. Syst. Technol.*, 2022, pp. 285–295, doi: 10.1007/978-981-16-8739-6_26.

[39] P. Sahu, S. Kumar Mohapatra, U. Punia, P. Kumar Sarangi, J. Mohanty, and M. Rohra, "Deep learning techniques based brain tumor detection," in *Proc. 11th Int. Conf. Reliab. Infocom Technol. Optim. (ICRITO)*, Mar. 2024, pp. 1–5, doi: 10.1109/ICRITO61523.2024.10522358.

[40] H. Dawood, M. Nawaz, M. U. Ilyas, T. Nazir, and A. Javed, "Attention-guided CenterNet deep learning approach for lung cancer detection," *Comput. Biol. Med.*, vol. 186, p. 109613, Mar. 2025, doi: 10.1016/j.compbiomed.2024.109613.

[41] A. Priya and P. Shyamala Bharathi, "SE-ResNeXt-50-CNN: A deep learning model for lung cancer classification," *Appl. Soft Comput.*, vol. 171, p. 112696, Mar. 2025, doi: 10.1016/j.asoc.2025.112696.

[42] N. Aydin Atasoy and A. Faris Abdulla Al Rahhawi, "Examining the classification performance of pre-trained capsule networks on imbalanced bone marrow cell dataset," *Int. J. Imaging Syst. Technol.*, vol. 34, no. 3, May 2024, doi: 10.1002/ima.23067.

Article Information Form

Author(s) Contributions: The authors confirm contribution to the paper as follows: Concept Design, Data Collection, Data Analysis and Interpretation, Technical Support, Critical Review, Literature Review: P. Sahu, A. Kumar, M. Singh, R. Jain, K. Upreti, J. Parashar. All authors reviewed the results and approved the final version of the manuscript.

Conflict of Interest Notice: We declare that we have no significant competing interests including financial or non-financial, professional, or personal interests interfering with the full and objective presentation of the work described in this manuscript.

Support/Supporting Organizations: The author(s) received no financial support for the research, authorship, and/or publication of this article.

Ethical Approval and Informed Consent: As no human, animal, or sensitive data were involved in this study, ethical approval was not applicable. Thus, ethical approval was not requested or needed.

Availability of data and material: Data is available on appropriate request to corresponding author.

Artificial Intelligence Statement: No artificial intelligence tools were used while writing this article.

Plagiarism Statement: This article has been scanned by iThenticate.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
http://saucis.sakarya.edu.tr/
e-ISSN: 2636-8129
Publisher: Sakarya University
Vol. 8, No. 2, 198-211, 2025
DOI: 10.35377/saucis...1671725
SAUCIS
SAKARYA UNIVERSITY
RESEARCH ARTICLE

Recruitment Model Proposal for IT Manager with SWARA, ARAS and GRA Methods

Şeyma Nur Aydın[1,*], Aşır Özbek[2], Ali Sevinç[3]

[1]Independent Researcher, Ankara,
[2]Kırıkkale University, Vocational School, Department of Computer Technologies, Kırıkkale, ror.org/01zhwwf82
[3]KOSGEB, Ankara, ror.org/025rbqk82

Corresponding author:
Şeyma Nur Aydın
Independent Researcher
seymanuraydin125@gmail.com

Article History:
Received: 08.04.2025
Accepted: 02.05.2025
Published Online: 13.06.2025

ABSTRACT
In the information age, one of the most strategic departments for modern businesses is the information technology department. Modern businesses receive the necessary support from the information technology department to keep up with the digital age and follow technological developments. Therefore, managers working in these departments must have the necessary skills in both informatics and management. In this study, a model was proposed using multi-criteria decision-making methods for the selection of an information technology manager for a food company. In the first stage of the model, criteria were determined according to the test subjects applied by the company under different titles to measure the professional and academic knowledge of the applicant candidates and the criteria were weighted with the SWARA method. In the second stage, an objective decision matrix was created by using the test scores of the candidates. In the last stage, the best candidate for the company was determined with the ARAS and Gray Relational Analysis method. The greatest contribution of the study to the literature is to show the applicability of a model that combines the objective decision matrix and subjective evaluations.

Keywords: Information Technology Manager, Multi-Criteria Decision Making, SWARA, ARAS, Grey Relational Analysis

1. Introduction

Developments in information technology have changed business structures and organizational processes. Today, every organization knows the benefits of information technologies to businesses and their ability to create competitive advantage. The current worldwide mobility shows that information technologies will continue to be an important source for the development of businesses. The intensive use of information technologies has made businesses more innovative, accelerated business processes and increased their competitive power. For this reason, information technologies are seen as a competitive tool for businesses. In the future, only organizations with technological solutions will be able to have competitive advantage [1].

Due to the increasing competition conditions, the survival of companies becomes possible by quickly adapting to new methods and information technologies. Information technologies used in companies can be summarized as computers, communication technologies, internet, robots, office automation systems, management information systems, expert systems, decision support systems, artificial intelligence and electronic data exchange systems [2]. An expert workforce is needed to use these technologies in business activities. It is extremely important for individuals working in the information technologies department to have the necessary skills in both informatics and management in order to meet the technological needs of companies and to ensure the functioning of the department.

In this study, a model for the selection of an information technology manager for a food company is proposed. A new recruitment model has been developed for the information technology department, which is one of the most strategic departments for modern businesses, by using multi-criteria decision making (MCDM) methods. The difference of the study from other recruitment models is that both an objective assessment is made during the recruitment process, and the subjective opinions of the company managers are also taken into account. In the study, the criteria weights were determined by the SWARA (Step - Wise Weight Assessment Ratio Analysis) method, while the selection of the information technology manager was determined by the ARAS (AdditiveRatioASSessment) and GRA (Grey Relational Analysis) methods and the results were compared.

Cite as: Ş. N. Aydın, A. Özbek and A. Sevinç. (2025). Recruitment model proposal for IT manager with SWARA, ARAS and GRA methods. *Sakarya University Journal of Computer and Information Sciences*. vol. 8, no. 2, pp. 198-211, 2025 doi: 10.35377/saucis...1671725

The study consists of 6 sections in total. In the second section following the introduction, general information about information technologies is presented. In the third section, a literature review is conducted on the subject and studies conducted in this field are examined. In the fourth section, the methods used in the study are explained in detail. In the fifth section, the analysis process of the study is modeled, and its findings are given. In the last section, the study results are evaluated, and suggestions are made for researchers who will conduct studies on similar subjects.

2. Information Technologies Manager

With the effect of globalization, developments in computer and communication technologies and increasing competition conditions have made it necessary for companies to use information technologies more. Today, information technologies are used in many areas such as education, health, trade, entertainment, communication and transportation. The use of information technologies brings many advantages for both individuals and businesses. In particular, the use of the most widely used internet makes life easier for individuals and quickly brings the work done to a conclusion. Information technologies are very important in creating a more secure environment for the acquisition, storage and transmission of information and for businesses to control and manage their processes more effectively and efficiently. In addition, information technologies make great contributions to companies in order to reduce the costs of businesses and increase their efficiency [3].

The main duties of IT (Information Technology) managers in private companies are to manage the company's information technology infrastructure and develop strategic solutions to increase efficiency by digitizing business processes. IT managers determine information technology policies and data security standards, and undertake the planning, installation, maintenance and optimization of the company's entire technology infrastructure (networks, servers, databases, software). In this context, they take precautions against cyber threats for data security and create risk management strategies. In addition, their job descriptions include integrating corporate applications such as ERP (Enterprise Resource Planning) and CRM (Customer Relationship Management) into the company's business processes and ensuring their effective use, training and supporting users on these systems. They automate manual processes by managing the digital transformation process and try to increase the efficiency of the company by adapting new technologies (cloud solutions, artificial intelligence, IoT) to business models.

IT managers manage the company's IT budget, ensure efficient use of resources, and evaluate the technology needs of departments by constantly communicating with business units. They work in compliance with legal regulations (e.g. KVKK - Personal Data Protection Law) and implement procedures to ensure data security and confidentiality. Thus, they support the competitive advantage of the company with multifaceted tasks such as strategic IT planning, digital transformation, and process optimization. Thanks to these efforts, the company's technological infrastructure is both secure and constantly evolving, which allows IT to create value in line with the company's general goals.

MCDM methods enable the evaluation of decision alternatives by solving decision problems that include many criteria. These methods are frequently used to solve decision problems that include many criteria in the process, such as personnel selection.

3. Literature Research

It is possible to find many different studies with different methods on personnel selection in the literature. Some of the studies conducted using multi-criteria methods are given in Table 1.

4. Methods

In this study, SWARA was used to weight the criteria, and ARAS and GRA methods were used to determine the performance of the candidates.

4.1. SWARA Method

The SWARA method, which can be translated into Turkish as "Step-by-Step Weight Assessment Ratio Analysis", was developed by Keršulienė, Zavadskas and Turskis in 2010. This method has been successfully applied to solve many MCDM problems to date. It has been widely used in many fields in recent years due to reasons such as being very suitable for working with experts and being very easy to use [30], [31], [32], [33].

In this method, the decision maker first ranks the criteria in decreasing order of importance. In the presence of more than one decision maker, each decision maker ranks the criteria in decreasing order of importance. Accordingly, the criterion ranking is obtained as many as the number of decision makers. In the group decision application, the general ranking is determined by taking the geometric mean of the criterion rankings determined by the decision makers. Based on the general ranking, the criteria are compared with the previous criterion starting from the 2nd criterion by the decision makers. Each decision maker individually compares the criteria in the general ranking. The weights of the criteria are determined according to the SWARA method after the comparisons of the decision makers. As a result, priority vectors showing the weights of the criteria as many as the number of decision makers emerge. As the last step, the geometric mean of the priority value of each criterion is taken and the final general priority values are obtained [30], [33], [34].

In the SWARA method, the required comparison rates are significantly lower compared to other flares, so the separation parts made through the survey provide much more accurate answers, allowing the SWARA method to be more accurate. In the SWARA method, the criteria can be evaluated freely without any parts [35].

The process of determining the weight of the criteria using the SWARA distribution includes the following steps [31], [35].

Table 1. Literature Research

Year	Author(s)	Method	Application Area
2024	B. Tezcan and T. Eren [4]	Defense industry project manager AHP- Pisagor Fuzzy TOPSIS	Defense industry project manager selection
2024	E. Genç, et al. [5]	Grey MAUT and Grey MOORA	Personnel selection in the tourism sector
2022	G. Elmas [6]	Fuzzy TOPSIS	Selection of sales representative for the maritime department
2021	A. Taş and P. Ç. Karataş [7]	AHP and TOPSIS	Selection of a project manager in a software company
2021	M. Popović [8]	SWARA and CoCoSo	Personnel selection
2020	E. Ayçin [9]	CRITIC and MAIRCA	Selection of personnel for the information systems department
2020	C. T. Chen and W.Z. Hung [10]	TOPSIS and PROMETHEE	Selection of overseas marketing manager
2020	A. Raj Mishra et al.[11]	IF and ARAS	Selection of information technologies personnel
2020	G. Elidolu et al. [12]	Fuzzzy AHP	Selection of the ship crew
2019	C. Erdin [13]	Fuzzy TOPSIS	Site manager selection
2019	A. Ulutaş [14]	Entropi and MABAC	Marketing manager selection
2019	B. Yıldırım et al. [15]	ARAS	Personnel selection in the aviation sector
2019	A. O. Kuşakçı et al. [16]	MULTIMOORA, AHP and TOPSIS	Selection of expert personnel in the airline company
2018	N. Akça et al. [17]	Analytical Network Process	Selection of a financial manager
2018	Y. Çelikbilek [18]	Grey AHP and MOORA	Selection of managers in the healthcare sector
2018	A. Ulutaş et al. [19]	Fuzzy AHP and Fuzzy GRA	Production planning manager selection
2018	A. Tuş and E.A. Adalı [20]	CRITIC, CODAS and PSI	Selection of marketing personnel in the textile sector
2018	D. Karabašević et al. [21]	SWARA and EDAS	Personnel selection in the information systems sector
2017	M. D. Kenger and A. Organ [22]	Entropy and ARAS	Personnel selection in the banking sector
2017	L. O. Uğur [23]	MOORA	Construction project manager selection
2015	D. Karabašević et al. [24]	SWARA and ARAS	Selection of sales managers in the telecommunications sector
2015	R. M. Alguliyev et al. [25]	Fuzzy VIKOR	Selection of information technologies personnel
2015	R. P. Kusumawardani and M. Agintiara [26]	Fuzzy AHP and TOPSIS	Human resources manager selection
2015	A. Özbek [27]	MOORA	Academic unit manager selection

SWARA Process Steps

- The criteria are ranked in decreasing order of importance. In cases where there is more than one decision maker, each decision maker ranks the criteria individually in decreasing order and an overall ranking is created by taking the geometric mean of the rankings [31], [36].
- Starting from the 2nd criterion; (j+1) criterion is compared with the jth criterion and the importance level s_j of the jth criterion is determined.
- The variable k_j, shown in equation (1), is obtained by pairwise comparison of the criteria and expresses how important the jth criterion is compared to the (j+1)th criterion.

$$k_j = \begin{cases} 1 & j = 1 \\ s_j + 1 & j > 1 \end{cases} \quad (1)$$

- The q_j variable, which shows the corrected value, is calculated as shown in Equation 2 and takes a value between 1 and 0 [37].

$$q_j = \begin{cases} 1 & j = 1 \\ \frac{q_{j-1}}{k_j} & j > 1 \end{cases} \quad (2)$$

- The relative weights w_j of the criteria are determined as shown in Equation 3.

$$w_j = \frac{q_j}{\sum_{k=1}^{n} q_k} \quad (3)$$

4.2. ARAS Method

The ARAS method is a method developed by Zavadskas and Turskis for solving MCDM problems and compares the benefit function value ratios of the decision options with the most appropriate benefit function value [33].

ARAS Process Steps [38]

- The decision matrix is created. The rows of the decision matrix represent the options, and the columns represent the criteria.
- A row of the decision matrix consisting of optimal values is placed in the matrix as the first row.
- The decision matrix is normalized using Equality 4 for benefit-oriented criteria and 5 for cost-oriented criteria.

$$\bar{x}_{ij} = \frac{x_{ij}}{\sum_{i=0}^{m} x_{ij}} \quad (4)$$

$$x_{ij} = \frac{1}{x_{ij}^*}; \qquad \bar{x}_{ij} = \frac{x_{ij}}{\sum_{i=0}^{m} x_{ij}} \quad (5)$$

- Each criterion $\bar{x}_{ij}$ of the normalized matrix is weighted by multiplying it with the corresponding criterion weight w_j as shown in Equation 6.

$$\hat{x}_{ij} = \bar{x}_{ij}\, w_j; \quad i = 0, \dots, m; \quad j = 1, \dots, n \quad (6)$$

- The optimality function value of the decision options is calculated using Equation 7.

$$S_i = \sum_{j=1}^{n} \hat{x}_{ij}; \quad i = 0, \dots, m; \quad j = 1, \dots, n \quad (7)$$

- The largest S_i value indicates the best option and the smallest S_i value indicates the worst option [38], [39].
- Equality 8 is used to calculate the benefit levels and sort them from largest to smallest.

$$K_i = \frac{S_i}{S_0}; \quad i = 0, \dots, m \quad (8)$$

4.3. Grey Relational Analysis

GRA is a method used to determine the degree of relationship between each criterion in a grey system and the reference series compared. The degree of relationship calculated as a result of the applied operations takes a value between 0 and 1 and is defined as the grey relationship degree [31].

Process steps of the GRA method [31], [40]

- The decision matrix is created. The rows of the decision matrix show the options x_i, and the performance value of the options according to each criterion is $x_i(j)$.
- The reference series is determined and placed in the first row of the decision matrix.
- The matrix is normalized according to the benefit, cost or most suitable situation of the criteria [31]. Equality (9) is used in the case of benefit, (10) in the case of cost and (11) in the most suitable situation.

$$x_i^* = \frac{x_i(j) - \min_j x_i(j)}{\max_j x_i(j) - \min_j x_i(j)} \quad (9)$$

$$x_i^* = \frac{\max_j x_i(j) - x_i(j)}{\max_j x_i(j) - \min_j x_i(j)} \quad (10)$$

$$x_i^* = \frac{|x_i(j) - x_{0b}(j)|}{\max_j x_i(j) - x_{0b}(j)} \tag{11}$$

- After the normalization process, the decision matrix is formulated as shown in Equation 12.

$$X_i^* = \begin{bmatrix} x_1^*(1) & x_1^*(2) & \cdots & x_1^*(n) \\ x_2^*(1) & x_2^*(2) & \cdots & x_2^*(n) \\ \vdots & \vdots & \ddots & \vdots \\ x_m^*(1) & x_m^*(2) & \cdots & x_m^*(n) \end{bmatrix} \tag{12}$$

- By determining the difference $\Delta_{0i}(j)$ of the absolute value between x_0^* and x_i^*, the absolute value matrix is created as formulated in Equation 13.

$$X_i^* = \begin{bmatrix} \Delta_{01}(1) & \Delta_{01}(2) & \cdots & \Delta_{01}(n) \\ \Delta_{02}(1) & \Delta_{02}(2) & \cdots & \Delta_{02}(n) \\ \vdots & \vdots & \ddots & \vdots \\ \Delta_{0m}(1) & \Delta_{0m}(2) & \cdots & \Delta_{0m}(n) \end{bmatrix} \tag{13}$$

- The grey relationship coefficient matrix is created with the help of Equation 14.

$$\gamma_{0i}(j) = \frac{\Delta \min + \zeta \Delta \max}{\Delta_{0i}(j) + \zeta \Delta \max} \tag{14}$$

The parameter ζ in Equation 14 regulates the difference between Δ_{0i} and Δ_{max} by taking a value in the range [0,1] and is called the Separating Coefficient [31].

- Grey relationship degrees are determined by Equality (15) when the criteria weights are equal, and by Equality (16) when they are different.

$$\Gamma_{0i} = \frac{1}{n}\sum_{j=1}^{n} \gamma_{0i}(j), \quad i = 1, \dots, m \tag{15}$$

$$\Gamma_{0i} = \sum_{j=1}^{n} [w_i(j)\gamma_{0i}(j)], \quad i = 1, \dots, m \tag{16}$$

Γ_{0i} shows the grey relationship degree, while w_i shows the importance degree of the ith criterion. After the grey relationship degree is calculated, it is sorted from largest to smallest. At the end of the sorting, it is determined that the option in the first place is the most suitable alternative.

5. Model, Dataset and Findings

In this study, an information technology manager selection was made for a food company operating in Turkey using the SWARA, ARAS and GRA methods in an integrated manner. The model developed for the selection process is given in the form of a flow chart in Figure 1.

Each part represents different stages of the process. This model aims to effectively evaluate IT manager candidates by providing a systematic approach.

The operation of the model is as follows:

- Determination of criteria according to test topics,
- Weighting of criteria,
  - The SWARA survey is applied to three managers of the company and the importance weights of the criteria are obtained. (General manager, chief technology officer and human resources manager)
- Receiving test scores of candidates,
- Creation of an objective decision matrix with test scores,
- Evaluation of candidates using MCDM methods.

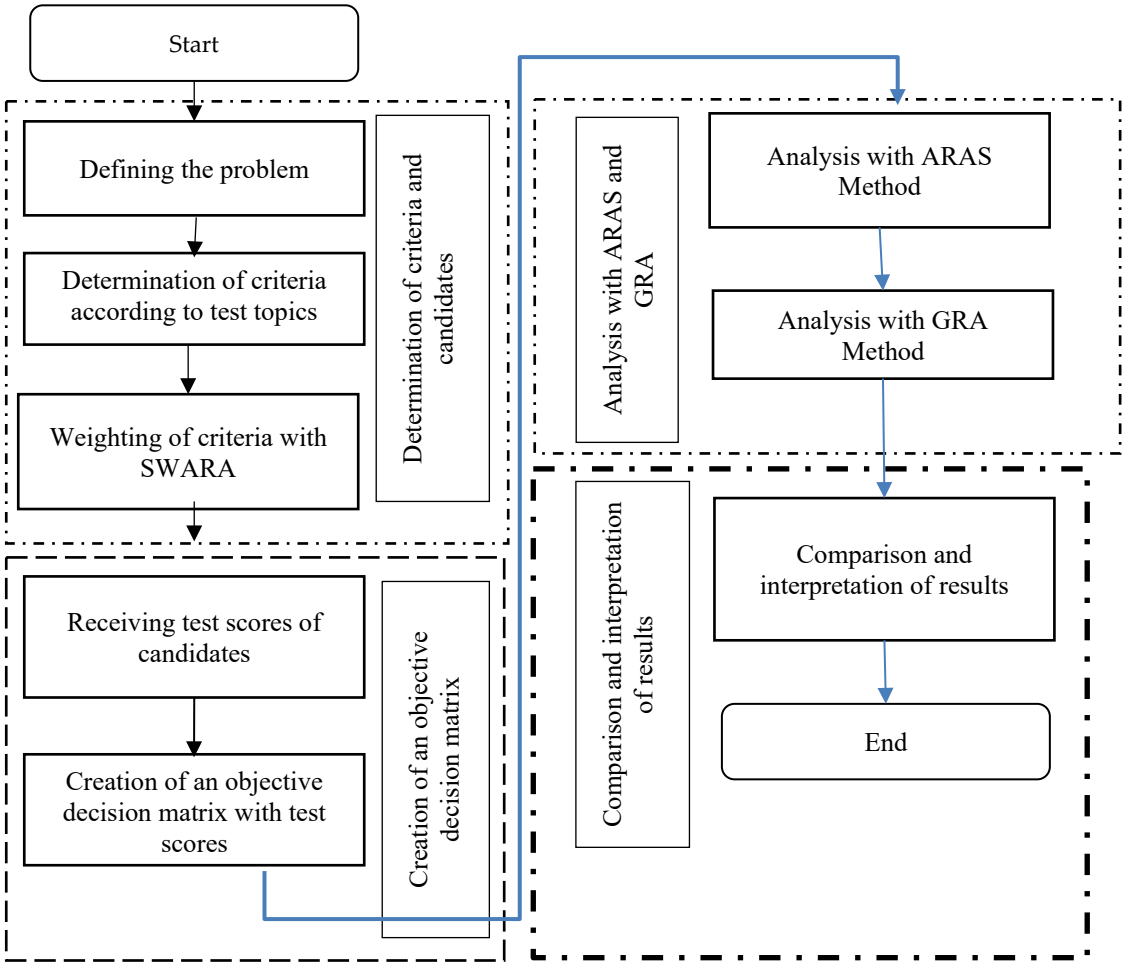


Figure 1: Proposed Model

5.1. Determination of Criteria

The criteria used in the study were determined according to the test titles applied by the company to the candidates. The criteria to be used in the analysis are given in Table 2.

Table 2. Decision Criteria

Decision Criteria	**Abbreviations**
Graduating from any of the Computer Science departments	C1
Foreign language knowledge	C2
Sectoral experience	C3
Ability to think systematically	C4
Process management skills	C5
Project management skills	C6
Linux system mastery	C7
Mastery of server operating systems	C8
Ability to manage database servers	C9
Ability to set up and manage networks	C10
Ability to set up and manage firewalls	C11
Ability to set up and manage virtual servers	C12
Ability to provide user support	C13
Monitoring and threat detection skills	C14
Interdepartmental harmony	C15
Problem solving skills	C16
Ability to follow and adapt to technological innovations	C17
Leadership	C18
Professional ethics	C19

5.2. Calculating Criteria Weights with SWARA Method

In the first stage of the study, 19 decision criteria were weighted. In the study, a subjective method, SWARA method, was used to weight the criteria. The importance of the criteria was determined by the subjective opinions of the company managers. The importance levels of the criteria were calculated by the comparisons made by the general manager, the chief technology officer and the human resources manager. The reason for this is to benefit from the opinions of the company managers for the selection of the most appropriate candidate.

In the first step of the method, three managers were asked to rank 19 criteria according to their importance levels. Table 3 shows the criteria ranking of each manager, the geometric mean of each criterion and the ranking of the criteria based on this mean, which will form the basis of the SWARA analysis.

Table 3. Ranking of Criteria According to Importance by Managers

Criterion	M1	M2	M3	Geometric Mean	Order
C1	2	1	19	3,362	3
C2	3	18	10	8,143	8
C3	1	2	9	2,621	2
C4	15	5	5	7,211	5
C5	14	6	7	8,378	9
C6	13	10	8	10,132	12
C7	12	15	15	13,925	17
C8	5	7	14	7,884	7
C9	6	8	13	8,545	10
C10	18	9	12	12,481	15
C11	7	11	16	10,720	13
C12	8	12	17	11,774	14
C13	17	16	11	14,410	18
C14	9	13	18	12,818	16
C15	19	19	2	8,971	11
C16	10	14	3	7,489	6
C17	11	4	4	5,604	4
C18	16	17	6	11,774	14
C19	4	3	1	2,289	1

Table 4. Comparison of Criteria by Managers

Criterion	Point	Order	M1	M2	M3
C19	2,289	1			
C3	2,621	2	0,45	0,20	0,2
C1	3,362	3	0,50	0,25	0,5
C17	5,604	4	0,35	0,50	0,4
C4	7,211	5	0,30	0,25	0,4
C16	7,489	6	0,25	0,20	0,4
C8	7,884	7	0,30	0,20	0,4
C2	8,143	8	0,25	0,25	0,4
C5	8,378	9	0,40	0,25	0,4
C9	8,545	10	0,35	0,10	0,4
C15	8,971	11	0,30	0,10	0,3
C6	10,132	12	0,25	0,10	0,2
C11	10,720	13	0,30	0,20	0,2
C12	11,774	14	0,25	0,10	0,2
C18	11,774	14	0,35	0,15	0,2
C10	12,481	15	0,30	0,10	0,2
C14	12,818	16	0,20	0,10	0,2
C7	13,925	17	0,2	0,10	0,2
C13	14,410	18	0,15	0,10	0,2

All criteria were ranked separately by each manager. According to Table 3, the most important criterion for the first manager was C3, for the second manager it was C1, and for the third manager it was C19. Then, the geometric means of the ranking scores given by the managers for each criterion were calculated. For example, the C19 criterion was ranked fourth for the first manager, third for the second manager, and first for the third manager. The geometric mean for the criterion was calculated as 2.289. According to this score, the C19 criterion ranked first among the other criteria.

In the second stage of the method, the criteria ranked according to their importance by taking the geometric mean, starting from the (i+1)th criterion, were compared with the previous criterion, and it was determined how much more important the ith criterion was than the (i+1)th criterion. Table 4 shows the comparisons made by the managers.

As a result of the application of the SWARA method, three different criteria weight series (w_i) were obtained. The final weights of the criteria were found by taking the geometric average of the elements with the same index of these series (Table 5).

Table 5. Final Weights of Criteria According to SWARA Method

Criterion	w_1	w_2	w_3	Arithmetic Mean	Geometric Mean
C19	0,275	0,196	0,260	0,24367	0,24109
C3	0,190	0,163	0,217	0,19000	0,18871
C1	0,127	0,131	0,145	0,13433	0,13412
C17	0,094	0,087	0,103	0,09467	0,09444
C4	0,072	0,070	0,074	0,07200	0,07198
C16	0,058	0,058	0,053	0,05633	0,05628
C8	0,044	0,048	0,038	0,04333	0,04313
C2	0,036	0,039	0,027	0,03400	0,03359
C5	0,025	0,031	0,019	0,02500	0,02451
C9	0,019	0,028	0,014	0,02033	0,01953
C15	0,014	0,026	0,011	0,01700	0,01588
C6	0,012	0,023	0,009	0,01467	0,01354
C11	0,009	0,019	0,007	0,01167	0,01062
C12	0,007	0,018	0,006	0,01033	0,00911
C18	0,005	0,015	0,005	0,00833	0,00721
C10	0,004	0,014	0,004	0,00733	0,00607
C14	0,003	0,013	0,004	0,00667	0,00538
C7	0,003	0,012	0,003	0,00600	0,00476
C13	0,002	0,010	0,002	0,00467	0,00342

When Table 5 and Figure 2 are examined together, the C19 (Professional Experience) criterion is the criterion with the highest weight among the 19 criteria. The C3 (Sectoral Experience) criterion is in second place, and the C1 (graduate from any computer science department) criterion is in third place. The C13 (Ability to provide user support) criterion has the least weight and is the criterion with the lowest level of importance.

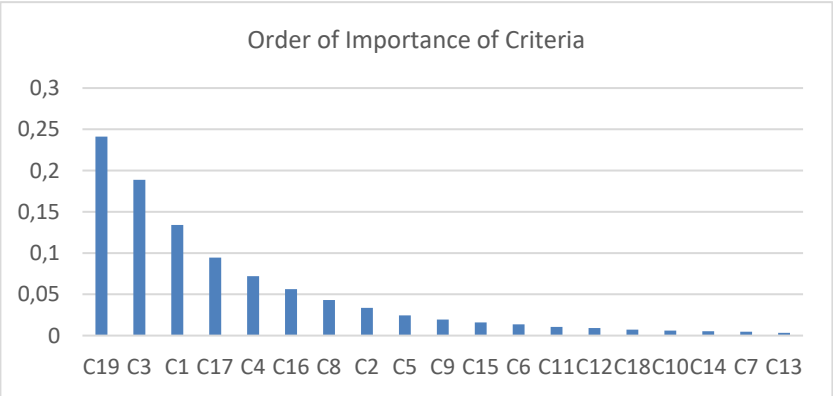


Figure 2. Order of Importance of Criteria

5.3. Selection of IT Manager with ARAS Method

In this study, MCDM methods were used to select an information technology manager for a food company. The company applied tests to 12 applicants according to certain criteria and an initial decision matrix was created based on the scores they received from the tests. Grading was done for the first three criteria in the matrix, and test scores were used for the other 16 criteria.

Among the criteria specified in Table 1, scoring for the C1 criterion was done with three grades according to the candidates' graduation (undergraduate, graduate and doctorate). Scoring for the C2 criterion was done with six grades according to the candidates' foreign language proficiency (A1, A2, B1, B2, C1, C2). Scoring for the C3 criterion was done with three grades according to their experience (1-5 years, 5-10 years, over 10). For the other 16 criteria, the scores of the answers to the test questions prepared by the company to measure the academic and professional knowledge of the candidates were used. For example, the C7 criterion is "Linux system mastery". A multiple-choice test was applied to understand the candidate's mastery of the Linux system and the scores they received are shown as the relevant criterion score in the initial decision matrix.

In the first step of the ARAS method, the initial decision matrix that will form the basis of the analysis is created. This created matrix will be considered as the initial decision matrix in other methods. For example, A1 (Candidate 1) received a total of 30 points from the test scores for the C5 criterion and 80 points from the C9 criterion. The initial decision matrix is shown in Table 6.

Table 6. Initial Decision Matrix

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19
A1	2	5	1	30	30	20	40	50	80	60	50	50	40	10	10	80	30	10	40
A2	3	5	1	20	50	20	20	20	70	40	10	80	40	90	40	30	10	40	40
A3	3	6	2	10	40	10	70	40	40	50	70	20	80	30	20	40	40	50	50
A4	2	6	1	30	20	60	40	60	50	70	70	10	30	20	60	40	20	50	80
A5	1	5	3	40	30	50	70	80	10	60	30	40	70	90	60	40	50	60	40
A6	1	4	2	40	40	60	50	20	20	10	30	50	70	70	80	50	30	50	40
A7	1	5	3	20	50	30	30	30	70	50	60	50	10	30	20	30	50	10	30
A8	1	6	3	50	60	20	30	30	60	40	10	30	20	40	30	30	40	30	20
A9	2	5	2	20	70	60	70	30	30	20	40	10	30	50	50	40	60	40	40
A10	3	6	1	30	80	60	50	40	60	50	10	20	40	50	40	40	10	20	50
A11	2	4	2	40	90	50	70	40	40	20	50	40	40	30	10	20	40	60	40
A12	2	6	3	40	50	50	70	80	50	20	40	60	30	30	40	50	60	70	50

The initial decision matrix shown in Table 6 was analyzed according to the ARAS method and the results are given in Table 7. When Table 7 is examined, it is seen that the most suitable candidate for the food business is A12. However, A2 is at the bottom of the ranking and is determined to be the candidate that the company should not prefer. The first three candidates are A12, A5 and A4. The last three are A1, A6 and A2.

Table 7. Ranking of Candidates According to the ARAS Method

	Si	Ki	Order
Optimum	0,1205	1,000	
A1	0,0655	0,544	10
A2	0,0598	0,496	12
A3	0,0761	0,631	4
A4	0,0786	0,652	3
A5	0,0793	0,658	2
A6	0,0654	0,543	11
A7	0,0670	0,556	8
A8	0,0658	0,546	9
A9	0,0727	0,603	5
A10	0,0689	0,572	7
A11	0,0711	0,590	6
A12	0,0926	0,769	1

In the ARAS application without considering the criteria weights, no significant changes were observed in the candidates' rankings compared to the application with the criteria weights. When the criteria weights were taken into account, A12 ranked first and A5 ranked second; when the criteria weights were not taken into account, A5 ranked first and A12 ranked second. In other words, there was a change of place between the two candidates. In addition, A6 ranked 11th when the criteria weights were taken into account, but ranked 6th when the criteria weights were not taken into account. Apart from these, no significant change was observed in the general ranking according to both applications.

5.4. Selection of IT Manager with Grey Relational Analysis Method

The Grey relationship coefficient matrix in Table 8 was obtained by using the initial decision matrix shown in Table 6 and Equations (9)-(14).

Table 8. Grey Relationship Coefficient Matrix

A1	0,50	0,50	0,33	0,50	0,37	0,38	0,45	0,50	1,00	0,75	0,60	0,54	0,47	0,33	0,33	1,00	0,45	0,33	0,43
A2	1,00	0,50	0,33	0,40	0,47	0,38	0,33	0,33	0,78	0,50	0,33	1,00	0,47	1,00	0,47	0,38	0,33	0,50	0,43
A3	1,00	1,00	0,50	0,33	0,41	0,33	1,00	0,43	0,47	0,60	1,00	0,37	1,00	0,40	0,37	0,43	0,56	0,60	0,50
A4	0,50	1,00	0,33	0,50	0,33	1,00	0,45	0,60	0,54	1,00	1,00	0,33	0,41	0,36	0,64	0,43	0,38	0,60	1,00
A5	0,33	0,50	1,00	0,67	0,37	0,71	1,00	1,00	0,33	0,75	0,43	0,47	0,78	1,00	0,64	0,43	0,71	0,75	0,43
A6	0,33	0,33	0,50	0,67	0,41	1,00	0,56	0,33	0,37	0,33	0,43	0,54	0,78	0,67	1,00	0,50	0,45	0,60	0,43
A7	0,33	0,50	1,00	0,40	0,47	0,45	0,38	0,38	0,78	0,60	0,75	0,54	0,33	0,40	0,37	0,38	0,71	0,33	0,38
A8	0,33	1,00	1,00	1,00	0,54	0,38	0,38	0,38	0,64	0,50	0,33	0,41	0,37	0,44	0,41	0,38	0,56	0,43	0,33
A9	0,50	0,50	0,50	0,40	0,64	1,00	1,00	0,38	0,41	0,38	0,50	0,33	0,41	0,50	0,54	0,43	1,00	0,50	0,43
A10	1,00	1,00	0,33	0,50	0,78	1,00	0,56	0,43	0,64	0,60	0,33	0,37	0,47	0,50	0,47	0,43	0,33	0,38	0,50
A11	0,50	0,33	0,50	0,67	1,00	0,71	1,00	0,43	0,47	0,38	0,60	0,47	0,47	0,40	0,33	0,33	0,56	0,75	0,43
A12	0,50	1,00	1,00	0,67	0,47	0,71	1,00	1,00	0,54	0,38	0,50	0,64	0,41	0,40	0,47	0,50	1,00	1,00	0,50

After the Grey Relationship Coefficient Matrix was created, the Grey Relationship degrees showing the ranking of the candidates were obtained using Equality (16) and are given in Table 9. The graphical representation of the ranking of the candidates is given in Figure 3.

Table 9. Ranking of Candidates According to the GRA Method

	Degree	**Order**
A1	0,4697	11
A2	0,4800	10
A3	0,5635	5
A4	0,5989	3
A5	0,6031	2
A6	0,4584	12
A7	0,5368	6
A8	0,5654	4
A9	0,5139	8
A10	0,5338	7
A11	0,4890	9
A12	0,6916	1

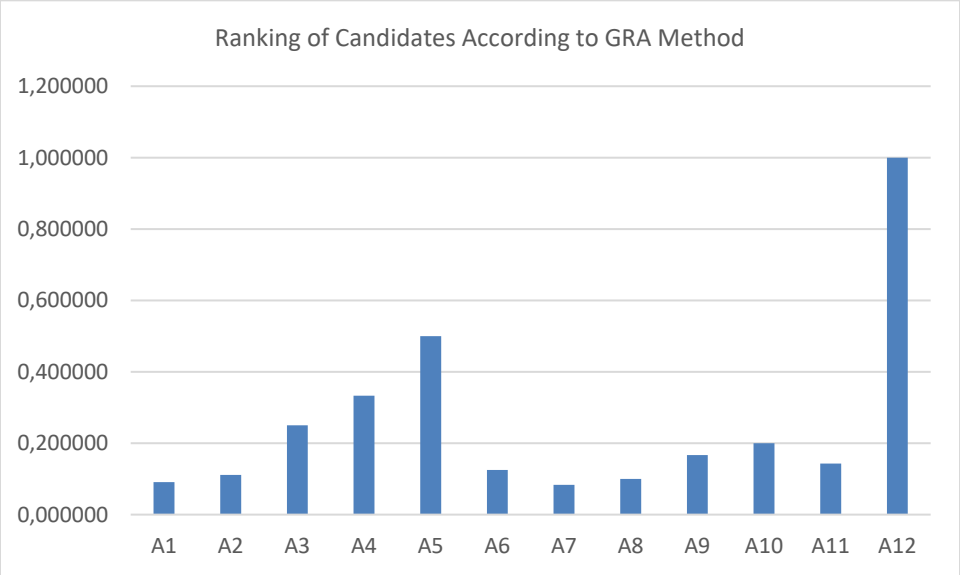


Figure 3. Ranking of Candidates According to GRA Method

The initial decision matrix shown in Table 6 was analyzed according to the GRA method and the results are given in Table 9. When Table 9 and Figure 3 are examined together, it is seen that the most suitable candidate for the food business is A12. However, A6 is at the bottom of the ranking and is determined to be the candidate that the company should not prefer. The first three candidates are A12, A5 and A4. The last three are A2, A1 and A6.

In the GRA application without considering the criteria weights, no significant changes were observed in the candidates' rankings compared to the application with the criteria weights. In the application with the criteria weights taken into account, A6 ranked 12th, A7 ranked 6th and A8 ranked 4th; in the application without considering the criteria weights, A6 ranked 8th, A7 ranked 12th and A8 ranked 10th. Apart from these, no significant changes were observed in the general ranking according to both applications.

When the applications made according to ARAS and GRA methods were compared (Figure 4), it was seen that the candidates' rankings were very similar. Only very small changes were observed in the candidates' rankings. The rankings of the first three candidates according to ARAS and GRA methods did not change. According to the ARAS method, A3 was in fourth place, while according to GRA method, A8 was in fifth place. According to the ARAS method, A9 was in fifth place, while according to the GRA method, A3 was in fifth place. According to the ARAS method, the last three places were A1, A6 and A2, while according to the GRA method, A2, A1 and A6 were in third place.

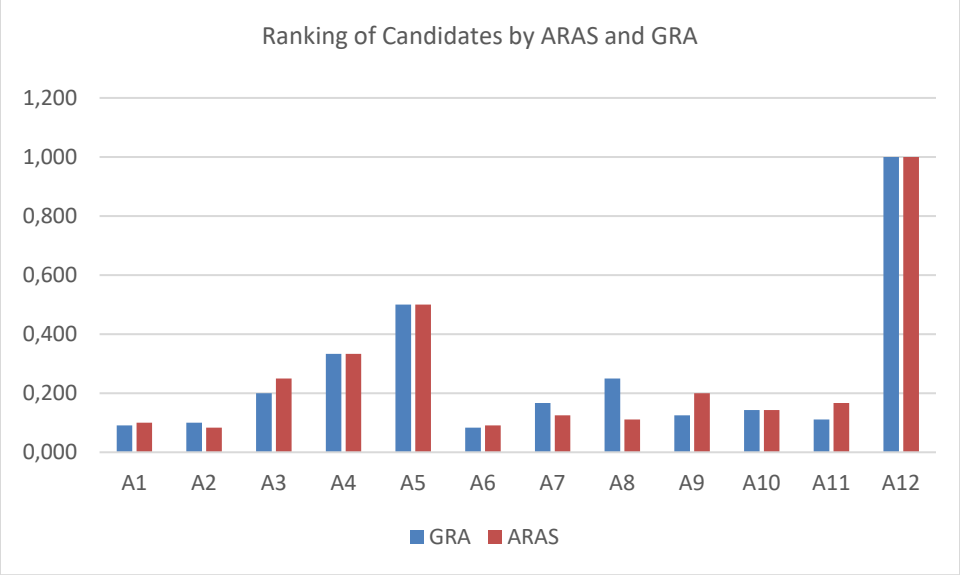


Figure 4. Ranking of Candidates According to ARAS and GRA Methods

6. Results

In recent years, the most important way for businesses to gain competitive advantage is to adapt to the requirements of the digital age and use technology accordingly. Information technologies make numerous contributions to the acceleration of business processes, increased efficiency and reduced operating costs. The most important task in this process falls on the information technology manager. For this reason, managers who will work in the information technology department should have both professional equipment and management skills.

In this study, 12 candidates were evaluated according to 19 criteria to be employed as an information technology manager in the information systems department of a company operating in the food sector. Since many criteria that may affect the decision are taken into consideration during the evaluation process, the most suitable candidate was determined by using MCDM methods. A new recruitment model was developed for the food company by using SWARA, ARAS and GRA methods in an integrated manner.

In the first stage of the model, criteria were determined according to the test subjects that the company applied under different headings to measure the professional and academic knowledge of the applicants and the criteria were weighted by SWARA method. According to the findings obtained with this method, the criterion with the highest weight among the 19 criteria was K19 (Professional ethics). The second criterion was K3 (Sectoral experience) and the third criterion was K1 (Graduating from any of the Computer Science departments). The criterion with the least weight and the lowest level of importance was K13 (Ability to provide user support).

In the next stage, an objective decision matrix was created by utilizing the test scores of the candidates and the best candidate was determined by ARAS and GRA methods. According to the analysis conducted using the ARAS method, A12 was found to be the most suitable candidate for the food business. However, A2 was at the bottom of the ranking and was determined as the candidate that the company should not prefer. The first three candidates were A12, A5 and A4. The last three places were A1, A6 and A2. There was no significant change in the ranking of the candidates in the ARAS application without taking into account the criteria weights compared to the application with the criteria weights. When criterion weights were taken into account, A12 ranked first and A5 ranked second; when criterion weights were not taken into account, A5 ranked first and A12 ranked second. In other words, there was a change in ranking between the two candidates. In addition, A6 ranked 11th when criterion weights were taken into account, while it ranked 6th when criterion weights were not taken into account. Apart from these, no significant change was observed in the overall ranking according to both applications.

According to the GRA method, A12 was found to be the most suitable candidate for the food business. However, A6 was at the bottom of the ranking and was determined as the candidate that the company should not prefer. The first three candidates were A12, A5 and A4. The last three places were A2, A1 and A6. There was no significant change in the ranking of the candidates in the GRA application made without taking into account the criteria weights compared to the application made with the criteria weights. In the application with criterion weights, A6 ranked 12th, A7 6th and A8 4th; in the application without criterion weights, A6 8th, A7 12th and A8 10th. Apart from these, there was no significant change in the overall ranking according to both applications.

When the applications made according to the ARAS and GRA methods were compared, it was observed that the ranking of the candidates was very similar. Only very small changes were observed in the ranking of the candidates. According to ARAS and GRA methods, the ranking of the first three candidates did not change. A3 ranked fourth according to the ARAS method and A8 ranked fifth according to the GRA method. According to the ARAS method, A9 ranked fifth, while A3 ranked fifth according to the GRA method. According to the ARAS method, A1, A6 and A2 took the last three places, while A2, A1 and A6 took the third place according to the GRA method.

The major contribution of the study to the literature is to demonstrate the applicability of a model that combines an objective decision matrix with subjective evaluations. A recruitment model that includes a scoring system that measures the professional and academic knowledge of candidates, and the subjective evaluations of managers can guide firms to identify the best candidate. However, since there are very few studies in the national literature in which SWARA-ARAS-GRA methods are applied in an integrated manner, it is thought that the study will contribute to the literature in this regard. Future researchers can apply the proposed recruitment model with different MCDM methods.

References

[1] M. Berisha and D. Konxheli, "Information technology as a factor of creating and developing of competitive advantages of businesses," in *Proc. Regional Sci. Conf. Int. Participation*, 2012, p. 87.

[2] H. Şahin and B. Topal, "The effect of the use of information technologies in businesses on cost and financial performance," *Int. J. Eng. Innov. Res.*, vol. 5, no. 6, pp. 394-402, 2016.

[3] P. Küsbeci, "Use of information technologies in businesses," *Uluslararası Sosyal Bilimler Dergisi*, vol. 7, no. 28, pp. 47-58, 2023.

[4] B. Tezcan and T. Eren, "Bulanık ortamda proje yöneticisi seçimi: Savunma sanayi firmasında bir uygulama," *SAVSAD Savunma ve Savaş Araştırmaları Dergisi*, vol. 34, no. 1, pp. 153-168, 2024.

[5] E. Genç, M. K. Keleş, and A. Özdağoğlu, "A hybrid MCDM model for personnel selection based on a novel Gray AHP, Gray MOORA and Gray MAUT methods in terms of business management: An application in the tourism sector," *J. Decision Analytics Intell. Comput.*, vol. 4, no. 1, pp. 263-284, 2024.

[6] G. Elmas, "Bulanık TOPSIS yöntemi ile personel seçimi: Bir freight forwarder şirketinde uygulama," *Avrupa Bilim ve Teknoloji Dergisi*, no. 35, pp. 595-602, 2022.

[7] A. Taş and P. Ç. Karataş, "Yazılım sektöründe nitelikli personel seçiminin nötrosofik AHP ve TOPSIS yöntemleri ile incelenmesi," *İşletme Araştırmaları Dergisi*, vol. 13, no. 1, pp. 969-979, 2021.

[8] M. Popović, "An MCDM approach for personnel selection using the CoCoSo method," *J. Process Manage. New Technol.*, vol. 9, no. 3-4, pp. 78-88, 2021.

[9] E. Ayçin, "Personel seçim sürecinde CRITIC ve MAIRCA yöntemlerinin kullanılması," *İşletme*, vol. 1, no. 1, pp. 1-12, 2020.

[10] C. T. Chen and W. Z. Hung, "A two-phase model for personnel selection based on multi-type fuzzy information," *Mathematics*, vol. 8, no. 10, p. 1703, 2020.

[11] A. R. Mishra, G. Sisodia, K. R. Pardasani, and K. Sharma, "Multi-criteria IT personnel selection on intuitionistic fuzzy information measures and ARAS methodology," *Iranian J. Fuzzy Syst.*, vol. 17, no. 4, pp. 55-68, 2020.

[12] G. Elidolu, T. Uyanık, and Y. Arslanoğlu, "Seafarer personnel selection with fuzzy AHP," in *Proc. 5th Int. Mediterranean Sci. Eng. Congr.*, Antalya, 2020, pp. 632-636.

[13] C. Erdin, "Bulanık TOPSIS yöntemiyle yönetici seçimi," *Yıldız Sosyal Bilimler Enstitüsü Dergisi*, vol. 3, no. 1, pp. 37-50, 2019.

[14] A. Ulutaş, "Entropi ve MABAC yöntemleri ile personel seçimi," *OPUS Int. J. Society Res.*, vol. 13, no. 19, pp. 1552-1573, 2019.

[15] B. I. Yıldırım, F. Uysal, and A. Ilgaz, "Havayolu işletmelerinde personel seçimi: Aras yöntemi ile bir uygulama," *Süleyman Demirel Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, vol. 2, no. 33, pp. 219-231, 2019.

[16] A. O. Kuşakçı, B. Ayvaz, S. Öztürk, and F. Sofu, "Bulanık MULTIMOORA ile personel seçimi: Havacılık sektöründe bir uygulama," *Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi*, vol. 8, no. 1, pp. 96-110, 2019.

[17] N. Akça, S. Sönmez, Ş. Gür, A. Yılmaz, and T. Eren, "Kamu hastanelerinde analitik ağ süreci yöntemi ile finans yöneticisi seçimi," *Optimum Ekonomi ve Yönetim Bilimleri Dergisi*, vol. 5, no. 2, pp. 133-146, 2018.

[18] Y. Çelikbilek, "Personel seçimi için bütünleşik gri AHP–MOORA yaklaşımının kullanılması: Sağlık sektöründe yönetici seçimi üzerine bir uygulama," *Alphanumeric J.*, vol. 6, no. 1, pp. 69-82, 2018.

[19] A. Ulutaş, "Entropi ve MABAC yöntemleri ile personel seçimi," *OPUS Int. J. Society Res.*, vol. 13, no. 19, pp. 1552-1573, 2019.

[20] A. Tuş and E. A. Adalı, "Personnel assessment with CODAS and PSI methods," *Alphanumeric J.*, vol. 6, no. 2, pp. 243-256, 2018.

[21] D. Karabašević, E. K. Zavadskas, D. Stanujkic, G. Popovic, and M. Brzakovic, "An approach to personnel selection in the IT industry based on the EDAS method," *Transformations Bus. Econ.*, vol. 17, no. 2, pp. 54-65, 2018.

[22] M. D. Kenger and A. Organ, "Banka personel seçiminin çok kriterli karar verme yöntemlerinden entropi temelli Aras yöntemi ile değerlendirilmesi," *Adnan Menderes Üniversitesi, Sosyal Bilimler Enstitüsü Dergisi*, vol. 4, no. 4, pp. 152-170, 2017.

[23] L. O. Uğur, "MOORA optimizasyon yaklaşımı ile inşaat proje müdürü seçimi: Çok kriterli bir karar verme uygulaması," *Politeknik Dergisi*, vol. 20, no. 3, pp. 717-723, 2017.

[24] D. Karabašević, D. Stanujkić, and S. Urošević, "The MCDM model for personnel selection based on SWARA and ARAS methods," *Management*, vol. 20, no. 77, pp. 43-52, 2015.

[25] R. M. Alguliyev, R. M. Alguliyev, and R. S. Mahmudova, "Multicriteria personnel selection by the modified fuzzy VIKOR method," *Scientific World J.*, vol. 1, p. 612767, 2015.

[26] R. P. Kusumawardani and M. Agintiara, "Application of fuzzy AHP-TOPSIS method for decision making in human resource manager selection process," *Procedia Comput. Sci.*, vol. 72, pp. 638-646, 2015.

[27] A. Özbek, "Akademik birim yöneticilerinin MOORA yöntemiyle seçilmesi: Kırıkkale üzerine bir uygulama," *Erciyes Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, vol. 1, no. 38, pp. 1-18, 2015.

[28] A. Özbek, "Yöneticilerin çok kriterli karar verme yöntemi ile belirlenmesi," *Yönetim ve Ekonomi Araştırmaları Dergisi*, vol. 12, no. 24, pp. 209-225, 2014, doi: 10.11611/JMER314.

[29] J. Varajão and M. M. Cruz-Cunha, "Using AHP and the IPMA competence baseline in the project managers selection process," *Int. J. Prod. Res.*, vol. 51, no. 11, pp. 3342-3354, 2013.

[30] V. Keršulienė and Z. Turskis, "Integrated fuzzy multiple criteria decision making model for architect selection," *Technol. Econ. Dev. Econ.*, vol. 17, no. 4, pp. 645-666, 2011.

[31] A. Özbek, *Çok Kriterli Karar Verme Yöntemleri ve Excel ile Problem Çözümü*, 3rd ed. Ankara, Turkey: Seçkin Yayıncılık, 2021.

[32] A. Özbek and E. Erol, "AHS ve SWARA yöntemleri ile yem sektöründe iş sağlığı ve güvenliği kriterlerinin ağırlıklandırılması," *AKÜ İktisadi Ve İdari Bilimler Fakültesi Dergisi*, vol. 20, no. 2, pp. 51-66, 2018.

[33] A. Özbek, "BİST'te işlem gören faktoring şirketlerinin mali yapılarının çok ölçütlü karar verme yöntemleri ile değerlendirilmesi," *Manisa Celal Bayar Üniversitesi İ.İ.B.F Yönetim ve Ekonomi Dergisi*, vol. 25, no. 1, pp. 29-53, 2018.

[34] S. H. Zolfani, E. K. Zavadskas, and Z. Turskis, "Design of products with both international and local perspectives based on Yin-Yang balance theory and SWARA method," *Econ. Res.-Ekonomska Istraživanja*, vol. 26, no. 2, pp. 153-166, 2013.

[35] D. Stanujkic, D. Karabasevic, and E. K. Zavadskas, "A framework for the selection of a packaging design based on the SWARA method," *Eng. Econ.*, vol. 26, no. 2, pp. 181-187, 2015.

[36] A. Ruzgys, R. Volvačiovas, Č. Ignatavičius, and Z. Turskis, "Integrated evaluation of external wall insulation in residential buildings using SWARA-TODIM MCDM method," *J. Civil Eng. Manage.*, vol. 20, no. 1, pp. 103-110, 2014.

[37] H. Yurdoğlu and N. Kundakcı, "SWARA ve WASPAS yöntemleri ile sunucu seçimi," *Balıkesir Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, vol. 20, no. 38, pp. 253-269, 2017.

[38] E. K. Zavadskas and Z. Turskis, "A new additive ratio assessment (ARAS) method in multicriteria decision-making," *Technol. Econ. Dev. Econ.*, vol. 16, no. 2, pp. 159-172, 2010.

[39] A. Özbek and E. Erol, "Ranking of factoring companies in accordance with ARAS and COPRAS methods," *Int. J. Acad. Res. Account. Finance Manage. Sci.*, vol. 7, no. 2, pp. 105-116, 2017.

[40] H. H. Wu, "A comparative study of using grey relational analysis in multiple attribute decision making problems," *Quality Eng.*, vol. 15, no. 2, pp. 209-217, 2002.

Author(s) Contributions: Şeyma Nur Aydın contributed to the design of the original model, data collection, construction of the matrix, data analysis and interpretation, writing, and literature review. Aşır Özbek contributed to the data analysis and interpretation, writing, and literature review. Ali Sevinç contributed to the data collection, writing, and literature review.

Conflict of Interest Disclosure: There is no conflict of interest.

Ethical Approval and Informed Consent: The test scores required for the objective decision matrix in the study were obtained from the company as secondary data under confidentiality. The data obtained from the company does not contain any information about the candidates. Only test scores were used. In addition, the comparisons made by the managers for the calculation of the criteria weights are given as a table in the study. Scientific and ethical principles have been followed and all sources utilized are shown in the bibliography section.

Artificial Intelligence Statement: No artificial intelligence tools were used while writing this article.

Plagiarism Statement: This article has been reviewed by iThenticate™.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
http://saucis.sakarya.edu.tr/
e-ISSN: 2636-8129
Publisher: Sakarya University
Vol. 8, No. 2, 212-222, 2025
DOI: 10.35377/saucis...1579599
RESEARCH ARTICLE

Enhanced Oil and Gas Production Forecasting Through Stacked generalization Ensemble Learning Technique

Gülüzar Çit [1], Azhar Alyahya [2,*]

[1]Department of Software Engineering, Sakarya University, Sakarya, Türkiye, ror.org/04ttnw109
[2]Department of Software Engineering, Sakarya University, Sakarya, Türkiye, ror.org/04ttnw109

Corresponding author:
Azhar Alyahya, Sakarya University, Department of Software Engineering
azhar.alyahya@ogr.sakarya.edu.tr

Article History:
Received: 05.11.2024
Revised: 14.04.2025
Accepted: 21.04.2025
Published Online: 13.06.2025

ABSTRACT
Planning a strategy throughout the oil and gas sector depends on production forecasting. Precise projections aid in estimating future output rates, streamlining processes, and effectively allocating resources. Techniques like " Decline Curve Analysis (DCA) and Numerical Reservoir Simulation (NRS) " have been used in the past, but they have drawbacks such reliance on static models and time consumption. A stacked generalization ensemble learning method for predicting oil and gas production is presented in this work. Using Python and data from wells in the state of "New York State", the model contains four machine learning techniques: " Random Forest Regressor (RFR), Extremely Randomized Trees Regressor (ETR), K-Nearest Neighbors (KNN), and Gradient Boosting Regressor (GBR) ". The stacked model works better than separate models, according to the results of experiments, via R2 scores of 0.9709 per oil and 0.9998 per gas.

Keywords: Machine learning models, Random Forest regressor, Extremely Randomized Trees Regressor, K-Nearest Neighbors, Gradient boosting regressor, Stacking model

1. Introduction

Many transportation systems rely on oil as their primary energy source, including vehicles, aircraft, ships, and other machinery [1]. Oil also plays an important role in various industrial uses. Crude oil is extracted from wells and refined into petroleum products suitable for consumption as part of the oil production process. The exploratory, extraction, and distributional phases make up this all-encompassing process [2]. Predicting production is essential in oil field development since it helps with economic evaluations, scheduling drilling operations, and designing facility capacity. Therefore, accurate production predictions using data from both operational and dormant wells are in great demand [3]. In order to create efficient economic planning, production forecasting is crucial for businesses and governments alike [4]. Complex numerical reservoir simulations (NRS) and comprehensive engineering evaluations are usual tools for oil and gas forecasting[5]. If oil reservoirs are to be monitored and optimized efficiently, these forecasts must be quite accurate. When it comes to calculating reservoir output, the petroleum sector typically uses conventional approaches like " decline curve analysis (DCA) and numerical reservoir simulations (NRS) " [6]. However, due to intricate static models and numerous dynamic parameters, numerical reservoir simulation models can be difficult and time-consuming. [7]. Conversely, decline curve analysis has many applications but also requires a lot of time and computing power [8]. One potential solution to these problems is machine learning, which can now anticipate oil and gas production more accurately and more quickly than ever before [9]. To efficiently and accurately predict future output, machine learning models use past data. Worldwide, the use of fossil fuels, including oil and natural gas, produced almost 30% of the world's energy in 2020, according to the World Energy Report [10]. Many academics are drawn to applying machine learning approaches to oil production operations because of the need for accurate production forecasts. More and more, the oil and gas sector is looking to machine learning, especially for quick evaluations and production predictions [11]. The field of computer science known as artificial intelligence (AI) aims to teach computers to think and act like humans by simulating human intelligence and applying it to problems that are very complicated and non-linear [12]. A branch of artificial intelligence known as machine learning leverages massive datasets and statistical models to discover answers, enabling computers to mimic human learning in their ability to learn and adapt [13].

2. Related works

In recent times, " the use of machine learning (ML) and deep learning (DL) methods " has become increasingly effective for forecasting oil and gas production, showcasing their capabilities across different formations and data sets. For instance, a study by " Kim et. al. focused on predicting cumulative gas production (CGP) in Canada's Montney Formation using a

Cite as: G. Çit and A. Alyahya, "Enhanced oil and gas production forecasting through stacked generalization ensemble learning technique," *Sakarya University Journal of Computer and Information Sciences,* vol. 8, no. 2, pp. 212-222, 2025. doi: 10.35377/saucis...1579599

range of models, including artificial neural networks (ANN), 1D convolutional neural networks (1D-CNN), long short-term memory (LSTM) networks, and a combination of 1D-CNN and LSTM models. By utilizing early production data, well details, and fracture treatment parameters, the hybrid model showed outstanding performance in enhancing gas production predictions [14].

Similarly, research by Zanjani et al. assessed the effectiveness of ANN, linear regression (LR), and support vector regression (SVR) using data from the Volve field. Although ANN excelled in forecasting hydrocarbon production for well NO159-F-1C, the research underscored the necessity of customizing model selection for specific datasets, as no single algorithm is universally dominant [15].

Tan et al. utilized six algorithms—MLR, XGBoost, LightGBM, ML, RF, and back propagation—to predict output in the WY shale gas block in China, building on previous work in the field. Despite the research's limitations, such as its constricted dataset from a specific area, XGBoost emerged as the most efficient model via an R^2 value of 0.87 [16]. Instead, extra trees achieved the greatest accuracy ($R^2 = 0.809$) when Hui et al. evaluated shale gas production in the Fox Creek region using four methods: linear regression, neural networks, XGBoost, and extra trees [17].

Eight DL and ML models were investigated in Saudi Arabia by N. M. Ibrahim et al.: ANN, RNN, Decision Tree Regression (DTR), XGBoost, SVR, MLR, Polynomial Linear Regression (PLR), and Random Forest Regression (RFR). Based on the data provided by Saudi Aramco, the top-performing networks were ANN, XGBoost, and RNN, via R^2 values of 0.9627 for oil, 0.9012 for gas, and 0.926 for water, respectively. A disadvantage of the research is that the dataset is limited to wells in Saudi Arabia and does not include any geological variation [18]. For time-series data, S. Hosseini et.al proposed a hybrid LSTM-1D CNN model for predicting oil production in the Volve field, with the LSTM model achieving an R^2 score of 0.98. While promising, the study emphasized the need for further investigation into the model's generalizability across different wells [19].

In another study, Song et al. conducted a comprehensive analysis of the productivity of 394 offshore oil wells in China using various machine learning models, such as Linear Regression (LR), XGBoost, LightGBM, Back Propagation (BP) Neural Network, and Long Short-Term Memory (LSTM). XGBoost emerged as the leading model due to its exceptional generalization ability and stability across diverse datasets. In contrast, LightGBM displayed issues with overfitting, highlighting the critical need for selecting suitable machine learning algorithms that align with the dataset's characteristics and specific application requirements. The study underscores that the choice of model can significantly impact the performance and reliability of production forecasts, especially in complex offshore environments [20].

Lastly, Liu et al. proposed a stacked generalization ensemble model to optimize and predict the rate of penetration (ROP) during gas well drilling in Xinjiang, China. This model integrated six machine learning algorithms: Support Vector Regression (SVR), Random Forest (RF), Extremely Randomized Trees (ET), Gradient Boosting (GB), LightGBM, and Extreme Gradient Boosting (XGB). While the ET model showed the highest individual performance with an R^2 of 0.9268 on the test set, the stacked generalization model surpassed all individual models, achieving an R^2 of 0.9568 on the test set. The study effectively demonstrated the enhanced predictive power of combining multiple models, particularly in complex operations like drilling, where high predictive accuracy is crucial for optimizing operational efficiency [21].

Similarly, F.Ye et al. explored production prediction of tight and shale gas wells using a dataset comprising geological, reservoir, engineering parameters, and production data from over 200 wells in Gas Field A and 207 wells in Shale Gas Field B. The study utilized several machine learning algorithms, including Random Forest (RF), Extremely Randomized Trees (ET), LightGBM, and Gradient Boosting Regression (GBR), alongside a blending ensemble learning model. The blending model demonstrated superior predictive accuracy and generalization, particularly for shale gas production, outperforming individual models. The ensemble learning approach achieved an impressive R^2 of 0.9524 for shale gas production, emphasizing the efficacy of integrating multiple models to enhance prediction accuracy in complex reservoir conditions [22].

3. Machine Learning Models (ML)

The current digital transformation activities depend mainly on machine learning, a subfield of Artificial Intelligence AI [23]. It consists of a variety of approaches that enable systems to learn from data and make decisions accordingly. In order to tackle a wide range of problems, the three primary branches of “ machine learning—supervised, unsupervised, and reinforcement learning ”—offer a general framework [24]. “ This study utilized four different supervised machine learning models and one ensemble learning technique ”.

3.1 Random Forest regressor (RFR)

When it comes to classification and regression, the Random Forest method is the better option. It takes input data as a starting point, builds multiple models, gathers predictions from each, and then utilizes a voting process to choose the best answer [25]. Decision trees are the backbone of this method, which averages the results from various decision tree regressors (DTR) to arrive at the final forecast. The forecast is derived by averaging the results from each individual tree [26]. “ Created in 2001 by Professor Leo Breiman, this method is also known as Random Decision Forests ” [27]. To build the model, we first split the input into several samples according to the number of trees, then generate a basic prediction model for each sample, and finally use a bagging technique to combine the results of all the models to get the final

prediction [28]. Each decision tree in the Random Forest is completely grown, so there's no need to slow down processing. To avoid overfitting and get more accurate findings, it's recommended to increase the number of trees [29]. “As seen in Figure 1”.

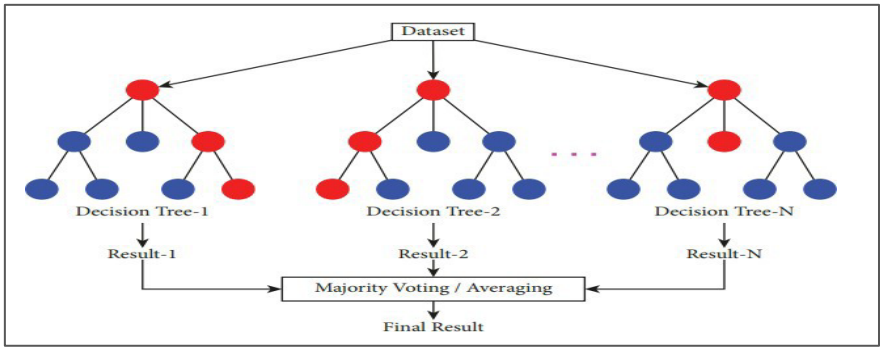


Figure 1. Design of a random forest model [26].

Predicting how many trees to use, how many predictor variables to evaluate at each split, node sizes, and minimum sample counts at leaf nodes are all critical components of the Random Forest Regression (RFR) model [30]. A dependent variable can be predicted by a random forest of simple trees within the framework of a regression model. Using the input variable x, the method generates K separate regression trees $h_{kk}(x)$. The model's forecast is the mean of all the forecasts made by all the trees in the forest using the given set of inputs (x), where k ranges from 1 to K. Using a process called bootstrapping, this methodology can increase the trees' variety, which in turn reduces the possibility that their aggregated results will be similar to other trees [29], “as seen in Equation 1” :

$$\text{RFR prediction } \frac{1}{kk}\sum_{kk=1}^{kk} h_{kk}(x) \qquad (1)$$

3.2 Extremely Randomized Trees Regressor (ETR)

The Extremely Randomized Trees Regressor (ETR) is a robust ensemble machine learning model designed for both regression and classification tasks. Similar to the Decision Tree Regressor, ETR constructs decision trees to model relationships within a dataset. However, it introduces additional randomness to enhance diversity among trees, leading to more robust predictions and reduced overfitting [30].In ETR, the decision trees are built by randomly selecting features and thresholds at each split, as opposed to choosing the optimal split point. This randomness increases the variance among individual trees while maintaining low bias in the overall ensemble. The branches represent the decision-making criteria, while the nodes correspond to options or events. Each node is associated with features, and branches signify potential values of these features [31].

During training, ETR uses a random sampling of data (with replacement, known as bootstrapping) to grow multiple decision trees. The entire dataset is divided into sections using randomly chosen thresholds for the features at each node. This process continues recursively until a stopping criterion, such as the minimum node size, is reached, resulting in terminal nodes [32]. Unlike traditional decision tree algorithms, ETR does not solely rely on the Mean Squared Error (MSE) to determine splits. Instead, splits are chosen by evaluating a random subset of features and thresholds, ensuring faster training and greater model diversity. The model's prediction is the average of the outputs from all trees in the ensemble, calculated, “as seen in Equation 2”.

$$E(m) = \frac{1}{M}\sum_{i=1}^{M} [y_i(m) - \hat{y}(m)]^2 \qquad (2)$$

“As seen in Figure 2”, the initial node in each decision tree acts as the root, representing the entire dataset. Subsequent nodes enable the dataset to be divided into smaller, more homogenous subsets, ensuring accurate predictions while maintaining computational efficiency.

3.3 K -Nearest Neighbors (KNN)

Regression and classification are just two examples of the many applications of the widely used “ K-Nearest Neighbors (KNN) technique”. [33]. It is a simple yet powerful approach in machine learning, where it operates by comparing a data point to its nearest neighbors [34]. The fundamental principle of KNN is to assign an object to a category based on the

characteristics it shares most closely with nearby elements [35]. The value attributed to an object is calculated based on the average of its K nearest neighbors. Applying weights to nearby entities can enhance the method's effectiveness, particularly when closer neighbors have a more substantial influence on the average than those farther away [36]. KNN helps to avoid overfitting by adjusting a parameter, k, which is inversely related to the error rate [35]. “As seen in Equation 3,” One of the most common distance metrics used in KNN is the Euclidean distance, which measures the space between two points, (r, s).

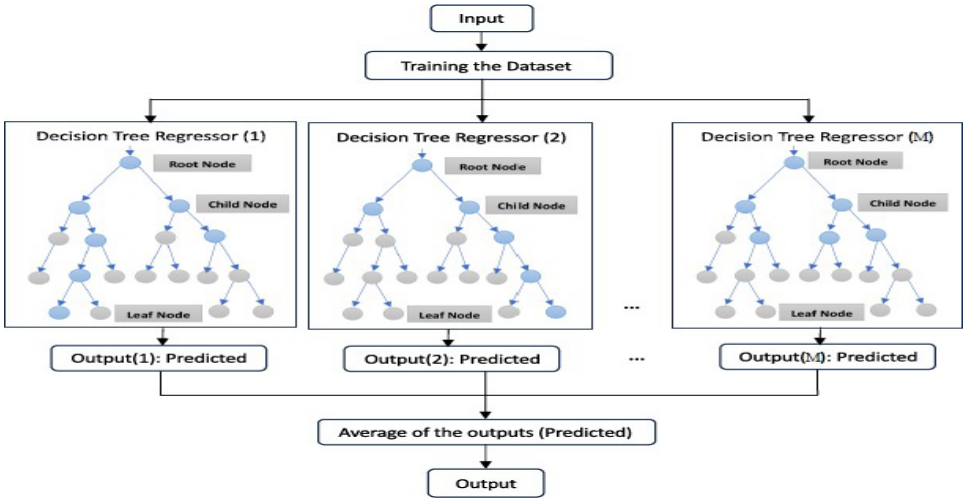


Figure 2. Shows the diagram of the extra-trees regressor [32].

3.3 K -Nearest Neighbors (KNN)

Regression and classification are just two examples of the many applications of the widely used “ K-Nearest Neighbors (KNN) technique”. [33]. It is a simple yet powerful approach in machine learning, where it operates by comparing a data point to its nearest neighbors [34]. The fundamental principle of KNN is to assign an object to a category based on the characteristics it shares most closely with nearby elements [35]. The value attributed to an object is calculated based on the average of its K nearest neighbors. Applying weights to nearby entities can enhance the method's effectiveness, particularly when closer neighbors have a more substantial influence on the average than those farther away [36]. KNN helps to avoid overfitting by adjusting a parameter, k, which is inversely related to the error rate [35]. “As seen in Equation 3,” One of the most common distance metrics used in KNN is the Euclidean distance, which measures the space between two points, (r, s).

$$DDDDDD\,(\mathrm{R,S}) = \overline{\sum_{4=1}^{m} (rr_{ii} - DD_{i})^{2}} \qquad (3)$$

In a space with m dimensions, let R be represented as $rr_1, rr_2, ..., rr_{mm}$, and S as $DD_1, DD_2, ..., DD_{mm}$ [33]. The k-Nearest Neighbors (KNN) Search technique for k equals 3, the three points in dataset D that are nearest to the query point DD_1 “As seen in Figure 3”.

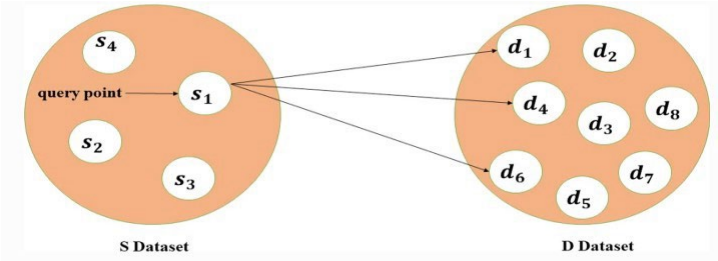


Figure 3. Illustration of the k-Nearest Neighbors (KNN) search method with k set to 3[33].

3.4 Gradient Boosting Regressor (GBR)

Gradient Boosting is a method in ensemble learning that constructs predictive models step by step by integrating the capabilities of weak learners, often decision trees, to develop a robust predictive model [37]. The fundamental principle of Gradient Boosting involves iteratively training new models to address the residual errors of prior models, thereby enhancing the overall accuracy of predictions [38] as seen in Figure 4.

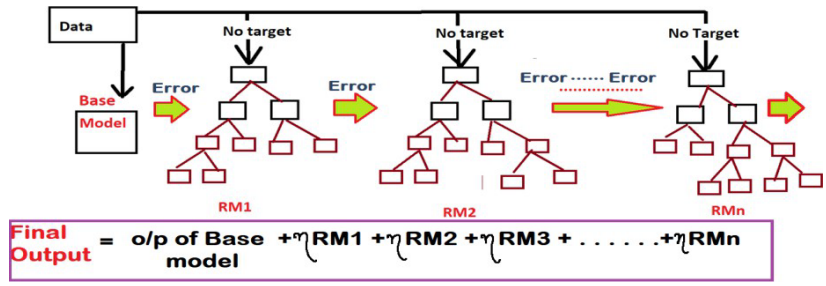


Figure 4. Understanding Gradient Boosting for Regression [38].

3.5 Stacking Regressor

Stacking, also referred to as Stacked Generalization, is an ensemble learning method aimed at boosting predictive accuracy by merging several models. This technique comprises two primary phases: initially, multiple base models (often called level-0 models) are trained using the same dataset; subsequently, a meta-model (level-1) is trained to make predictions from these base models. The outputs of the base models act as input features for the meta-model, which is designed to combine these predictions to enhance the accuracy of the overall prediction, as seen in Figure 5 [39]. In this study, stacking will be applied by integrating four machine learning algorithms. Each base model will analyze the dataset separately, producing predictions that will be utilized by the meta-model.

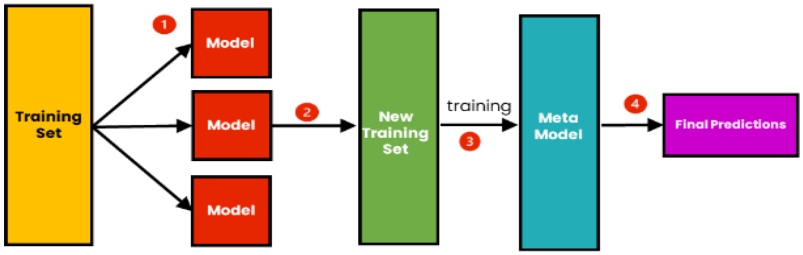


Figure 5. Shows the structure of Stacked Generalization [39].

4. The proposed system

This study utilizes four machine learning techniques, namely Random Forest Regressor (RFR), Extremely Randomized Trees Regressor (ETR), K-Nearest Neighbors (KNN), and Gradient Boosting Regressor (GBR), which are combined through a method known as Stacked Generalization. Prediction challenges frequently make use of these algorithms because of their adaptability, capacity to learn non-linear correlations, and ensemble abilities in learning. Their accuracy in forecasting results and ease of interpretation significantly aid decision-making processes concerning the optimization of the production and distribution of resources. The framework suggested includes several crucial phases, as seen in Figure 6. The procedure starts with the acquisition of the dataset and involves data preprocessing, cleaning and normalization. Subsequently, the machine learning models are applied, utilizing the stacking model to capitalize on the advantages of various algorithms. In the final phase, the system evaluates the models' performance using metrics to ensure accurate and reliable forecasting outcomes.

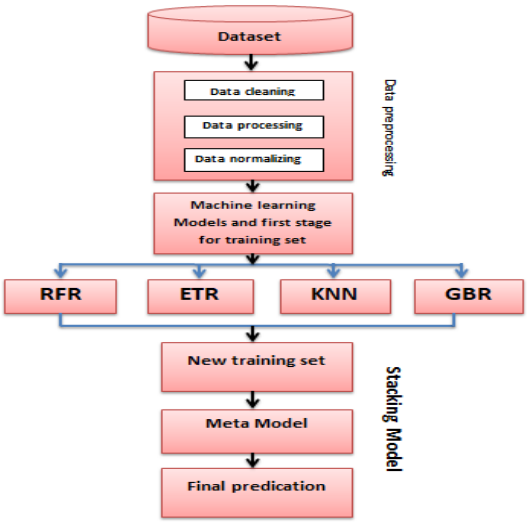


Figure 6. Overview of the Suggested System's Workflow.

4.1 Datasets Description

The main and most important phase of this research is collecting data. This dataset includes production records from oil and gas wells in New York State that were drilled between 2001 and the current day [40]. This data set features details such as county, company name, well status, well type, and producing formation. This dataset comprises 18 columns and approximately 302,000 rows. “As seen in Figures 7 and 8,” showcases a segment of this data set, highlighting oil and gas production for the ten most active wells.

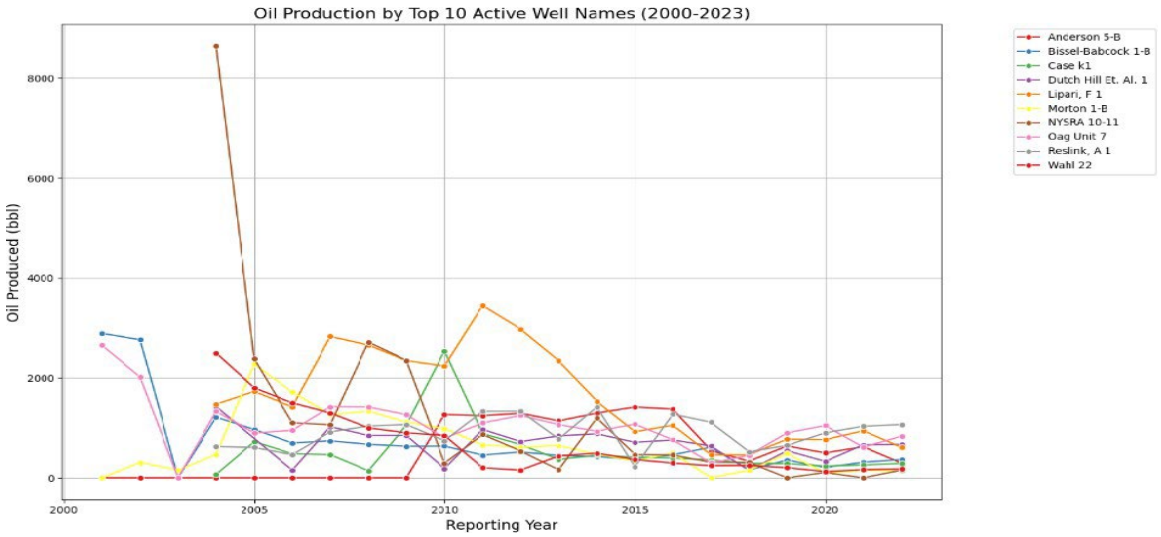


Figure 7.Illustrated oil production: top 10 active wells.

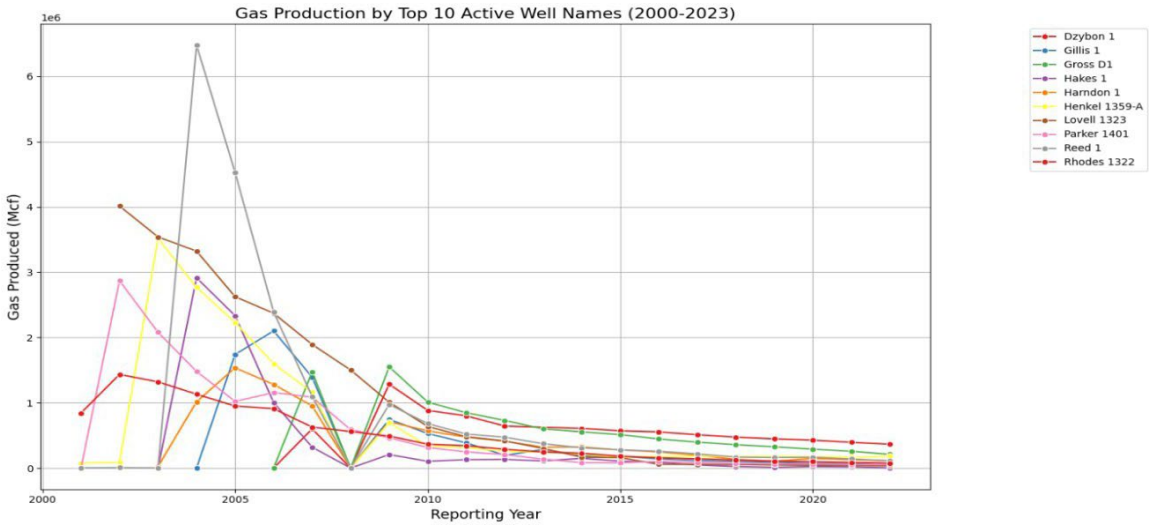


Figure 8. Illustrated gas production of the top 10 active wells.

4.2 Dataset Preprocessing

Before manipulating data, it's essential to prepare the dataset to ensure its quality and suitability for analysis. In this study, Google Colab was employed as the online programming platform for Python. The data preparation phase involved three main steps: cleaning the data, processing the data, and normalizing it. The initial and most crucial step, data cleaning, ensures that the dataset is devoid of errors and inconsistencies, confirming the accuracy and completeness of all information. Next, the data processing step and the normalization step adjust the numerical values using an L2 scaler, which scales the data to a range from 0 to 1.

4.3 Methodology

The essential first step in training a machine learning model involves identifying the parameters that will yield the best outcomes. Given the complexity of selecting these parameters, we explored every possible value until we established the best settings for each model as seen in Table 1.

Table 1. Parameters for machine learning models.

Model	Parameters
RFR	n_estimators = 40, max_depth =13, random_state = 33
ETR	n_estimators = 50, max_depth =15, random_state = 33
KNN	n_neighbors=13, weights='uniform', algorithm='auto
GBR	n_estimators=50, learning_rate=0.1, random_state = 33

When the oil and gas parameters have been defined, the data is split into two sets: the training dataset and the testing dataset. Machine learning models are trained using the training dataset, which is an essential subset of the dataset. It is divided into a training set and a validation set and takes up the most space, 75% of the total data. To evaluate the model's performance and adjust its hyperparameters, a separate dataset is needed after training. Hyperparameter optimization benefits from this dataset, which is called the validation set. The purpose of the testing dataset is to foretell how the model will respond to novel, unseen data in real-world scenarios. It makes up a quarter of all the data.

5. Experimental results

The methodology was consistently applied to four machine learning algorithms: Random Forest Regressor (RFR), Extremely Randomized Trees Regressor (ETR), K-Nearest Neighbors (KNN), and Gradient Boosting Regressor (GBR), each utilizing unique sets of parameters. These algorithms were implemented following the cleaning, normalization, and processing of the dataset. The results generated by these individual models are combined through Stacked Generalization, an ensemble learning model, and used as input features for a final model (known as the meta-model or final estimator), which, by combining the results of the base models, learns to predict the target variable. That captures different patterns and interactions in the data that individual models might miss. In this stacking regressor, the " final estimator is a RandomForestRegressor, with n_estimators=50 and random_state=42". This strategy helps to enhance predictive accuracy. The performance of these models was assessed using essential metrics such as Mean Absolute Error (MAE), R^2, and Mean Squared Error (MSE). "As seen in Table 2", the effectiveness of each single model and the stacked generalization model across two categories of data: Oil and Gas. The stacked model yielded superior results in predicting oil and gas production, achieving the highest R^2 scores, with averages of 0.974750 for oil and 0.999848 for gas. Overall, the models demonstrated slightly better predictive accuracy for gas data compared to oil data, highlighting their superior performance for forecasting in this scenario.

Table 2. Results of machine learning models and the stacking model.

Models	OUTPUT	MAE	MSE	R^2
RFR	Oil	0.000273	0.000011	0.948631
	Gas	0.000273	0.000011	0.999720
ETR	Oil	0.000295	0.000073	0.932277
	Gas	0.000295	0.000007	0.999819
KNN	Oil	0.000324	0.000017	0.938847
	Gas	0.000324	0.000017	0.999580
GBR	Oil	0.00306	0.000276	0.391319
	Gas	0.004396	0.000137	0.996638
Stacking model	Oil	0.000469	0.000013	0.970900
	Gas	0.000221	0.000007	0.999806

As seen in Figure 9, RFR, ETR, KNN, GBR, and the Stacking model correlation coefficient scores (R^2 *vvvvvvvvv*).

As seen in Figure 10, a comparison of the actual oil output with the forecasted oil production using several machine learning methods, such as " RFR, ETR, KNN, GBR, and a Stacking Model". According to the findings, there was an accurate correspondence between the forecasted and actual oil production. Similarly, "As seen in Figure 11", the labeled figure also compares actual gas output with predicted gas production using the same ML models and the Stacking Model. The models' capacity to precisely forecast gas output is demonstrated by the findings, which reveal a high link. The Stacking Model closely matches the real data for both gas and oil production, demonstrating outstanding forecasting capabilities.

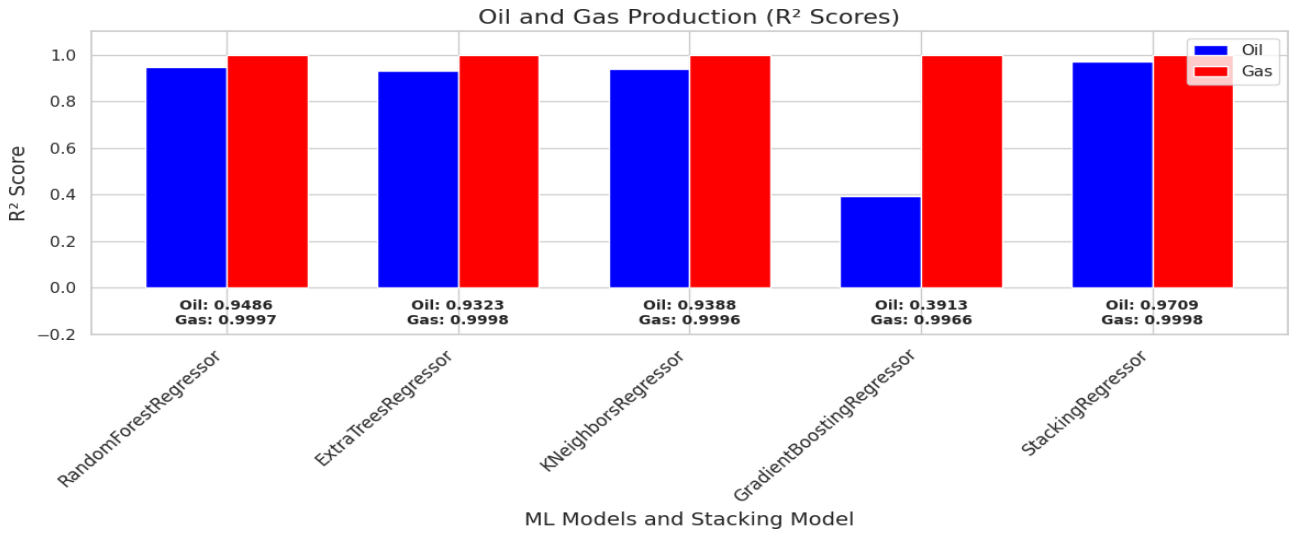


Figure 9. Displays Oil and Gas R² values in ML models and the Stacking Model.

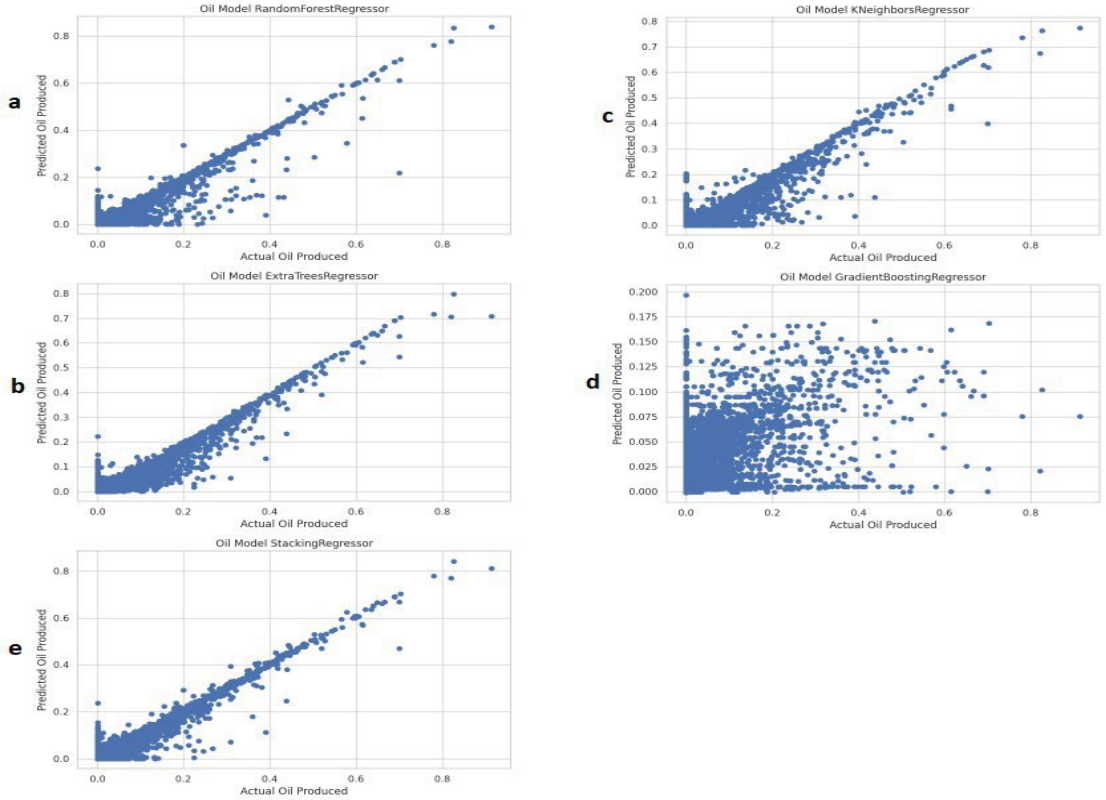


Figure 10. presents a comparison between the forecasted and actual Oil production employing various machine learning models, as well as a stacking model. The results are as follows: (a) Random Forest Regressor (RFR), (b) Extremely Randomized Trees Regressor (ETR), (c) K-Nearest Neighbors (KNN), (d) Gradient Boosting Regressor (GBR), and (e) the Stacking Model.

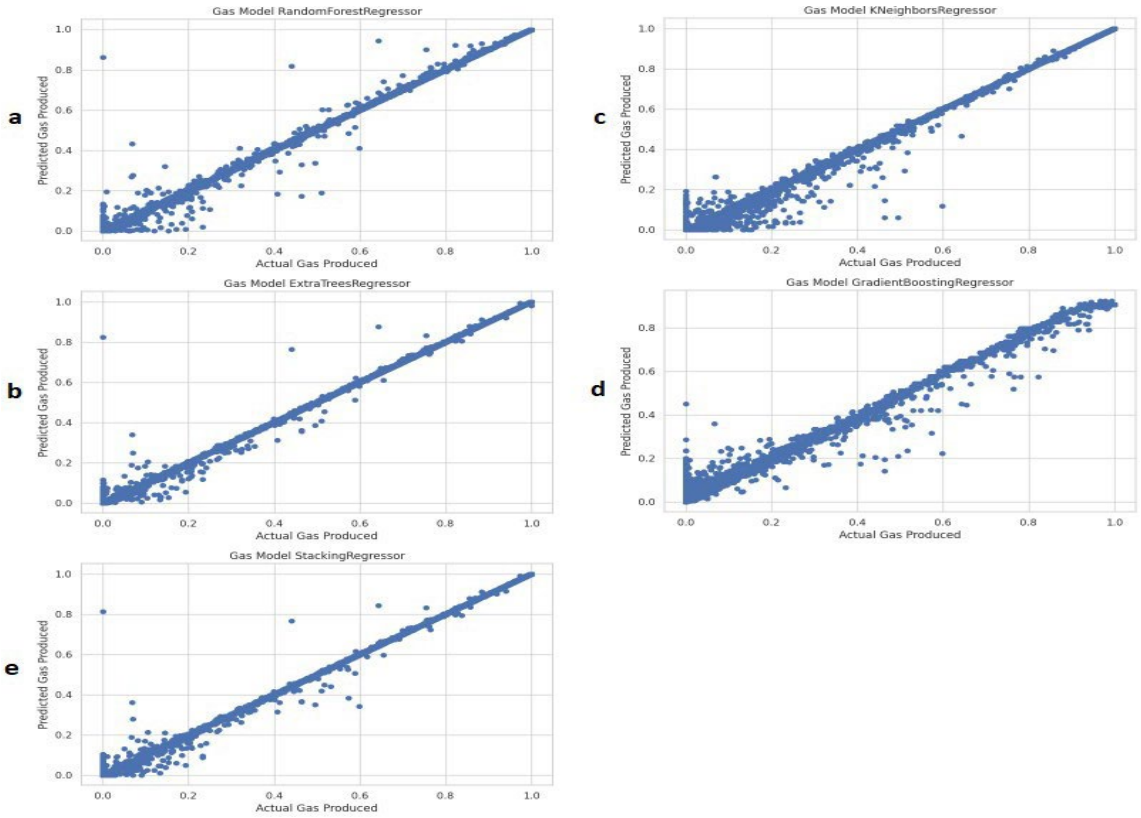


Figure 11. presents a comparison between the forecasted and actual Gas production employing various machine learning models, as well as a stacking model. The results are as follows: (a) Random Forest Regressor (RFR), (b) Extremely Randomized Trees Regressor (ETR), (c) K-Nearest Neighbors (KNN), (d) Gradient Boosting Regressor (GBR), and (e) the Stacking Model.

6. Conclusion

Predicting the amount of achievable oil and gas with any degree of accuracy is necessary for the petroleum sector .This ability allows companies to efficiently allocate resources, optimize production, and validate the advantages of predicting oil and gas output. Various techniques and models are employed to assess the potential recovery from current and future reserves over a specified timeframe. “ The study employs machine learning models such as Random Forest Regressor (RFR), Extremely Randomized Trees Regressor (ETR), K-Nearest Neighbors (KNN), and Gradient Boosting Regressor (GBR) for production prediction ”. These models are trained and tested on production data, and their results are combined using Stacked Generalization, a method of ensemble learning. The performance of the models is assessed using metrics like “ Mean Squared Error (MSE), Mean Absolute Error (MAE), and R^2 ”. The experimental findings show that the Stacking Model delivers the highest accuracy, with an R-squared value of 99%, indicating its superior predictive capability . Improving predicted accuracy is the primary goal of this research, which uses an ensemble learning method. In the future, we want to build a more unified system that integrates ML and DL models with other ensemble learning techniques, evaluates them against each other, and chooses the best model according to dataset type, attributes, and other pertinent factors.

References

[1] British Petroleum, "Statistical Review of World Energy," BP Global, 2021. [Online]. Available: https://www.bp.com

[2] J. G. Speight, Handbook of Petroleum Refining. 2014. [Online]. Available: https://www.academia.edu/63659108/Handbook_of_Petroleum_Refining

[3] D. Orodu, O. F. Aworinde, and A. F. Alayande, "A hybrid machine learning framework for enhanced reservoir characterization," J. Petroleum Sci. Eng., vol. 207, p. 109114, 2021, doi: 10.1016/j.petrol.2021.109114.

[4] A. F. Khan and S. R. Alam, "Adaptive Neuro-Fuzzy Inference System with metaheuristic tuning for petroleum production forecasting," Applied Soft Computing, vol. 114, p. 108050, 2022, doi: 10.1016/j.asoc.2021.108050.

[5] M. A. Ullah, S. M. Khaleque, and S. Sikder, "Prediction of oil production using optimized machine learning models," Energies, vol. 14, no. 16, p. 4923, 2021, doi: 10.3390/en14164923.

[6] M. J. Fetkovich, "Decline Curve Analysis Using Type Curves," J. Petroleum Technol., vol. 32, no. 6, pp. 1065-1077, 1980.

[7] M. J. Abhishek and V. Kumar, "Gradient boosting regression tree model for enhanced oil production prediction," Processes, vol. 10, no. 2, p. 234, 2022, doi: 10.3390/pr10020234.

[8] K. M. Ali and J. Zhang, "Application of metaheuristic optimization algorithms for predictive analysis in petroleum engineering," J. Petroleum Exploration Production Technol., vol. 12, no. 5, pp. 1325–1335, 2022, doi: 10.1007/s13202-021-01402-w.

[9] C. S. W. Ng, A. J. Ghahfarokhi, and M. N. Amar, "Well production forecast in Volve field: Application of rigorous machine learning techniques and metaheuristic algorithm," J. Petroleum Sci. Eng., vol. 208, p. 109468, 2022, doi: 10.1016/j.petrol.2021.109468.

[10] S. D. Mohaghegh, "Machine Learning Applications in Reservoir Engineering: Part 1," J. Petroleum Technol., vol. 69, no. 6, pp. 70-77, 2017, doi: 10.2118/0617-0070-JPT.

[11] J. X. Chen, H. L. Wang, and K. Zhao, "Comparative evaluation of machine learning techniques for hydrocarbon reservoir prediction," Energies, vol. 14, no. 3, p. 806, 2021, doi: 10.3390/en14030806.

[12] A. S. Abou-Sayed, "AI in the Petroleum Industry," Society of Petroleum Engineers AI Newsletter, 2021. [Online]. Available: https://www.spe.org

[13] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed. Pearson, 2021.

[14] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[15] M. Kim, "Deep Learning-Based Prediction of the Cumulative Gas Production of the Montney Formation, Canada," GeoConvention, 2020. [Online]. Available: https://geoconvention.com/wp-content/uploads/abstracts/2020/57980-deep-learning-based-prediction-of-the-cumulative-g.pdf

[16] M. S. Zanjani, M. A. Salam, and O. Kandara, "Data-Driven Hydrocarbon Production Forecasting Using Machine Learning Techniques," Int. J. Comput. Sci. Inf. Security, vol. 18, no. 6, pp. 65–72, 2020.

[17] C. Tan et al., "Fracturing productivity prediction model and optimization of the operation parameters of shale gas well based on machine learning," Lithosphere, vol. 2021, no. Special 4, p. 2884679, 2021, doi: 10.2113/2021/2884679.

[18] G. Hui, S. Chen, Y. He, H. Wang, and F. Gu, "Machine learning-based production forecast for shale gas in unconventional reservoirs via integration of geological and operational factors," J. Natural Gas Sci. Eng., vol. 94, p. 104045, 2021, doi: 10.1016/j.jngse.2021.104045.

[19] N. M. Ibrahim et al., "Well Performance Classification and Prediction: Deep Learning and Machine Learning Long Term Regression Experiments on Oil, Gas, and Water Production," Sensors, vol. 22, no. 14, p. 5326, 2022, doi: 10.3390/s22145326.

[20] S. Hosseini and T. Akilan, "Advanced Deep Regression Models for Forecasting Time Series Oil Production," arXiv preprint arXiv:2308.16105, 2023.

[21] L. Song, C. Wang, C. Lu, S. Yang, and C. Tan, "Machine Learning Model of Oilfield Productivity Prediction and Performance Evaluation," J. Physics: Conference Series, vol. 2468, no. 1, p. 012084, 2022, doi: 10.1088/1742-6596/2468/1/012084.

[22] N. Liu, H. Gao, Z. Zhao, Y. Hu, and L. Duan, "A stacked generalization ensemble model for optimization and prediction of the gas well rate of penetration: a case study in Xinjiang," J. Petroleum Exploration Production Technol., vol. 11, pp. 3533-3546, 2021, doi: 10.1007/s13202-021-01402-z.

[23] F. Ye, X. Li, N. Zhang, and F. Xu, "Prediction of Single-Well Production Rate after Hydraulic Fracturing in Unconventional Gas Reservoirs Based on Ensemble Learning Model," Processes, vol. 12, no. 6, p. 1194, 2024, doi: 10.3390/pr12061194.

[24] S. Ray, "A quick review of machine learning algorithms," in Proc. Int. Conf. Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), 2019, pp. 35-39, doi: 10.1109/comitcon.2019.8862451.

[25] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," Science, vol. 349, no. 6245, pp. 255-260, 2015, doi: 10.1126/science.aaa8415.

[26] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[27] M. Y. Khan, "Automated prediction of Good Dictionary EXamples (GDEX): a comprehensive experiment with distant supervision, machine learning, and word embedding-based deep learning techniques," Complexity, vol. 2021, pp. 1-18, 2021, doi: 10.1155/2021/2553199.

[28] L. Breiman, "Bagging Predictors," Machine Learning, vol. 24, no. 2, pp. 123–140, 1996, doi: 10.1007/BF00058655.

[29] A. K. Ali and A. M. Abdullah, "Fake accounts detection on social media using stack ensemble system," Int. J. Electrical Comput. Eng., vol. 12, no. 3, pp. 3013-3022, 2022.

[30] S. P. Rao and A. V. K. Shetty, "Random forest-based predictive models for enhanced fluid flow estimation in pipelines," J. Petroleum Sci. Eng., vol. 199, p. 108382, 2021, doi: 10.1016/j.petrol.2021.108382.

[31] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," Machine Learning, vol. 63, no. 1, pp. 3–42, 2006, doi: 10.1007/s10994-006-6226-1.

[32] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. Springer, 2009.

[33] T. Aziz and M. R. Camana, "REM-Based Indoor Localization with an Extra-Trees Regressor," Electronics, vol. 12, no. 20, p. 4350, 2023, doi: 10.3390/electronics12204350.

[34] R. K. Halder, "Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications," J. Big Data, vol. 11, no. 1, 2024, doi: 10.1186/s40537-024-00973-y.

[35] T. Timbers, T. Campbell, and M. Lee, "Chapter 7 Regression I: K-nearest neighbors," in Data Science: A First Introduction, CRC Press, 2022. [Online]. Available: https://datasciencebook.ca/regression1.html

[36] C. Gkerekos, I. Lazakis, and G. Theotokatos, "Machine learning models for predicting ship main engine Fuel Oil Consumption: A comparative study," Ocean Eng., vol. 188, p. 106282, 2019, doi: 10.1016/j.oceaneng.2019.106282.

[37] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," The Annals of Statistics, vol. 29, no. 5, pp. 1189-1232, 2001.

[38] A. Ali, "Gradient Boosting Machine Learning Algorithm," Dec. 2023, doi: 10.13140/RG.2.2.31609.65123.

[39] M. Kalirane, "Ensemble Learning in Machine Learning: Bagging, Boosting and Stacking," Analytics Vidhya, Jan. 2024. [Online]. Available: https://www.analyticsvidhya.com/blog/2023/01/ensemble-learning-methods-bagging-boosting-and-stacking/

[40] "Oil and Gas Annual Production: Beginning 2001," Data.gov. [Online]. Available: https://catalog.data.gov/dataset/oil-and-gas-annual-production-beginning-2001

Author Information Form

Author(s) Contributions

Azhar Alyahya was responsible for carrying out the research, analyzing the data, and preparing the first draft of the manuscript. Dr. Gülüzar Çit provided academic supervision throughout the project, contributed to the study design and interpretation of the findings, and offered valuable feedback and revisions on the manuscript. Both authors reviewed and approved the final version of the paper.

Conflict of Interest Notice

No potential conflict of interest was declared by authors.

Ethical Approval

It is declared that during the preparation process of this study, scientific and ethical principles were followed, and all the studies benefited from are stated in the bibliography.

Availability of data and material

https://catalog.data.gov/dataset/oil-and-gas-annual-production-beginning-2001

Artificial Intelligence Statement

No artificial intelligence tools were used while writing this article.

Plagiarism Statement

This article has been scanned by iThenticate ™.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
http://saucis.sakarya.edu.tr/
e-ISSN: 2636-8129
Publisher: Sakarya University
Vol. 8, No. 2, 223-244, 2025
DOI: 10.35377/saucis...1663435
RESEARCH ARTICLE

Leveraging Graph Neural Networks for IoT Attack Detection

Onur Ceran[1,*], Erdal Özdoğan[2], Mevlüt Uysal[1]

[1]Gazi University, Ankara, Türkiye, ror.org/054xkpr46
[2]Uludag University, Bursa, Türkiye, ror.org/03tg3eb07

Corresponding author:
Onur Ceran, Gazi University, Ankara, Türkiye,
onur.ceran@gazi.edu.tr

Article History:
Received: 22.03.2025
Revised: 09.05.2025.2025
Accepted: 28.05.2025
Published Online: 16.06.2025

ABSTRACT
The widespread adoption of Internet of Things (IoT) devices in multiple sectors has driven technological progress; however, it has simultaneously rendered networks vulnerable to advanced cyber threats. Conventional intrusion detection systems face challenges adjusting to IoT environments' ever-changing and diverse characteristics. To address this challenge, researchers propose a novel hybrid approach combining Graph Neural Networks and XGBoost algorithm for robust intrusion detection in IoT ecosystems. This paper presents a comprehensive methodology for integrating GNNs and XGBoost in IoT intrusion detection and evaluates its effectiveness using diverse datasets. The proposed model preprocesses data by standardization, handling missing values, and encoding categorical features. It leverages GNNs to model spatial dependencies and interactions within IoT networks and utilizes XGBoost to distill complex features for predictive analysis. The late fusion technique combines predictions from both models to enhance overall performance. Experimental results on four datasets, including CICIoT-2023, CICIDS-2017, UNSW-NB15, and IoMT-2024, demonstrate the efficacy of the hybrid model. High accuracy, precision, recall, and AUC values indicate the model's robustness in detecting attacks while minimizing false alarms. The study advances IoT security by introducing synergistic solutions and provides practical insights for implementing intrusion detection systems in real-world IoT environments.

Keywords: GNN, IoT IDS, XGBoost, IPS, IoT Security

1. Introduction

The rife adoption of the Internet of Things (IoT) has revolutionized different sectors, including agriculture, the power industry, transportation, and healthcare. However, this quick proliferation of IoT ecosystems has also exposed them to increasingly sophisticated cyber threats and security vulnerabilities. Protecting the IoT device and the data it processes has become a major concern [1]. Consequently, there is a critical need for robust and efficient intrusion detection systems (IDS) specifically tailored for IoT environments. An Intrusion Detection System (IDS) is crucial for network operations. It actively observes network traffic and promptly notifies the administrator of any irregularities or suspicious activities within the network [2]. However, traditional IDS approaches often struggle to adapt to IoT networks' ever-changing and diverse characteristics. These networks are characterized by diverse device types, communication protocols, and network topologies [3]. To respond to these difficulties, researchers and practitioners have turned to innovative machine learning techniques such as support vector machines, Naïve Bayes, decision trees, and random forests, capable of modeling the complex relationships and interactions within IoT networks [4]. Integrating various machine learning algorithms into hybrid models has proven to be a promising approach for improving the accuracy and dependability of IDSs within IoT environments [5]. Graph Neural Networks (GNNs) and XGBoost have received considerable recognition in machine learning algorithms due to their proficiency in determining multifarious patterns and relationships in high-dimensional and graph-structured datasets.

The effectiveness of hybrid models lies in their ability to leverage the complementary strengths of different algorithms. GNNs excel at learning representations of graph-structured data to execute tasks that follow in the sequence [6], such as IoT network traffic, by exploiting the inherent relational structure among network entities at either the edge or node level [7]. By propagating information across interconnected nodes and edges, GNNs can effectively capture localized patterns and dependencies within the network [8]. On the other hand, Extreme Gradient Boosting (XGBoost) is adept at handling tabular data and capturing nonlinear relationships between features by discarding missing values and mitigating overfitting problems through parallel processing [9]. The XGBoost algorithm, rooted in gradient-boosted decision trees, is a potent tool for enhancing gradients, offering effective solutions for regression and classification tasks by integrating new algorithms with

Cite as: O. Ceran, E. Özdoğan & M. Uysal. (2025). Leveraging graph neural networks for IoT attack detection. *Sakarya University Journal of Computer and Information Sciences*. vol. 8, no. 2, pp. 223-244, 2025 doi: 10.35377/saucis...1663435

GBDT methods into a versatile soft computing library [10]. The decision to adopt a hybrid approach combining GNNs and XGBoost in IoT intrusion detection is motivated by several factors. Firstly, GNNs are appropriate for modeling spatial dependencies and interactions among IoT devices and traffic flows, enabling fine-grained network behavior analysis. Meanwhile, XGBoost provides a robust framework for integrating the learned representations from GNNs and making global predictions based on comprehensive feature sets. Furthermore, integrating GNNs and XGBoost offers synergistic benefits, including enhanced feature extraction, improved generalization, and robustness against noise and adversarial attacks. By combining the capabilities of two algorithms, the hybrid model can effectively mitigate the limitations of individual approaches and achieve superior performance in IoT intrusion detection tasks. The widespread integration of IoT devices has triggered technological advancements across various sectors. However, this expansion has made IoT networks vulnerable to more complex cyber threats. Traditional IDSs frequently face difficulties adjusting to IoT settings' ever-changing and diverse characteristics. The proposed approach in this study has been developed to address these challenges.

1.1 Motivation

Adopting a hybrid approach that integrates Graph Neural Networks and XGBoost for IoT intrusion detection is deeply rooted in the intricate nature of IoT environments and the imperative need for robust Intrusion Detection Systems. IoT networks, characterized by diverse, interconnected devices, sensors, and actuators across various domains, such as healthcare and industrial automation, pose significant challenges to traditional IDS methods. The dynamic nature of these ecosystems, coupled with diverse device types, communication protocols, and network topologies, renders conventional rule-based and signature-based intrusion detection approaches inadequate in addressing evolving cyber threats. GNNs offer a compelling solution for modeling and analyzing complex relationships and dependencies among network entities in IoT networks, where data is inherently graph-structured. As the network environment becomes increasingly complex, conventional neural network solutions struggle to harness the wealth of information within network traffic data due to their singular structure. GNNs facilitate the propagation of information across interconnected nodes and edges, enabling fine-grained analysis of network behavior and the identification of anomalous patterns or malicious activities. By leveraging the graph structure of IoT data, GNNs can effectively capture localized patterns and dependencies within the network, providing insights into the dynamic interactions among IoT devices and traffic flows.

Complementing the capabilities of GNNs, XGBoost stands out as a powerful gradient-boosting algorithm that is noted for its capability to handle tabular data and capture nonlinear relationships between features. By sequentially constructing an ensemble of decision trees, XGBoost can learn from high-dimensional feature sets and capture complex decision boundaries, making it appropriate for IoT intrusion detection tasks. Furthermore, XGBoost's robustness and accuracy in classifying instances enhance its applicability in IoT security.

The motivation for this study arises from the need to develop a more efficacious IDS against the rapidly evolving cyber threats in IoT environments, as traditional methods often fall short in detecting these threats due to the complex structure of IoT networks and the interactions between devices. The proposed hybrid model in this study contributes an extensive solution to these challenges by combining GNN and XGBoost algorithms. The synergy between GNNs and XGBoost offers a holistic approach to addressing IoT environments' intricacy and dynamic nature. By integrating the strengths of both algorithms, the hybrid approach can cope with the limitations of individual methods and achieve superior performance in IoT intrusion detection. This hybrid model enables enhanced feature extraction, improved generalization, and robustness against noise and adversarial attacks. Consequently, it offers a robust and adaptive intrusion detection mechanism to safeguard IoT ecosystems against emerging cyber threats, thereby strengthening the security and resilience of IoT networks.

1.2 Research Gap

Even if the amount of research on IoT intrusion detection is increasing, existing solutions remain insufficiently effective against advanced threats that continue to evolve in complexity and sophistication. Traditional intrusion detection systems rely on rule-based and signature-based methodologies [11]. However, these methods struggle to adapt to the ever-changing IoT environments. Such environments are characterized by diverse network topologies, communication protocols, and device types [12], [13]. This inadequacy increases the risk of undetected intrusions, as conventional systems may fail to identify novel attack patterns that do not match predefined signatures or rules. Furthermore, the intricate interactions among interconnected IoT devices create complex, graph-structured data that conventional neural network solutions cannot fully exploit [14]. These conventional methods are constrained by their failure to adequately capture the intricate relational information inherent in IoT traffic data. This leads to a significant loss of essential insights regarding network behavior and potential security vulnerabilities. Consequently, there is a pressing need for innovative methodologies that can successfully deal with these problems. The research gap lies in the under-explored potential of hybrid models that combine the strengths of Graph Neural Networks and XGBoost for intrusion detection in IoT settings. GNNs show promise for graph and network data analysis, including in Network Intrusion Detection Systems NIDS. However, their effectiveness is hindered by limitations such as poor performance with limited or imbalanced training data and susceptibility to adversarial attacks. Consequently, further research into GNN-based NIDS is crucial to address these vulnerabilities and improve their robustness [10]. While GNNs offer a promising framework for comprehending complex relationships among network entities, they are often used in isolation without considering the complementary capabilities of ensemble methods like XGBoost, which excel in capturing nonlinear relationships and enhancing classification accuracy.

This study aims to fill this gap by providing a hybrid model that leverages the synergistic effects of GNNs and XGBoost, thus addressing the shortcomings of existing methods. Integrating these algorithms enhances feature extraction, improves generalization and fortifies the model's robustness against noise and adversarial attacks. By doing so, the proposed approach aspires to provide a more effective and adaptive intrusion detection mechanism capable of safeguarding IoT ecosystems against the multifaceted and evolving nature of cyber threats. This research contributes to the ongoing discourse on IoT security by introducing a novel framework that promises to elevate the performance of IDSs in increasingly complex environments.

1.3 Contribution

Firstly, a novel hybrid approach is presented in this paper, integrating GNNs and XGBoost for IoT intrusion detection. This approach offers a comprehensive solution to address IoT environments' complexity and dynamic nature. By leveraging the strengths of both algorithms, the hybrid model enhances the accuracy, robustness, and efficiency of IDSs in IoT ecosystems. Secondly, the article contributes to the body of knowledge in IoT security by demonstrating the effectiveness of synergistic solutions in mitigating the evolving cyber threats IoT networks face. Combining GNNs' ability to model graph-structured data and capture localized patterns with XGBoost's capability to handle tabular data and capture complex relationships, the hybrid model provides a holistic approach to intrusion detection in IoT environments. Furthermore, the article advances machine learning techniques in IoT security applications by exploring innovative methodologies and algorithms tailored specifically for IoT intrusion detection. Integrating GNNs and XGBoost represents a paradigm shift in intrusion detection research, offering new insights and avenues for enhancing the security and resilience of IoT ecosystems against emerging threats. Moreover, the article contributes to the practical implementation and deployment of IDSs in real-world IoT environments by providing insights into the hybrid model's performance, scalability, and adaptability. By evaluating the model on diverse IoT datasets and benchmarking it against existing approaches, the article offers valuable perceptivity into the feasibility and efficacy of hybrid solutions in addressing the unique challenges of IoT security.

Overall, the article's contribution lies in its innovative approach to IoT intrusion detection, exploration of synergistic solutions combining GNNs and XGBoost, and practical insights into implementing and deploying intrusion detection systems in IoT ecosystems. Through its findings and recommendations, the article aims to inform and inspire further research and development efforts to strengthen IoT network security and resilience against changing cyber threats.

1.4 Limitations

Although this study presents high success rates in IoT intrusion detection, it has certain limitations. First, the model's training time and computational cost increase, particularly in scenarios where GNN and XGBoost are used together. Therefore, optimizing these computational costs for real-time applications in large IoT networks is essential. Second, the model's generalization capability may be limited. Without further testing on diverse IoT networks and datasets, it isn't easy to ascertain whether the model will be effective across all IoT environments. Specifically, aspects such as multi-class attack detection and real-time performance must be explored more deeply in future research.

Finally, the datasets used in this study have concentrated on specific types of attacks. The model's performance should be tested extensively across different datasets and attack types to evaluate its effectiveness comprehensively.

1.5 Article Organization

This study is organized as follows: Current studies in the field are presented in the second section. The methodology of the model we address is described in the third section. Model architecture is presented in the fourth section. Experimental results obtained are presented in the fifth section. Finally, the Conclusion section summarizes the contributions of our paper and provides suggestions for future studies.

2. Related Work

This section will focus on IoT-based GNN and XGBoost IDS and examine relevant studies in the current literature. Given the complexity of IoT networks and the constantly evolving threat landscape, research on the effectiveness and reliability of such systems is of great importance. In this context, the capabilities of GNN and XGBoost-based approaches to detect security threats in IoT networks will be examined and evaluated, along with findings from previous studies.

Altaf et al. [7] introduced a deep learning-based IDS for IoT networks, utilizing a Node Edge-Graph Convolutional (NE-GConv) network. This approach employs Recursive Feature Elimination (RFE) to select 13 pertinent features, optimizing the model to effectively address IoT devices' resource constraints. The model enhances attack detection capabilities by integrating node and edge features, demonstrating significant improvements in computational efficiency and memory usage. A study [15] proposes deep machine learning techniques for developing an effective IDS targeting smart power grids against cyberattacks. The proposed IDS merges cyber-physical features collected from a practical trial platform, enabling the fusion of these features and adopts a GNN-based topology-aware model to utilize the spatial and temporal correlations in the data. Experimental results show that the proposed IDS performs superiorly to benchmark models lacking topology awareness that rely only on cyber or physical data. The study does not include detailed analysis or testing in real-world applications for further improvement of the proposed IDS's performance. Additionally, more information is needed regarding the proposed

IDS's generalization ability and applicability to different power systems. Moreover, a more detailed explanation of the data collection and modeling techniques used in the study could enhance the reproducibility of the research. The study [16] examines the use of GNNs for unsupervised intrusion and anomaly detection in computer networks, and an approach named Anomal-E is proposed. With this approach, attack patterns can be identified without using labeled data, and experiments show that Anomal-E significantly improves performance compared to other methods. However, further testing of Anomal-E's generalization ability and performance on real-world network traffic is required. Additionally, more research is needed on how Anomal-E can be more effectively used in large-scale networks. In another study [6], a new NIDS using GNNs was proposed. The GNN approach, named E-GraphSAGE, allows for capturing edge features and topological information in IoT networks. The performance of this system was demonstrated through extensive experimental evaluations on four different IoT NIDS benchmark datasets. These evaluations showed that the E-GraphSAGE-based NIDS surpassed the best-reported classifiers based on the F1-score criterion. For example, the F1-scores achieved in the NF-ToN-IoT and NF-BoT-IoT experiments were 1.0 and 0.97, respectively, indicating performance comparable to existing algorithms. Areas for improvement in this study include testing on a broader dataset and evaluating the generalization capability of the performance across different network scenarios. Additionally, exploring explainable graph neural network algorithms (such as GNN Explainer) to gain more insights into GNN model outputs and investigating neighborhood sampling techniques (especially irregular sampling techniques) to improve the runtime of the study are also considered important. Altaf et al. [17] introduced a concatenated Multigraph Neural Network (M-GNN) for detecting IoT intrusions, enhancing the capabilities of Network Intrusion Detection Systems. This novel GNN model utilizes a multi-edged graph structure to encapsulate comprehensive interactions between IoT nodes, effectively capturing spectral and spatial data characteristics. Extensive testing on multiple datasets showcases M-GNN's superior performance, demonstrating improvements in accuracy, precision, recall, and F1 scores by 2% to 5% over traditional GNN models. The research highlights the advantages of integrating multi-dimensional edge features and a complex graph topology, resulting in a more effective detection system with reduced model size and training time. Another study by Duan et al. [18] introduces a novel dynamic line graph neural network (DLGNN) method for network intrusion detection using semi-supervised learning. This approach captures both the spatial features of network traffic and the temporal dynamics between communication events, improving detection accuracy with fewer labeled samples. The model transforms network traffic into dynamic, spatiotemporal graphs, using a line graph structure to express edge relationships better and enhance message aggregation capabilities. Extensive tests on multiple datasets demonstrate superior performance over existing methods, particularly in multi-class detection scenarios.

Zivkovic [19] proposed an improved firefly (FA) optimization algorithm, CFAEESCA. The proposed improved metaheuristics are used to optimize the XGBoost classifier for the intrusion detection problem. The CFAEE-SCAXGBoost framework has been proposed, based on the XGBoost classifier, with its hyperparameters optimized and tuned using the newly proposed model, which outperforms the variation supported by the original FA algorithm, the PSO-XGBoost, and the basic implementation of the XGBoost, which is used in the comparative analysis. The experimental results show that the CFAEE-SCA-XGBoost model obtained the best accuracy compared to the original model and suggest the potential for using swarm intelligence algorithms for NIDS. Bhattacharya et al. [9] addressed the problem of IDS classification by proposing a hybrid machine learning model combining Principal Component Analysis (PCA), the Firefly algorithm, and XGBoost. Their framework involved initially transforming the IDS dataset using One-Hot encoding. Subsequently, a hybrid PCA-Firefly algorithm was employed for dimensionality reduction before applying the XGBoost algorithm to the reduced data to classify unanticipated cyberattacks. The experimental results presented in their study indicated that their proposed hybrid approach achieved higher accuracy than traditional methods. Abdulganiyu et al. [20] proposed the XIDINTFL-VAE framework, integrating a Class-Wise Focal Loss Variational AutoEncoder (CWFL-VAE) for targeted synthetic data generation with XGBoost for classification, to address the challenge of detecting minority class attacks in imbalanced network intrusion data. This work highlights the effectiveness of combining advanced data augmentation with robust ensemble learning to enhance the detection of rare intrusions and achieve superior performance in severe class imbalance. Song et al. [21] have proposed the WOA-XGBoost algorithm for intrusion detection, combining the XGBoost framework with the Whale Optimization Algorithm (WOA). This method innovatively utilizes WOA to automatically select and optimize XGBoost's parameters, offering a broader search range and improved accuracy compared to traditional manual or grid search optimization techniques. Evaluated on the KDD CUP 99 dataset, the WOA-XGBoost algorithm demonstrated significantly better performance than methods based on WOA-SVM, suggesting its potential as an effective tool for network data intrusion detection. Amaouche et al. [22] proposed IDS-XGbFS, a smart intrusion detection system. Their framework utilizes the XGBoost classifier with feature selection techniques, including Boruta and the Adaptive Synthetic Sampling Approach (ADASYN), to handle class imbalance. Evaluated on the NSL-KDD and 5RoutingMetrics datasets, their model demonstrated high accuracy, recall, and precision performance compared to other methods like CatBoost and CNN. The related work summary is shown in Table 1.

Table 1. Studies and findings

Study	Method	Findings
[15]	Cyber-Physical GNN-Based IDS	GNN-based IDS has demonstrated superior performance by integrating cyber and physical data with topological awareness.
[16]	Anomal-E: GNN-based unsupervised anomaly detection	Anomal-E, a GNN-based IDS, improves attack detection performance by leveraging unlabeled edge features and graph topological structure
[6]	E-GraphSAGE – Graph Sample and Aggregate GNN	E-GraphSAGE captures edge features and a network flow graph's topological pattern to enhance anomaly and attack detection performance.
[17]	Multigraph-GNN - A multi-edge graph structure	Multigraph-GNN shows an improved detection performance by processing multiple edges with multi-dimensional edge features in the graph structure.
[7]	Node Edge-GNN - Lightweight IDS	Node Edge-GNN performs enhanced anomaly detection in both payload content and network flow, considering the resource constraints of IoT devices.
[18]	Dynamic Line GNN	Dynamic Line GNN enhances detection accuracy by transforming network flows into dynamic, spatial, and temporal graphs with fewer labeled examples.
[19]	CFAEE-SCA-XGBoost	XGBoost with enhanced firefly algorithm CFAEESCA improves attack detection accuracy.
[9]	PCA-XGBoost	Utilizing XGBoost with an enhanced Principal Component Analysis algorithm enhances attack detection accuracy.
[20]	CWFL-VAE - XGBoost	Combining data augmentation with XGBoost improves the accuracy of anomaly detection.
[21]	WOA-XGBoost	XGBoost framework with the Whale Optimization Algorithm improves the accuracy of intrusion detection by automatically selecting and optimizing the parameters.
[22]	IDS-XGbFS	XGBoost with Boruta, selecting the most relevant features, and ADASYN, coping with the imbalanced dataset, provide improved intrusion detection accuracy.

Based on a review of existing literature, studies in network intrusion detection frequently demonstrate that integrating multiple techniques often yields superior performance compared to employing models in isolation. Hybrid approaches, combining core classifiers with methods such as feature selection, dimensionality reduction, data augmentation, or optimization algorithms, have been shown to effectively address challenges like severe class imbalance and complex feature spaces. These integrated methodologies apply structural modifications to features, more effective feature selection processes, or the strategic combination of models and data processing steps to leverage their respective strengths. Consequently, research indicates that these combined strategies improve detection accuracy and overall performance metrics.

3. Methodology

The hybrid model for IoT intrusion detection integrates the strengths of GNNs and XGBoost to fortify the defense mechanisms against cyber threats within IoT environments. The innovative aspect of this study is developing a hybrid model that combines the GNN and XGBoost algorithms. GNN effectively captures local patterns by modeling the complex relationships between devices and traffic flows in IoT networks. On the other hand, XGBoost is successful in processing tabular data and capturing nonlinear relationships between features.

The developed hybrid model demonstrates superior performance on graph-based and tabular data by combining GNN's capacity to learn network structures with XGBoost's robust classification capabilities. Additionally, the late fusion technique used in this model allows for higher accuracy rates by merging the predictions of both algorithms. Another innovation offered by this model is its ability to simultaneously address spatial dependencies and complex relationships by providing a solution based on both the graph structures and attributes of IoT data.

3.1 Graph Neural Networks (GNN)

GNN is a deep learning model used to process features in nodes of a graph structure [23]. GNNs capture geometric and topological features of entities by embedding relational inductive biases within their deep learning architectures [17]. A graph consists of nodes and the edges that connect these nodes. The main idea of GNN is to update feature vectors using neighborhood information and structural information in nodes. Feature vectors are determined for each node and edge. These

features represent the roles of nodes and edges within the graph. GNNs typically consist of several consecutive GNN layers. Each GNN layer updates the features of a node based on its neighbors' features and its features [24]. Aggregation functions are typically used to compute the feature vector of a node in the next layer using neighborhood information. These functions process the feature vectors collected from the node's neighbors and create a new feature vector to be transferred to the next layer [25]. If the feature vectors of neighboring nodes are represented as hi, the number of the layer is denoted as l, and the weights of the edges are omega alt i. j, s are ω_{ij} then the calculation of the feature vector of a node in the next layer can be represented as Equation 1.

$$\mathbf{h_i^{(l+1)}} = \boldsymbol{\sigma}\left(\boldsymbol{\Sigma_{j\in N_{(i)}}\omega_{ij}\cdot h_j(l)}\right) \quad (1)$$

The $N_{(i)}$, here represents the neighbourhood set of nodes i and σ is an activation function. This formula aggregates the feature vectors of neighboring nodes with weights and then passes them through an activation function to obtain the new feature vector. In general, a GNN model used to update the feature vector of a node can be expressed as Equation 2.

$$\mathbf{h_i^{(l+1)}} = \mathbf{Agg}(\left\{\mathbf{Update}\left(\mathbf{h_j^{(l)}}, \mathbf{x_j}\right)\right\}_{\mathbf{j\in N_{(i)}}}) \quad (2)$$

"Update" represents the function used to update the feature vector of a node. This function considers the information gathered from its neighbors to update the feature vector of a node and produces a new feature vector. The feature vector of each neighboring node is processed by Aggregation and Update operations to be transformed into an updated feature vector of the node. As a result, the feature vector of a node is updated with a combination of information from its neighbors and its features.

GNN iterates through these processes across multiple layers to process information in the graph structure. Each layer further processes the features in the nodes, enhancing the model's overall performance by iteratively refining the information.

In this study, GNN was utilized to build a model using the features present in the dataset. We did not incorporate a feature like the IP address into the GNN node structure. This decision was made because a feature such as an IP address is variable and can be altered by an attacker. GNNs can be employed as flexible and powerful modeling tools capable of utilizing structural information and features, enhancing their utility in various applications. The features are important in the dataset and do not necessarily have to be associated with nodes or edges. In cases where the features provide sufficient information to accurately predict a specific target, models like GNNs can be particularly effective. In the dataset we utilized, the relationships between rows or the interaction of features in columns are not directly apparent. Therefore, GNNs were employed solely to learn patterns and relationships among the data using the features themselves.

The GNN architecture employed in this study begins with an input layer consisting of nodes equal in number to the features in the dataset. This is followed by a first hidden layer comprising 64 neurons, utilizing the boost activation function to enable non-linear learning. The second hidden layer, also activated by ReLU, contains 32 neurons. These layers facilitate the gradual abstraction of representations learned from the data. Using the sigmoid activation function, the output layer maps the 32-dimensional input to a single output neuron, producing a probability value between 0 and 1. The Mean Squared Error (MSE) loss function is used to evaluate model performance, and the Adam optimization algorithm is employed for updating the weights. The architectural details of the GNN model used in this study are summarized in Table 2.

Table 2. GNN components and description

GNN Components	Description
Input Layer	input_dim = number of dataset features
Hidden Layer #1	input_dim : 64 neurons, activation function: ReLU
Hidden Layer #1	input_dim : 32 neurons, activation function: ReLU
Output Layer (Binary Classification)	32 to 1 neuron; Activation function: Sigmoid
Loss Function	Mean Squared Error
Optimizer	Adam

GNNs are relevant in IoT environments, particularly in network traffic analysis and intrusion detection. By modeling the interactions between IoT devices, sensors, and network components, GNNs can effectively detect anomalies, identify patterns of malicious behavior, and enhance the accuracy of IDSs. A GNN can be used to learn the normal interaction patterns of devices in an IoT network and detect deviations from these patterns. This can be utilized to detect potential security threats and enhance the accuracy of IDS.

3.2 Extreme Gradient Boosting (XGBoost)

XGBoost is an optimized implementation of the Gradient Boosting algorithm, an ensemble learning method [26]. XGBoost builds its predictions on decision trees, which are weak predictors, and aggregates the predictions of these trees. After calculating the prediction of each tree, the formula used by XGBoost to add its prediction to the prediction of the previous tree is shown in Equation 3.

$$\hat{\mathbf{y}}_{\mathbf{i}}^{(\mathbf{t})} = \hat{\mathbf{y}}_{\mathbf{i}}^{(\mathbf{t-1})} + \mathbf{f}_{\mathbf{t}}(\mathbf{x}_{\mathbf{i}}) \quad (3)$$

In Equation 3, $\hat{y}_i^{(t)}$, represents the prediction of tree t for instance i. $f_t(x_i)$, represents the prediction of example i for tree t based on the features x. This formula adds the prediction of the next tree to the prediction of the previous tree. In this way, XGBoost adds the residuals of each tree to the predictions of the previous trees, attempting to reduce the residuals in the total sum of consecutive trees. Each tree focuses on correcting the residuals of the previous trees. XGBoost's objective function is a measure of error that needs to be optimized during the model training. The general formula that calculates the total of this objective function is determined through gradients (first-order derivatives), second-order derivatives, and other terms. The general formula that calculates the total of XGBoost's objective function is as in Equation 4:

$$\boldsymbol{obj} = \boldsymbol{\Sigma_{i=1}^{n} l(\hat{y}_i, y_i) + \sum_{k=1}^{K} \Omega(f_k)} \quad (4)$$

The objective function is optimized to improve the accuracy of predictions and control the complexity of the model. This ensures the model is trained to have a low loss function value while protecting against overfitting. XGBoost exhibits effectiveness in handling feature-rich IoT datasets and intrusion detection tasks. Its ability to capture nonlinear relationships and high-dimensional feature spaces makes it well-suited to identify anomalous behavior patterns and detect intrusions in IoT environments.

3.3 Integration of GNNs and XGBoost in IoT Intrusion Detection

Integrating GNNs and XGBoost in IoT IDS presents a promising approach to bolstering security measures in IoT environments. By leveraging the complementary strengths of both methodologies, researchers aim to enhance the accuracy and efficiency of intrusion detection systems tailored to the specific requirements of IoT systems. This integration leverages GNNs to model the intricate relationships within IoT networks. Simultaneously, it utilizes XGBoost to distill complex features into potent predictors. This combined approach facilitates robust intrusion detection mechanisms. In conclusion, integrating GNNs and XGBoost in IoT intrusion detection represents a promising avenue for enhancing security measures in IoT environments. By harnessing the unique capabilities of GNNs to process graph data and capture complex relationships, coupled with the predictive power of XGBoost, researchers are paving the way for more robust and intelligent intrusion detection mechanisms tailored to the specific requirements of IoT systems. This integration holds significant promise for advancing the field of cybersecurity and ensuring robust protection for IoT systems against emerging threats.

3.4 Datasets

In our study, we utilized four different datasets. In this section, we briefly outline the characteristics of these datasets and specify the reasons for selecting them.

CICIoT-2023 [27]: Derived from 105 real IoT devices, the dataset is provided by the Canadian Institute for Cybersecurity (CIC). It encompasses a total of 33 attack types categorized into 7 classes. The training dataset consists of 466,868 records and 47 attributes. We chose this dataset for its currency, including real attacks and specificity to IoT.

CICIDS-2017 [28]: Produced by the CIC, this dataset contains benign and attack traffic. It includes 14 attack classes and 1 benign class. The training dataset comprises 2,827,876 records and 79 attributes. Given its widespread use in academic research, we used it as a benchmark for comparison.

UNSW-NB15 [29], [30], [31], [32], [33]: This dataset contains real modern activities along with synthetic contemporary attack behaviors. We utilized the test dataset comprising 82,332 records and 49 attributes. While not IoT-specific, it is commonly used in IDS applications, thus serving as a benchmark for our study.

IoMT – 2024 (CICIoMT2024) [34]: Developed for the Internet of Medical Things, this realistic dataset employs 25 real and 15 simulated IoT devices. Supporting various protocols, it includes a total of 18 distinct cyberattacks. Its selection was based on its contemporaneity and the representation of complex healthcare networks by combining real and simulated devices. Additionally, we utilized it to observe the impact of our study, particularly in real-world applications such as the medical field. Datasets and related cyberattacks in IoT are shown in Table 3.

Table 3. Datasets used in the study and attack classes in the data sets

Dataset name	Attacks	
CICIoT-2023	* DDoS	* Brute force
	* DoS	* Spoofing
	* Recon	* Mirai

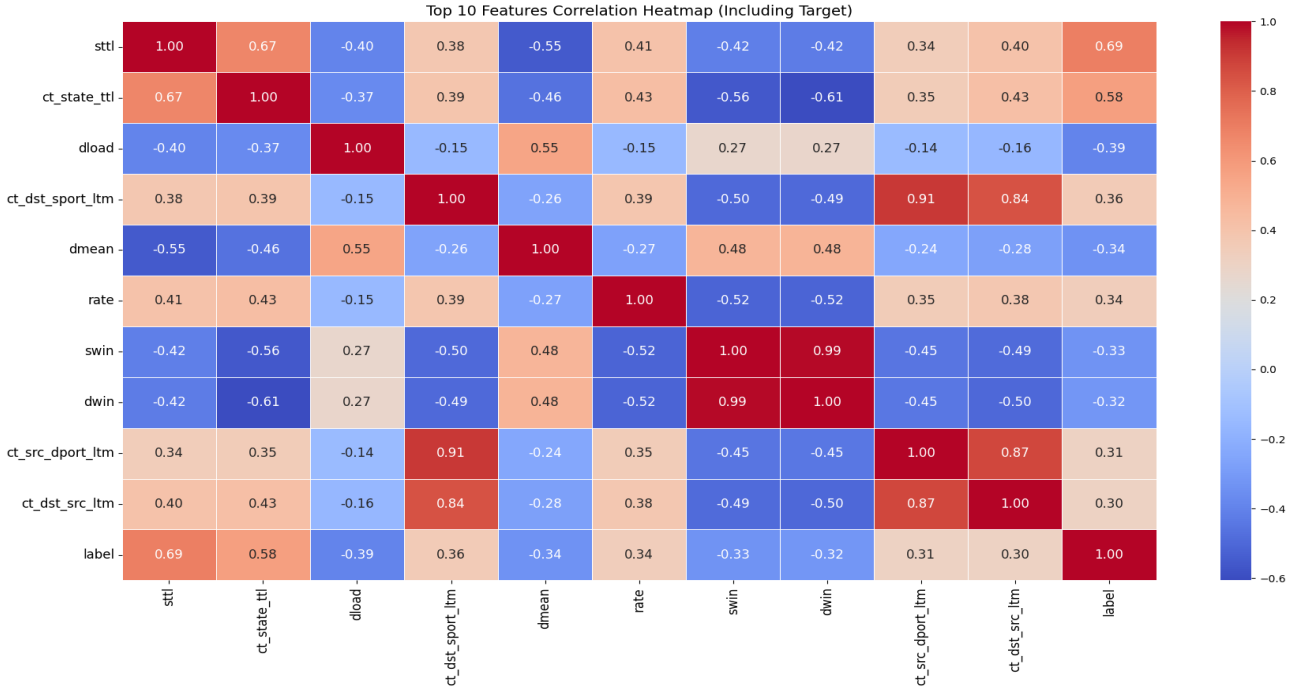
	* Web-based	
CICIDS-2017	* DoS * DDoS * Brute force * XSS	* SQL injection * Infiltration * Port scan * Botnet
UNSW-NB15	* Fuzzers * Analysis * Backdoors * DoS * Exploits	* Generic * Reconnaissance * Shellcode * Worms
IoMT – 2024	* ARP spoofing * Ping Sweep * Recon VulScan * OS Scan * Port Scan	* MQTT Malformed Data * MQTT DoS Connect flood * MQTT DoS Publish flood * MQTT DDoS Connect flood * MQTT DDoS Publish flood * DoS TCP/ICMP/SYN/ UDP * DDoS TCP/ICMP/SYN/UDP

3.5 Preprocessing and Feature Selection

Data preprocessing is a critical step for enhancing model performance. Particularly, data from IoT networks is often high-dimensional and irregular. Therefore, the preprocessing steps of standardization, handling the missing values, and encoding categorical features ensure that the model can derive accurate results from the data [35]. Scaling features help maintain all features on the same scale, aiding the model in producing high-performance results [36]. Additionally, converting categorical data into a numerical format allows machine learning algorithms to process this data effectively [37].

Standardization, handling missing values, and encoding categorical features for this dataset were conducted as preprocessing steps. As a preprocessing step, standardization involves scaling the features extracted from network traffic data to have a mean of 0 and a standard deviation of 1 [38]. This ensures that all features are on a similar scale, preventing any single feature from dominating others during model training. By bringing features to a comparable scale, standardization helps avoid biased results and aids in model convergence. Missing values, such as NaN or infinite values, were examined within the dataset and subsequently removed. This approach ensures data quality and prevents errors during model training. Dropping missing values is a common strategy, particularly when the number of missing values is relatively small compared to the dataset size [39]. Categorical features in the dataset underwent label encoding, which converts categorical variables into numerical representations. This transformation makes the data compatible with machine learning algorithms, which typically operate on numerical inputs [40]. Encoding categorical features enables the model to process and learn from these features effectively. Another process carried out during the preprocessing stage is the selection of the top 10 features. The best features selected during the XGBoost phase and the heatmap related to these features are shown in Figure 1.

Figure 1. The top 10 features and the heatmap related to these features



These preprocessing steps are crucial for preparing the input data for training the hybrid model. Standardization aids in convergence during model training and mitigates issues related to varying feature scales. Handling missing values ensures that the model learns from complete and accurate data, enhancing its performance. Encoding categorical features enables the model to utilize these features in learning. Overall, these preprocessing steps contribute to the robustness and effectiveness of the hybrid model for IoT intrusion detection. Feature selection enhances the model's performance by considering only the most meaningful features [41]. It has been observed that certain features are more effective in detecting attacks in IoT traffic analysis. This leads to faster model training and reduces errors arising from unnecessary features.

4. Model Architecture

In our study, we employed an architectural framework depicted in Figure 2 to illustrate the working principle. Our study consists of three stages: pre-processing, multi-model training, and fusion & prediction.

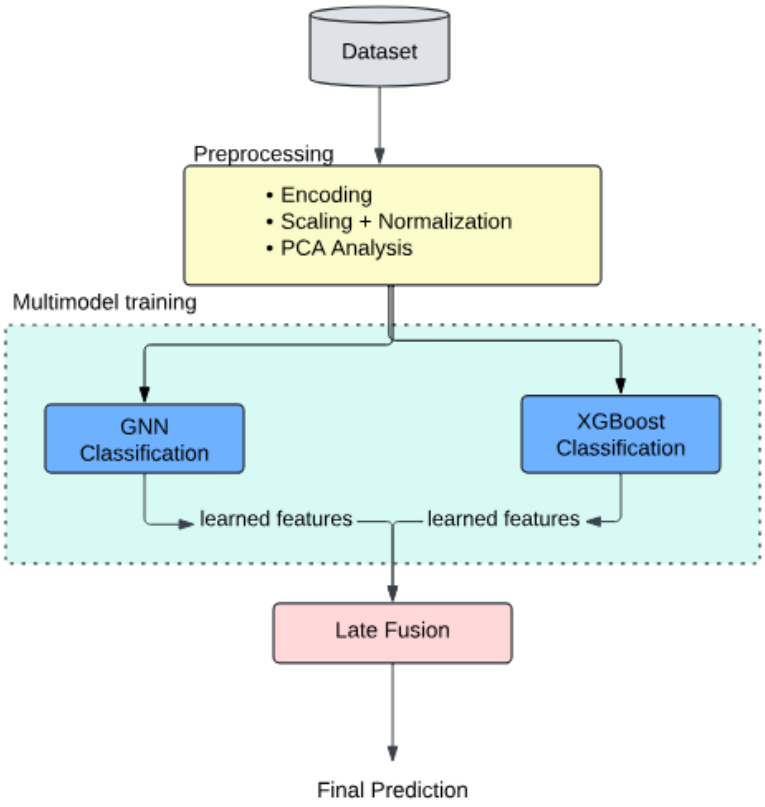


Figure 2. Architectural Framework for Proposed Model

The pseudo-algorithm of the model we developed is provided in Algorithm 1.

Algorithm 1:
The algorithm for GNN and XGBoost-based IDS design

INPUT

- $D = \{(x_i, y_i)\}_{i=1}^{N}$ *be the raw dataset where* $x_i \in \mathbb{R}^d, y_i \in \{0,1\}$.
- $\mathbf{X} \in \mathbb{R}^{\{N \times d\}}$ be the data matrix, and $\mathbf{y} \in \{0,1\}^N$ be the label vector

Preprocessing

- Remove irrelevant features.
  - $\boldsymbol{X} \leftarrow Drop\ (\boldsymbol{X}, irrelevant\ columns)$
- Handle missing values (e.g., imputation):
  - $\boldsymbol{X}_{imputed} = I(\boldsymbol{X})$
- Encode categorical features:

- $\boldsymbol{X}_{enc} = \mathcal{L}(\boldsymbol{X}_{imputed}), where\ \mathcal{L}\ is\ Label\ Encoding$
- Standardize features:
  - $\mathbf{X}_{\text{std}} = \frac{(\mathrm{X}_{\text{enc}} - \boldsymbol{\mu})}{\boldsymbol{\sigma}}$, where $\boldsymbol{\mu} = \mathbb{E}[\mathbf{X}], \boldsymbol{\sigma} = \text{std}(\mathbf{X})$
- Feature selection:
  - $\mathbf{X}_{\text{fs}} = \mathcal{S}(\mathbf{X}_{\text{std}})$
- Dimensionality reduction via PCA:
- $\mathbf{X}_{\text{pca}} = \mathbf{X}_{\text{fs}} \cdot \mathbf{P}_{\text{k}},$ where $\mathbf{P}_{\text{k}} \in \mathbb{R}^{d \times k},\ \text{k} < \text{d}$

GNN Architecture

Let the GNN be a function $\mathrm{f}_{\theta}: \mathbb{R}^{\mathrm{k}} \rightarrow \mathbb{R}$ where θ are learnable parameters

- Input: $\mathrm{x}_{\mathrm{i}} \in \mathbb{R}^{\mathrm{k}}$
- Hidden layers:
  - $\mathbf{h}^{\mathrm{l}} = \text{ReLU}\left(\mathbf{W}^{\mathrm{l}}\mathbf{h}^{\mathrm{l}-1} + \mathbf{b}^{\mathrm{l}}\right)$
- Output:
  - $\hat{\mathrm{y}}_{\mathrm{i}}^{\text{GNN}} = \mathrm{f}_{\theta}(\mathrm{x}_{\mathrm{i}})$
- Loss function (MSE):
  - $\mathcal{L}_{\text{GNN}} = \left(\frac{1}{\mathrm{N}}\right)\sum_{\{\mathrm{I}=1\}}^{\{\mathrm{N}\}}\left(\mathrm{y}_{\mathrm{I}} - \hat{\mathrm{y}}_{\mathrm{I}}^{\text{GNN}}\right)^{2}$
- Training via gradient descent with Adam optimizer:
  - $\theta \leftarrow \theta - \eta \cdot \nabla_{\theta}\mathcal{L}_{\text{GNN}}$, for each epoch

XGBoost Classifier

- Train a gradient boosting model $\mathrm{g}_{\phi}: \mathbb{R}^{\mathrm{k}} \rightarrow [0,1]$:
  - $\hat{\mathrm{y}}_{\mathrm{i}}^{\text{XGB}} = \mathrm{g}_{\phi}(\mathrm{x}_{\mathrm{i}})$
- Convert probabilities to binary prediction:
  - $\tilde{\mathrm{y}}_{\mathrm{i}}^{\text{XGB}} = \begin{cases} 1 & \text{if } \hat{\mathrm{y}}_{\mathrm{i}}^{\text{XGB}} \geq \tau \\ 0 & otherwise \end{cases}$

Late Fusion

- Define fusion weights:
  - $\alpha, \beta \in [0,1], \text{with } \alpha + \beta = 1$
- Combine predictions:
  - $\hat{\mathrm{y}}_{\mathrm{i}}^{\text{fused}} = \alpha \cdot \hat{\mathrm{y}}_{\mathrm{i}}^{\text{GNN}} + \beta \cdot \hat{\mathrm{y}}_{\mathrm{i}}^{\text{XGB}}$
- Final binary prediction:
  - $\tilde{\mathrm{y}}_{\mathrm{i}}^{\text{fused}} = \begin{cases} 1 & \text{if } \hat{\mathrm{y}}_{\mathrm{i}}^{\text{fused}} \geq \tau \\ 0 & otherwise \end{cases}$

OUTPUT

- $\tilde{\mathrm{y}}^{\text{fused}} = \left\{\tilde{\mathrm{y}}_{1}^{\text{fused}}, \ldots, \tilde{\mathrm{y}}_{\mathrm{N}}^{\text{fused}}\right\}$

This pseudo-algorithm outlines our model's workflow, from pre-processing the data to training multiple models and finally fusing their predictions for enhanced performance. During the preprocessing stage, irrelevant fields like IDs were dropped, categorical values were converted to numerical representations using Label Encoding, standardization was applied, and dimensionality reduction was performed using PCA, preparing the dataset for machine learning. Subsequently, the dataset was split into two groups: training and testing. In the Multi-Model training stage, training was initially conducted using GNN on the training dataset, and then the XGBoost algorithm was applied to the obtained features. The values obtained from both

training were combined using the late fusion technique. With late fusion, new, more accurate, and reliable decisions are produced by combining the decisions of each classifier [9]. This study used the Weighted Average Ensemble method as the Late fusion technique. It is expressed using a formula to calculate each model's weighted sum of predictions. Given the predictions of XGBoost and GNN models as y_{XGB} and y_{GNN} respectively, the Late Fusion method can be calculated as shown in Equation 5.

$$\mathbf{y_f = w_1 * y_{XGB} + w_2 * y_{GNN}} \quad (5)$$

In this context, y_f represents the merged predictions, while w_1 and w_2 denote the weights of the respective models. The weights are selected to ensure that their sum equals 1, with $w_1 = 0.5$ and w_2=0.5. Late Fusion employs a weighted approach when combining predictions from different models, aiming to leverage their diverse strengths and weaknesses to create a more robust predictor. It allows each model to be trained and optimized independently, offering flexibility tailored to its dataset and parameters, potentially achieving better performance.

5. Experimentation and Results

Performance metrics are essential to understanding how well a model performs. These metrics determine how accurately the model predicts and where its errors lie [42]. In this work, we used assessment metrics like accuracy, precision, recall, F1-score, and AUC to assess the performance of our constructed model. A confusion matrix depicts the link between the actual and anticipated classes, a tool frequently used to evaluate the effectiveness of a complex classification model. This table consists of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) terms. TP represents the number of true positive instances correctly predicted as positive by the model. TN represents the number of true negative instances correctly predicted as negative. FP indicates the number of negative instances incorrectly predicted as positive by the model, while FN indicates the number of positive instances incorrectly predicted as negative. Accuracy expresses the ratio of correctly predicted instances to the total number of instances [43] and is expressed by Equation 6.

$$\mathbf{accuracy} = \frac{\mathbf{TP+TN}}{\mathbf{TP+TN+FP+FN}} \quad (6)$$

Precision, the ratio of true positive instances correctly predicted as positive to all instances predicted as positive, is defined by Equation 7.

$$\mathbf{precision} = \frac{\mathbf{TP}}{\mathbf{TP+FP}} \quad (7)$$

Recall that the ratio of true positive instances correctly predicted as positive to all actual positive instances is defined by Equation 8.

$$\mathbf{recall} = \frac{\mathbf{TP}}{\mathbf{TP+FN}} \quad (8)$$

F1-score, the harmonic mean of precision and recall, is calculated using Equation 9.

$$\mathbf{f1-score} = \mathbf{2} * \frac{\mathbf{precision*recall}}{\mathbf{precision+recall}} \quad (9)$$

ROC-AUC (Receiver Operating Characteristic - Area Under Curve) represents the area under the curve of the graph where the recall and specificity (1 - false positive rate) [44] change at different threshold values. The AUC value ranges from 0 to 1, with a higher AUC indicating better model performance [45].

CICIoT-2023 evaluation: pre-processing steps were performed first in our experiment using the CICIoT-2023 dataset. Then, the top 10 attributes were selected using GNN. Here, the nodes represent the selected attributes, not the IoT devices.

After that, these attributes learned from GNN were given as input to the XGBoost algorithm. The evaluation of the CICIoT-2023 dataset is illustrated in the confusion matrix shown in Figure 3.

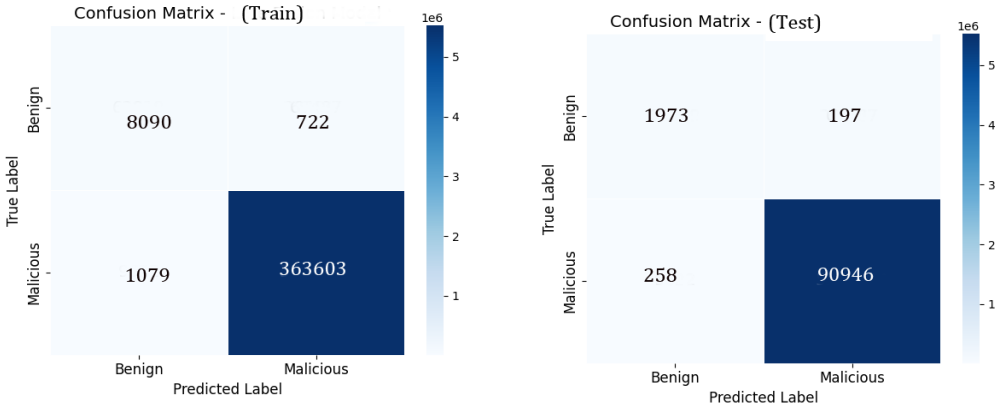


Figure 3. Training and Testing Confusion Matrix for CICIoT-2023.

The high values of both TP and TN indicate that the model correctly predicts both positive and negative classes. With a low FP value, we can say that the model has a low tendency to predict negative classes as positive incorrectly. However, the FN value is also notable, as there is a tendency to incorrectly predict positive classes as negative, although it is lower than FP. The ROC-AUC curve obtained from the model training is shown in Figure 4.

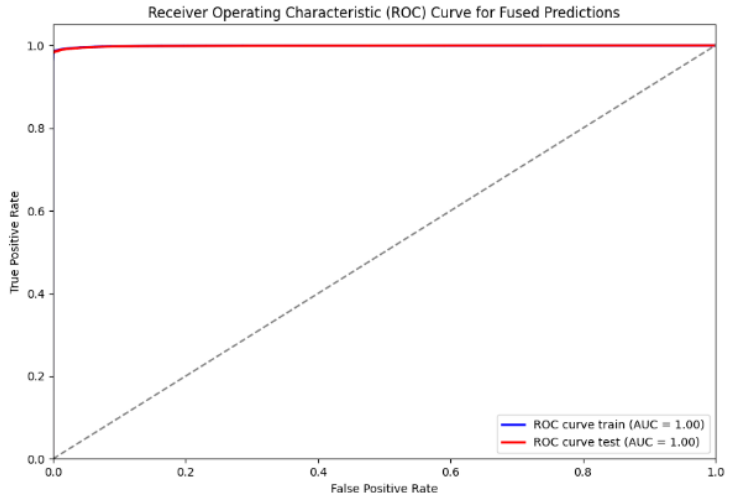


Figure 4. The ROC Curve for CICIoT-2023 Dataset Training Model

The point near the top-left corner of the graph indicates that your model achieves high precision and recall, meaning it accurately detects attacks while minimizing false alarms. The AUC value 1.0 signifies that your model demonstrates excellent discrimination, meaning it can reliably distinguish attacks from non-attack events [46]. These results are highly favorable for an IoT IDS because security is critical in IoT environments, and missing false alarms or attacks can have serious consequences. A high-performing IDS is crucial for protecting IoT devices and networks. Therefore, our results indicate that our model effectively detects attacks and is a robust tool for enhancing IoT security.

The precision, recall, f1-score, and AUC values of our developed model in the training and testing phases are provided in Table 4.

Table 4. Training and testing performance metrics for CICIoT-2023

	Accuracy	**Precision**	**Recall**	**f1-score**	**AUC**
Training	0.9952	0.9980	0.9970	0.9975	0.9988
Test	0.9951	0.9978	0.9972	0.9975	0.9988

Our model accurately classified almost all examples in the training and test datasets. This high accuracy indicates that our model performs well overall. According to the precision value, the rate of true positive predictions seems quite high. The high recall value indicates that our model tends to minimize the number of false negatives. High AUC values suggest that our model effectively distinguishes between positive and negative classes. Overall, based on the performance metrics we have considered, it can be said that our model performs quite well. With high accuracy, recall, precision, and f1-score in both the training and test sets, it is evident that our model reliably detects attacks and is generally effective.

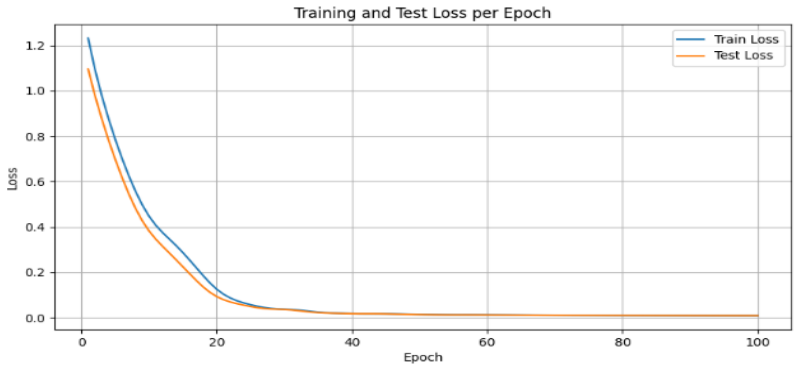


Figure 5. Training and testing loss

In our model's training and test phases, the loss amounts obtained initially with random weight values and after 100 epochs are provided, as shown in Figure 5. It can be observed that the values obtained for GNN are close to zero. The loss rate significantly decreased during the first 20 epochs. In the subsequent epochs, it can be seen that the losses for both training and testing settle into a lower balance. The significant decrease in loss during the first 20 epochs indicates that our model adapts better to the training data and initially deviates significantly from random weights. The subsequent epochs' loss settling into a lower balance demonstrates that the model exhibits a more consistent and balanced performance on training and test data. The reduction in the difference between training and test loss amounts indicates a decrease in the model's overfitting risk and improved generalization ability. This implies that the model can perform well on new and unseen data.

IoTMT – 2024 Evaluation: It is crucial to assess the performance of a model on real-world data in more detail. Therefore, an evaluation was conducted on a real-world application dataset, the Internet of Medical Things dataset. The confusion matrix for the evaluation conducted for the Internet of Medical Things is illustrated in Figure 6.

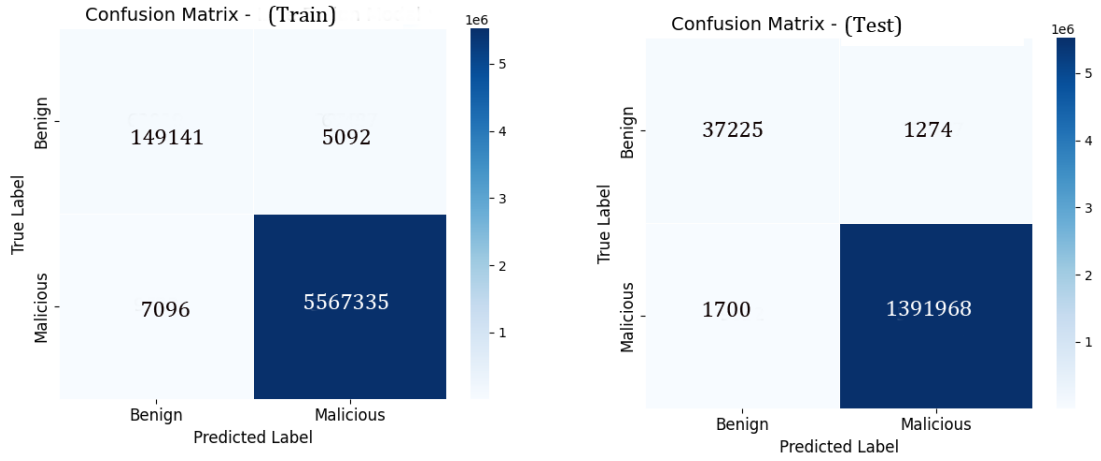


Figure 6. Training and testing the confusion matrix for the IoMT dataset.

The performance metrics measured within the scope of the study are presented in Table 5.

Table 5. Training and testing performance metrics for IoMT-2024

	Accuracy	Precision	Recall	f1-score	AUC
Training	0.9979	0.9991	0.9987	0.9989	0.9996
Test	0.9979	0.9991	0.9988	0.9989	0.9996

Considering these values, we can say that the IDS exhibits a very high accuracy, precision, and F1 score. High accuracy and precision values demonstrate the system's ability to accurately classify normal and attack traffic, while the high F1 score indicates a balanced combination of these two metrics. Our model achieved high accuracy for this dataset in the training and test sets. This indicates the overall success of our model in classifying traffic as either attack or benign. The rate of correctly identifying samples predicted as attacks is quite high, indicating a tendency to minimize false positives.

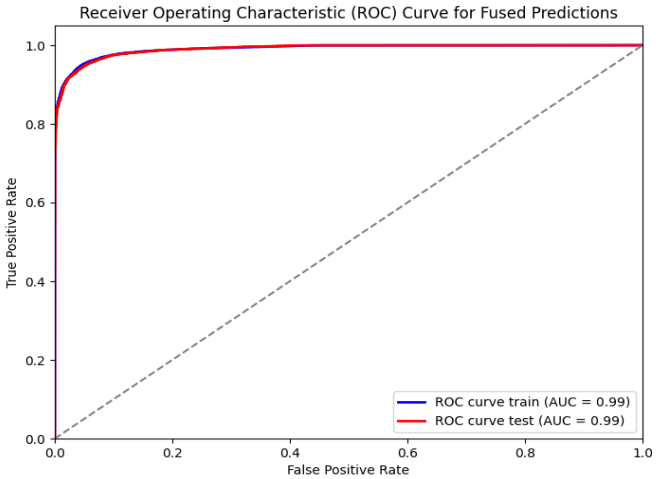


Figure 7. The ROC curve for the IoMT-2024 dataset training model

The area under the ROC curve, represented by high AUC values, indicates that the model can effectively distinguish between positive and negative classes (Figure 7).

CICIDS-2017 Evaluation: When we evaluated the performance of our study on the CICIDS-2017 dataset, the results obtained are shown in Figure 8 for the binary confusion matrix and in Table 6 for the performance metrics, respectively.

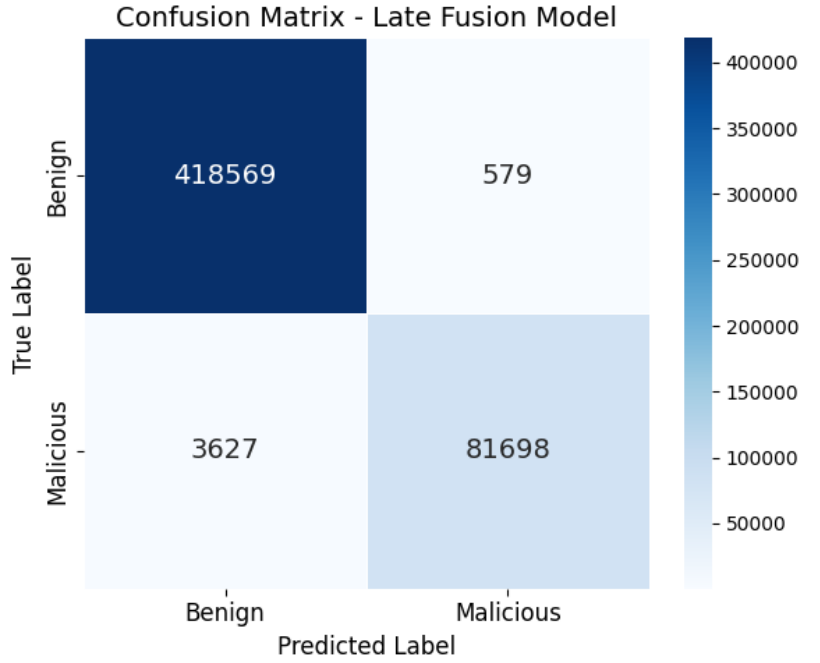


Figure 8. Confusion matrix for the CICIDS-2017 dataset.

Table 6. Training and testing performance metrics for CICIoT-2023

	Accuracy	**Precision**	**Recall**	**f1-score**	**AUC**
Training	0.9828	0.9734	0.9382	0.9555	0.9800
Test	0.9917	0.9930	0.9575	0.9749	0.9781

UNSW-NB15 Evaluation; We evaluated our study on the UNSW-NB15 dataset for benchmarking. The binary confusion matrix and performance metrics obtained are shown in Figure 9 and Table 7, respectively.

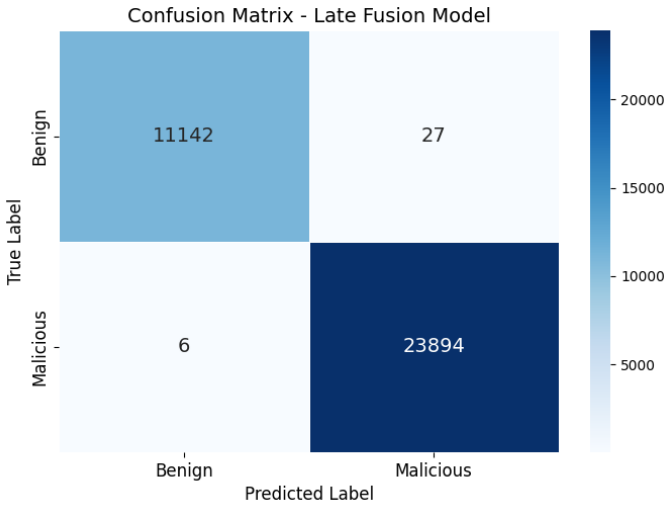


Figure 9. Training and testing confusion matrix for UNSW-NB15 dataset

Table 7. Training and testing performance metrics for the UNSW-NB15 dataset

	Accuracy	Precision	Recall	f1-score	AUC
Training	0.9862	0.9912	0.9882	0.9899	0.9900
Test	0.9991	0.9989	0.9997	0.9987	0.9987

5.1. Memory Usage and Time Consumption

In machine learning-based IDS, achieving high predictive performance is only part of the challenge. For practical deployment, particularly in real-time or resource-constrained environments such as edge devices or high-throughput networks, it is essential to assess how much time and memory a model requires during training and inference [47], [48]. Without such evaluations, even high-performing models may be unsuitable for operational use. Therefore, this section provides a comparative analysis of the computational cost of the proposed GNN and XGBoost-based IDS design. Specifically, we measured the training time and memory consumption across four benchmark datasets to examine the feasibility of real-time deployment.

The training times and memory usage of our proposed hybrid model on different datasets are presented in Table 8.

Table 8. Training Time comparison.

Dataset	Samples	Features	GNN Time (sec)	XGBoost Time (sec)	GNN Memory Usage (MB)	XGBoost Memory Usage (MB)
UNSW-NB15	175,341	49	2.05	1.79	113.08	99.59
CICIoMT2024	6,956,726	46	133.72	21.43	50.70	207.94
CICIoT-2023	45,019,243	38	2239.34	85.90	5188.84	2107.34
CICIDS-2017	2,830,743	79	39.03	7.24	1503.06	135.52

This study proposes a hybrid intrusion detection model combining GNN and XGBoost, designed to effectively address the challenges of detecting cyber threats in complex and dynamic IoT environments. The model was evaluated on multiple benchmark datasets, which vary in size, feature complexity, and attack diversity. Experimental results highlight both the strengths and limitations of the approach in terms of computational time and memory usage.

The results show that the GNN component demonstrates relatively low inference time on small datasets such as UNSW-NB15 (2.05 sec) and acceptable levels for moderate datasets like CICIDS-2017 (39.03 sec). However, processing time increases substantially in large-scale datasets such as CICIoT-2023 (2239.34 sec), which is expected due to GNN's graph-based representation and learning complexity. XGBoost, in contrast, consistently offers lower processing time across all datasets (e.g., 85.90 sec for CICIoT-2023), highlighting its efficiency and suitability for latency-sensitive applications. From a memory consumption perspective, the results reveal a nuanced pattern. While GNN tends to consume more memory in large datasets (e.g., 5188.84 MB in CICIoT-2023), it surprisingly uses significantly less memory than XGBoost on the IoMT-2024 dataset (50.70 MB vs. 207.94 MB). This variability suggests that memory demand depends not solely on dataset size but also on feature structure, model complexity, and internal data representations.

The combined use of GNN and XGBoost inevitably results in a higher overall resource footprint, especially regarding memory, as seen in high-scale datasets. Despite this, the hybrid model remains viable for IoT security, particularly when deployed at fog or gateway nodes, where moderate computational and memory resources are available. These deployment strategies allow the system to benefit from the complementary strengths of GNN (deep structural representation) and XGBoost (efficient, interpretable decision boundaries), offering a robust and scalable solution for real-time intrusion detection in IoT ecosystems.

Although the hybrid approach introduces increased resource demands, its superior detection capabilities and adaptability to heterogeneous IoT data justify its application in environments where edge intelligence or hierarchical resource distribution can be leveraged.

5.2. Qualitative Comparison with Recent Literature Studies

The CICIDS-2017, UNSW-NB15 and CICIoT-2023 datasets are extensively studied datasets. The IoMT-2024 dataset has also begun to attract the attention of researchers with its wide range of attack variations. Therefore, in our study, these datasets were used as benchmarks to evaluate our proposed model's performance and compare it with previous studies. High accuracy and F1-score values were obtained for both datasets.

For performance comparison, the accuracy (Acc.) and f1-score results obtained from studies focusing on hybrid approaches conducted in the last two years for the CICIDS-2017 dataset are provided in Table 9.

Table 9. Studies conducted using the CICIDS-2017 dataset in the last two years and their performance metrics

Paper	Methodology	Acc.	F1-score
[49]	GCN-BiLSTM-Attention	> 95.0	94.36
[50]	Decision Tree	> 90.0	96.88
[51]	CNN	98.61	98.09
[52]	Novel CNN	-	98.7
[53]	Bagging Ensemble-Based DNN	98.74	99.86
[54]	CBCO-ERNN	98.83	99.38
[55]	CNN-BiLSTM	99.76	98.50
[56]	Res-TranBiLSTM	99.15	-
[57]	BLoCNet	98	98
Our study	GNN + XGBoost	**98.32**	**95.66**

As presented in Table 9, our proposed model demonstrates competitive performance with an accuracy of 98.32% and an F1-score of 95.66%, outperforming several traditional approaches. However, certain studies report even higher performance metrics. This discrepancy arises from factors such as their use of dataset balancing techniques (which we did not employ) and their deployment of deep, complex architectures focused on maximizing accuracy. While these approaches can achieve higher scores, they often incur significant computational costs, making them less suitable for resource-constrained IoT environments where our study prioritized efficiency and practicality. We achieved strong performance with reduced complexity by employing a lightweight late fusion strategy (GNN + XGBoost). Thus, while some methods report slightly higher results, our model offers a more practical balance of performance and computational cost for real-world IoT.

The accuracy and F1-score achievements obtained from studies conducted in the last two years for the UNSW-NB15 dataset are provided in Table 10.

Table 10. Studies conducted using the UNSW-NB15 dataset in the last two years and their performance metrics

Paper	Methodology	Acc.	F1-score
[58]	GMM-WGAN-IDS	87.70	85.44
[59]	CNN + LSTM	87.6	88
[60]	VGG19 (CNN)	93.56	92
[61]	SAIDS (XGBoost+KNN+RF)	96.24	96.29
[62]	RF	90.1	90.0
[63]	DenseNet	98.6	98.7
Our study	GNN + XGBoost	**98.46**	**98.87**

The accuracy and F1-score achievements obtained from studies conducted for the CICIoT-2023 dataset are provided in Table 11.

Table 11. Studies conducted using the CICIoT-2023 dataset in the last two years and their performance metrics

Paper	Methodology	Acc.	F1-score
[64]	LSTM-Based	98.75	98.59
[65]	CNN-based	99.1	99.05
[66]	SSK-DDoS	89.05	-
[67]	Blending	99.51	99.07
[68]	EnsAdp_CIDS	98.93	99.45
[69]	AUWPAE	99.33	98.88
Our study	GNN + XGBoost	**99.51**	**99.75**

The accuracy results achieved in the experiments using the IoMT-2024 dataset are presented in Table 12.

Table 12. Studies conducted using the IoMT-2024 dataset in the last two years and their performance metrics

Paper	Methodology	Acc.	F1-score
[70]	Random Forest-Based	94.97	95
[71]	MA-DeepCRNN	99.12	99.12
[72]	DL LSTM	98	98
[73]	ROC with RoS	99.7	99.6
[74]	Random Forest-Based	99.22	66.09
[75]	CNN Based	95.63	95.16
Our study	GNN + XGBoost	**99.51**	**99.75**

When a general evaluation is made for four different datasets, it is seen that our proposed model shows high accuracy and F1 score. Considering the UNSW-NB15 and CICIDS-2017 datasets, it is seen that our proposed model is above average with higher performance. From the perspective of the more recent CICIoT-2023 dataset, it is observed that our model achieves slightly higher F1-score and accuracy compared to the referenced studies. Lastly, when considering the more recent IoMT dataset, it can be seen that our proposed model demonstrates quite high performance.

6. Discussion

The proposed hybrid model in this study significantly impacts IoT intrusion detection. Its ability to adapt to the dynamic nature of IoT environments presents a considerable advantage over traditional intrusion detection systems. The GNN has effectively modeled the relationships between devices in IoT networks, successfully capturing attack patterns, while XGBoost has improved classification performance by learning the nonlinear relationships among features.

Our hybrid model, which combines the GNN and XGBoost algorithms, has demonstrated greater performance in intrusion detection within IoT networks compared to previous studies. A recent study by [7] featuring the NE-GConv model offers a resource-friendly approach for IoT devices; however, it has not achieved our proposed model's accuracy and precision rates.

The E-GraphSAGE-based model developed by [76] has also successfully captured edge features, particularly in IoT networks. However, our proposed model provides higher accuracy rates by incorporating the strong classification capabilities of XGBoost.

Our work exhibits higher accuracy, precision, and F1 performance scores than the most recent techniques and other hybrid models tested on various datasets.

The study's results clearly show strong performance in detecting attacks in IoT-based networks, achieving high accuracy, precision, and F1 scores. Tests conducted on real-world datasets such as IoMT-2024 and CICIoT-2023 indicate that the model is suitable for practical applications. This suggests that the model could provide security solutions in various domains, including IoT-based healthcare, smart cities, and industrial IoT.

However, further testing of the model's real-time intrusion detection capabilities and scalability is necessary. In the future, evaluating the model's performance in larger and more complex IoT networks will be beneficial. Additionally, the results of the system in multi-class attack detection could also be investigated.

6. Conclusion

This study proposes a hybrid model combining Graph Neural Networks and the XGBoost algorithm to develop a robust IDS against cyber threats in IoT environments. The proposed model benefits GNNs to model complex relationships and features while analyzing and predicting complex features with the XGBoost algorithm. The study evaluates the model's effectiveness on different datasets, such as CICIoT-2023, CICIDS-2017, UNSW-NB15, and IoMT-2024. The results show that the proposed hybrid model can detect attacks with high accuracy, precision, and recall values. Additionally, it is identified that factors such as training time, which were not considered during the study, are important for future research. This study provides an innovative and effective approach to enhancing IoT security and a guiding framework for future research.

Furthermore, the top 10 features are selected in this study, and the model's performance is evaluated based on these selected features. Experiments conducted on a broader feature set and comparing results can provide a valuable roadmap for future studies. Additionally, it is noted that factors like training time were not considered, indicating a limitation that could be addressed in future evaluations, considering cost parameters such as training time and memory consumption.

The utilization of the IoMT dataset contributes significantly to field experience. However, using datasets from different sectors to assess the model's applicability with real data from other domains is advisable.

While our study evaluates attack and benign traffic scenarios, it's crucial to consider multi-class prediction involving the classification of different attack types. Future studies can thus focus on developing methods to detect and classify different attack types.

Nowadays, there is an increasing focus on multi-class attack classification to ensure the security of IoT systems. The various types of attacks encountered in IoT environments, such as DoS, DDoS, man-in-the-middle, and malware, exhibit different characteristics, making it essential to develop models to classify these attacks accurately. In this context, developing a hybrid design using a combination of XGBoost and GNN, along with evaluations performed on four different datasets, represents a significant step toward enhancing the effectiveness of multi-class attack classification. This approach provides in-depth information for a more comprehensive classification of attack types and can be supported by feature engineering and hyperparameter optimization techniques.

On the other hand, integrating alternative algorithms such as autoencoders and reinforcement learning holds the potential for improving attack detection accuracy. In particular, hybrid systems utilizing XGBoost and GNN can be employed better to understand the relationships and structure of the data. Autoencoders effectively detect anomalous behavior by obtaining low-dimensional representations of the data, while reinforcement learning can be used to adapt to the dynamic conditions of the environment. The integration of these methods presents opportunities to enhance the success of the XGBoost + GNN model.

The applicability of this hybrid model in edge computing environments has become a critical requirement for real-time attack detection. Edge computing reduces network latency by enabling data to be processed closer to its source, providing quick response times. Given the continuous data streams from IoT devices, these rapid response times are crucial for minimizing the impact of attacks. Integrating the hybrid model into edge computing architectures can improve the efficient use of resources and scalability, resulting in lower energy consumption and bandwidth savings.

To evaluate the performance of the developed hybrid model, metrics such as accuracy, precision, recall, F1 score, and ROC curve are employed. These indicators are crucial in assessing how well the attack detection system works. Additionally, conducting cross-validation methods and trials on different datasets will help understand the model's generalization capability. Evaluations performed on four datasets highlight the model's performance under various conditions.

In conclusion, developing the XGBoost + GNN hybrid model presents an innovative approach for multi-class attack classification. Future studies should focus on using deep learning techniques to increase the complexity of the model and provide more innovations in detecting more complex attack types. Furthermore, integrating artificial intelligence algorithms into edge computing for real-time attack detection and testing this model's performance on more datasets should be among the future research directions. It is necessary to continuously update and adapt the systems to develop more innovative and effective solutions for the security of IoT devices. These efforts hold great potential for enhancing security in IoT systems and contribute to developing modern security solutions.

References

[1] K. V. V. N. L. Sai Kiran, R. N. K. Devisetty, N. P. Kalyan, K. Mukundini, and R. Karthi, ‘Building an Intrusion Detection System for IoT Environment using Machine Learning Techniques’, *Procedia Computer Science*, vol. 171, pp. 2372–2379, 2020, doi: 10.1016/j.procs.2020.04.257.

[2] G. A. Mukhaini, M. Anbar, S. Manickam, T. A. Al-Amiedy, and A. A. Momani, ‘A systematic literature review of recent lightweight detection approaches leveraging machine and deep learning mechanisms in Internet of Things networks’, *Journal of King Saud University - Computer and Information Sciences*, vol. 36, no. 1, p. 101866, Jan. 2024, doi: 10.1016/j.jksuci.2023.101866.

[3] E. Anthi, L. Williams, M. Slowinska, G. Theodorakopoulos, and P. Burnap, ‘A Supervised Intrusion Detection System for Smart Home IoT Devices’, *IEEE Internet Things J.*, vol. 6, no. 5, pp. 9042–9053, Oct. 2019, doi: 10.1109/JIOT.2019.2926365.

[4] A. Nazir *et al.*, ‘Advancing IoT security: A systematic review of machine learning approaches for the detection of IoT botnets’, *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 10, p. 101820, Dec. 2023, doi: 10.1016/j.jksuci.2023.101820.

[5] I. Cvitić, D. Peraković, M. Periša, and B. Gupta, ‘Ensemble machine learning approach for classification of IoT devices in smart home’, *Int. J. Mach. Learn. & Cyber.*, vol. 12, no. 11, pp. 3179–3202, Nov. 2021, doi: 10.1007/s13042-020-01241-0.

[6] W. W. Lo, G. Kulatilleke, M. Sarhan, S. Layeghy, and M. Portmann, ‘XG-BoT: An explainable deep graph neural network for botnet detection and forensics’, *Internet of Things*, vol. 22, p. 100747, Jul. 2023, doi: 10.1016/j.iot.2023.100747.

[7] T. Altaf, X. Wang, W. Ni, R. P. Liu, and R. Braun, ‘NE-GConv: A lightweight node edge graph convolutional network for intrusion detection’, *Computers & Security*, vol. 130, p. 103285, Jul. 2023, doi: 10.1016/j.cose.2023.103285.

[8] K. Qian, H. Yang, R. Li, W. Chen, X. Luo, and L. Yin, 'Distributed Detection of Large-Scale Internet of Things Botnets Based on Graph Partitioning', *Applied Sciences*, vol. 14, no. 4, p. 1615, Feb. 2024, doi: 10.3390/app14041615.

[9] S. Bhattacharya *et al.*, 'A Novel PCA-Firefly Based XGBoost Classification Model for Intrusion Detection in Networks Using GPU', *Electronics*, vol. 9, no. 2, p. 219, Jan. 2020, doi: 10.3390/electronics9020219.

[10] X. Zhou, W. Liang, W. Li, K. Yan, S. Shimizu, and K. I.-K. Wang, 'Hierarchical Adversarial Attacks Against Graph-Neural-Network-Based IoT Network Intrusion Detection System', *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9310–9319, Jun. 2022, doi: 10.1109/JIOT.2021.3130434.

[11] M. A. Jabraeil Jamali, B. Bahrami, A. Heidari, P. Allahverdizadeh, and F. Norouzi, 'IoT Architecture', in *Towards the Internet of Things*, in EAI/Springer Innovations in Communication and Computing. , Cham: Springer International Publishing, 2020, pp. 9–31. doi: 10.1007/978-3-030-18468-1_2.

[12] A. Heidari and M. A. Jabraeil Jamali, 'Internet of Things intrusion detection systems: a comprehensive review and future directions', *Cluster Comput*, vol. 26, no. 6, pp. 3753–3780, Dec. 2023, doi: 10.1007/s10586-022-03776-z.

[13] Q. Li, L. Sun, B. Tang, H. Lu, J. Du, and X. Yu, 'Structure Enhancement Network Intrusion Detection Based on Graph Neural Network', in *Computer Supported Cooperative Work and Social Computing*, vol. 2344, H. Sun, H. Fan, Y. Gao, X. Wang, D. Liu, B. Du, and T. Lu, Eds., in Communications in Computer and Information Science, vol. 2344. , Singapore: Springer Nature Singapore, 2025, pp. 352–364. doi: 10.1007/978-981-96-2376-1_26.

[14] A. S. Ahanger, S. M. Khan, F. Masoodi, and A. O. Salau, 'Advanced intrusion detection in internet of things using graph attention networks', *Sci Rep*, vol. 15, no. 1, p. 9831, Mar. 2025, doi: 10.1038/s41598-025-94624-8.

[15] J. Sweeten, A. Takiddin, M. Ismail, S. S. Refaat, and R. Atat, 'Cyber-Physical GNN-Based Intrusion Detection in Smart Power Grids', in *2023 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, Glasgow, United Kingdom: IEEE, Oct. 2023, pp. 1–6. doi: 10.1109/SmartGridComm57358.2023.10333949.

[16] E. Caville, W. W. Lo, S. Layeghy, and M. Portmann, 'Anomal-E: A self-supervised network intrusion detection system based on graph neural networks', *Knowledge-Based Systems*, vol. 258, p. 110030, Dec. 2022, doi: 10.1016/j.knosys.2022.110030.

[17] T. Altaf, X. Wang, W. Ni, G. Yu, R. P. Liu, and R. Braun, 'A new concatenated Multigraph Neural Network for IoT intrusion detection', *Internet of Things*, vol. 22, p. 100818, Jul. 2023, doi: 10.1016/j.iot.2023.100818.

[18] G. Duan, H. Lv, H. Wang, and G. Feng, 'Application of a Dynamic Line Graph Neural Network for Intrusion Detection With Semisupervised Learning', *IEEE Trans.Inform.Forensic Secur.*, vol. 18, pp. 699–714, 2023, doi: 10.1109/TIFS.2022.3228493.

[19] M. Zivkovic, M. Tair, V. K, N. Bacanin, Š. Hubálovský, and P. Trojovský, 'Novel hybrid firefly algorithm: an application to enhance XGBoost tuning for intrusion detection classification', *PeerJ Computer Science*, vol. 8, p. e956, Apr. 2022, doi: 10.7717/peerj-cs.956.

[20] O. H. Abdulganiyu, T. A. Tchakoucht, Y. K. Saheed, and H. A. Ahmed, 'XIDINTFL-VAE: XGBoost-based intrusion detection of imbalance network traffic via class-wise focal loss variational autoencoder', *J Supercomput*, vol. 81, no. 1, p. 16, Jan. 2025, doi: 10.1007/s11227-024-06552-5.

[21] Y. Song, H. Li, P. Xu, and D. Liu, 'A Method of Intrusion Detection Based on WOA-XGBoost Algorithm', *Discrete Dynamics in Nature and Society*, vol. 2022, no. 1, p. 5245622, Jan. 2022, doi: 10.1155/2022/5245622.

[22] S. Amaouche, AzidineGuezzaz, S. Benkirane, and MouradeAzrour, 'IDS-XGbFS: a smart intrusion detection system using XGboostwith recent feature selection for VANET safety', *Cluster Comput*, vol. 27, no. 3, pp. 3521–3535, Jun. 2024, doi: 10.1007/s10586-023-04157-w.

[23] S. L(y)u, K. Wang, L. Zhang, and B. Wang, 'Global-local integration for GNN-based anomalous device state detection in industrial control systems', *Expert Systems with Applications*, vol. 209, p. 118345, Dec. 2022, doi: 10.1016/j.eswa.2022.118345.

[24] Q. Lin *et al.*, 'Robust Graph Neural Networks via Ensemble Learning', *Mathematics*, vol. 10, no. 8, p. 1300, Apr. 2022, doi: 10.3390/math10081300.

[25] T. Bilot, N. E. Madhoun, K. A. Agha, and A. Zouaoui, 'Graph Neural Networks for Intrusion Detection: A Survey', *IEEE Access*, vol. 11, pp. 49114–49139, 2023, doi: 10.1109/ACCESS.2023.3275789.

[26] T. Chen and C. Guestrin, 'XGBoost: A Scalable Tree Boosting System', in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco California USA: ACM, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[27] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, 'CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment', *Sensors*, vol. 23, no. 13, p. 5941, Jun. 2023, doi: 10.3390/s23135941.

[28] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, 'Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization':, in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, Funchal, Madeira, Portugal: SCITEPRESS - Science and Technology Publications, 2018, pp. 108–116. doi: 10.5220/0006639801080116.

[29] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, 'NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems', in *Big Data Technologies and Applications*, vol. 371, Z. Deze, H. Huang, R. Hou, S. Rho, and N. Chilamkurti, Eds., in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 371. , Cham: Springer International Publishing, 2021, pp. 117–135. doi: 10.1007/978-3-030-72802-1_9.

[30] N. Moustafa and J. Slay, 'The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set', *Information Security Journal: A Global Perspective*, vol. 25, no. 1–3, pp. 18–31, Apr. 2016, doi: 10.1080/19393555.2015.1125974.

[31] N. Moustafa, G. Creech, and J. Slay, 'Big Data Analytics for Intrusion Detection System: Statistical Decision-Making Using Finite Dirichlet Mixture Models', in *Data Analytics and Decision Support for Cybersecurity*, I. Palomares Carrascosa, H. K. Kalutarage, and Y. Huang, Eds., in Data Analytics. , Cham: Springer International Publishing, 2017, pp. 127–156. doi: 10.1007/978-3-319-59439-2_5.

[32] N. Moustafa, J. Slay, and G. Creech, 'Novel Geometric Area Analysis Technique for Anomaly Detection Using Trapezoidal Area Estimation on Large-Scale Networks', *IEEE Trans. Big Data*, vol. 5, no. 4, pp. 481–494, Dec. 2019, doi: 10.1109/TBDATA.2017.2715166.

[33] N. Moustafa and J. Slay, 'UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)', in *2015 Military Communications and Information Systems Conference (MilCIS)*, Canberra, Australia: IEEE, Nov. 2015, pp. 1–6. doi: 10.1109/MilCIS.2015.7348942.

[34] S. Dadkhah, E. Carlos Pinto Neto, R. Ferreira, R. Chukwuka Molokwu, S. Sadeghi, and A. Ghorbani, 'CICIoMT2024: Attack Vectors in Healthcare devices-A Multi-Protocol Dataset for Assessing IoMT Device Security', Feb. 16, 2024. doi: 10.20944/preprints202402.0898.v1.

[35] L. Liu, P. Wang, J. Lin, and L. Liu, 'Intrusion Detection of Imbalanced Network Traffic Based on Machine Learning and Deep Learning', *IEEE Access*, vol. 9, pp. 7550–7563, 2021, doi: 10.1109/ACCESS.2020.3048198.

[36] K. Polat, 'A novel data preprocessing method to estimate the air pollution (SO2): neighbor-based feature scaling (NBFS)', *Neural Comput & Applic*, vol. 21, no. 8, pp. 1987–1994, Nov. 2012, doi: 10.1007/s00521-011-0602-x.

[37] K. P. N. V. Satya Sree, J. Karthik, C. Niharika, P. V. V. S. Srinivas, N. Ravinder, and C. Prasad, 'Optimized Conversion of Categorical and Numerical Features in Machine Learning Models', in *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India: IEEE, Nov. 2021, pp. 294–299. doi: 10.1109/I-SMAC52330.2021.9640967.

[38] M. Mazziotta and A. Pareto, 'Everything you always wanted to know about normalization (but were afraid to ask)', *Rivista Italiana di Economia Demografia e Statistica*, pp. 41–52, 2021.

[39] T. Emmanuel, T. Maupong, D. Mpoeleng, T. Semong, B. Mphago, and O. Tabona, 'A survey on missing data in machine learning', *J Big Data*, vol. 8, no. 1, p. 140, Oct. 2021, doi: 10.1186/s40537-021-00516-9.

[40] J. T. Hancock and T. M. Khoshgoftaar, 'Survey on categorical data for neural networks', *J Big Data*, vol. 7, no. 1, p. 28, Dec. 2020, doi: 10.1186/s40537-020-00305-w.

[41] J. Huang, Y.-F. Li, and M. Xie, 'An empirical analysis of data preprocessing for machine learning-based software cost estimation', *Information and Software Technology*, vol. 67, pp. 108–127, Nov. 2015, doi: 10.1016/j.infsof.2015.07.004.

[42] S. S. Dhaliwal, A.-A. Nahid, and R. Abbas, 'Effective Intrusion Detection System Using XGBoost', *Information*, vol. 9, no. 7, p. 149, Jun. 2018, doi: 10.3390/info9070149.

[43] A. Churcher *et al.*, 'An Experimental Analysis of Attack Classification Using Machine Learning in IoT Networks', *Sensors*, vol. 21, no. 2, p. 446, Jan. 2021, doi: 10.3390/s21020446.

[44] Z. H. Hoo, J. Candlish, and D. Teare, 'What is an ROC curve?', *Emerg Med J*, vol. 34, no. 6, pp. 357–359, Jun. 2017, doi: 10.1136/emermed-2017-206735.

[45] C. Marzban, 'The ROC Curve and the Area under It as Performance Measures', *Weather and Forecasting*, vol. 19, no. 6, pp. 1106–1114, Dec. 2004, doi: 10.1175/825.1.

[46] Y. Qiu, J. Zhou, M. Khandelwal, H. Yang, P. Yang, and C. Li, 'Performance evaluation of hybrid WOA-XGBoost, GWO-XGBoost and BO-XGBoost models to predict blast-induced ground vibration', *Engineering with Computers*, vol. 38, no. S5, pp. 4145–4162, Dec. 2022, doi: 10.1007/s00366-021-01393-9.

[47] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, 'Network intrusion detection system: A systematic study of machine learning and deep learning approaches', *Trans Emerging Tel Tech*, vol. 32, no. 1, p. e4150, Jan. 2021, doi: 10.1002/ett.4150.

[48] N. Tekin, A. Acar, A. Aris, A. S. Uluagac, and V. C. Gungor, 'Energy consumption of on-device machine learning models for IoT intrusion detection', *Internet of Things*, vol. 21, p. 100670, Apr. 2023, doi: 10.1016/j.iot.2022.100670.

[49] X. Wang and Q. Wang, 'RETRACTED ARTICLE: An abnormal traffic detection method using GCN-BiLSTM-Attention in the internet of vehicles environment', *J Wireless Com Network*, vol. 2023, no. 1, p. 70, Jul. 2023, doi: 10.1186/s13638-023-02274-z.

[50] M. Bacevicius and A. Paulauskaite-Taraseviciene, 'Machine Learning Algorithms for Raw and Unbalanced Intrusion Detection Data in a Multi-Class Classification Problem', *Applied Sciences*, vol. 13, no. 12, p. 7328, Jun. 2023, doi: 10.3390/app13127328.

[51] J. Jose and D. V. Jose, 'Deep learning algorithms for intrusion detection systems in internet of things using CIC-IDS 2017 dataset', *IJECE*, vol. 13, no. 1, p. 1134, Feb. 2023, doi: 10.11591/ijece.v13i1.pp1134-1141.

[52] S. R and V. S, 'An Improving Intrusion Detection Model Based on Novel CNN Technique Using Recent CIC-IDS Datasets', in *2024 International Conference on Distributed Computing and Optimization Techniques (ICDCOT)*, Bengaluru, India: IEEE, Mar. 2024, pp. 1–6. doi: 10.1109/ICDCOT61034.2024.10515433.

[53] A. Thakkar and R. Lohiya, 'Attack Classification of Imbalanced Intrusion Data for IoT Network Using Ensemble-Learning-Based Deep Neural Network', *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11888–11895, Jul. 2023, doi: 10.1109/JIOT.2023.3244810.

[54] M. I. T. Hussan, G. V. Reddy, P. T. Anitha, A. Kanagaraj, and P. Naresh, 'DDoS attack detection in IoT environment using optimized Elman recurrent neural networks based on chaotic bacterial colony optimization', *Cluster Comput*, Nov. 2023, doi: 10.1007/s10586-023-04187-4.

[55] F. M. Aswad, A. M. S. Ahmed, N. A. M. Alhammadi, B. A. Khalaf, and S. A. Mostafa, 'Deep learning in distributed denial-of-service attacks detection method for Internet of Things networks', *Journal of Intelligent Systems*, vol. 32, no. 1, p. 20220155, Jan. 2023, doi: 10.1515/jisys-2022-0155.

[56] S. Wang, W. Xu, and Y. Liu, 'Res-TranBiLSTM: An intelligent approach for intrusion detection in the Internet of Things', *Computer Networks*, vol. 235, p. 109982, Nov. 2023, doi: 10.1016/j.comnet.2023.109982.

[57] B. Bowen, A. Chennamaneni, A. Goulart, and D. Lin, 'BLoCNet: a hybrid, dataset-independent intrusion detection system using deep learning', *Int. J. Inf. Secur.*, vol. 22, no. 4, pp. 893–917, Aug. 2023, doi: 10.1007/s10207-023-00663-5.

[58] J. Cui, L. Zong, J. Xie, and M. Tang, 'A novel multi-module integrated intrusion detection system for high-dimensional imbalanced data', *Appl Intell*, vol. 53, no. 1, pp. 272–288, Jan. 2023, doi: 10.1007/s10489-022-03361-2.

[59] A. Meliboev, J. Alikhanov, and W. Kim, 'Performance Evaluation of Deep Learning Based Network Intrusion Detection System across Multiple Balanced and Imbalanced Datasets', *Electronics*, vol. 11, no. 4, p. 515, Feb. 2022, doi: 10.3390/electronics11040515.

[60] Y. F. Sallam *et al.*, 'Efficient implementation of image representation, VISUAL GEOMETRY GROUP WITH 19 LAYERS and RESIDUAL NETWORK WITH 152 LAYERS for intrusion detection from UNSW-NB15 dataset', *Security and Privacy*, vol. 6, no. 5, p. e300, Sep. 2023, doi: 10.1002/spy2.300.

[61] M. H. Kabir, M. S. Rajib, A. S. M. T. Rahman, Md. M. Rahman, and S. K. Dey, 'Network Intrusion Detection Using UNSW-NB15 Dataset: Stacking Machine Learning Based Approach', in *2022 International Conference on Advancement in Electrical and Electronic Engineering (ICAEEE)*, Gazipur, Bangladesh: IEEE, Feb. 2022, pp. 1–6. doi: 10.1109/ICAEEE54957.2022.9836404.

[62] A. Shehadeh, H. ALTaweel, and A. Qusef, 'Analysis of Data Mining Techniques on KDD-Cup'99, NSL-KDD and UNSW-NB15 Datasets for Intrusion Detection', in *2023 24th International Arab Conference on Information Technology (ACIT)*, Ajman, United Arab Emirates: IEEE, Dec. 2023, pp. 1–6. doi: 10.1109/ACIT58888.2023.10453884.

[63] I. Tareq, B. M. Elbagoury, S. El-Regaily, and E.-S. M. El-Horbaty, 'Analysis of ToN-IoT, UNW-NB15, and Edge-IIoT Datasets Using DL in Cybersecurity for IoT', *Applied Sciences*, vol. 12, no. 19, p. 9572, Sep. 2022, doi: 10.3390/app12199572.

[64] A. I. Jony and A. K. B. Arnob, 'A long short-term memory based approach for detecting cyber attacks in IoT using CIC-IoT2023 dataset', *J. Edge Comp.*, vol. 3, no. 1, pp. 28–42, May 2024, doi: 10.55056/jec.648.

[65] F. L. Becerra-Suarez, V. A. Tuesta-Monteza, H. I. Mejia-Cabrera, and J. Arcila-Diaz, 'Performance Evaluation of Deep Learning Models for Classifying Cybersecurity Attacks in IoT Networks', *Informatics*, vol. 11, no. 2, p. 32, May 2024, doi: 10.3390/informatics11020032.

[66] N. V. Patil, C. R. Krishna, and K. Kumar, 'SSK-DDoS: distributed stream processing framework based classification system for DDoS attacks', *Cluster Comput*, vol. 25, no. 2, pp. 1355–1372, Apr. 2022, doi: 10.1007/s10586-022-03538-x.

[67] T.-T.-H. Le, R. W. Wardhani, D. S. C. Putranto, U. Jo, and H. Kim, 'Toward Enhanced Attack Detection and Explanation in Intrusion Detection System-Based IoT Environment Data', *IEEE Access*, vol. 11, pp. 131661–131676, 2023, doi: 10.1109/ACCESS.2023.3336678.

[68] K. Roshan and A. Zafar, 'Ensemble adaptive online machine learning in data stream: a case study in cyber intrusion detection system', *Int. j. inf. tecnol.*, Feb. 2024, doi: 10.1007/s41870-024-01727-y.

[69] Y. K. Beshah, S. L. Abebe, and H. M. Melaku, 'Drift Adaptive Online DDoS Attack Detection Framework for IoT System', *Electronics*, vol. 13, no. 6, p. 1004, Mar. 2024, doi: 10.3390/electronics13061004.

[70] A. Salehpour, M. A. Balafar, and A. Souri, 'An optimized intrusion detection system for resource-constrained IoMT environments: enhancing security through efficient feature selection and classification', *J Supercomput*, vol. 81, no. 6, p. 783, Apr. 2025, doi: 10.1007/s11227-025-07253-3.

[71] N. Sharma and P. G. Shambharkar, 'Multi-attention DeepCRNN: an efficient and explainable intrusion detection framework for Internet of Medical Things environments', *Knowl Inf Syst*, Apr. 2025, doi: 10.1007/s10115-025-02402-9.

[72] G. Akar, S. Sahmoud, M. Onat, Ü. Cavusoglu, and E. Malondo, 'L2D2: A Novel LSTM Model for Multi-Class Intrusion Detection Systems in the Era of IoMT', *IEEE Access*, vol. 13, pp. 7002–7013, 2025, doi: 10.1109/ACCESS.2025.3526883.

[73] F. G. Abdiwi, 'Hybrid Machine Learning and Blockchain Technology for Early Detection of Cyberattacks in Healthcare Systems', *IJSSE*, vol. 14, no. 6, pp. 1883–1893, Dec. 2024, doi: 10.18280/ijsse.140622.

[74] A. Misbah, A. Sebbar, and I. Hafidi, 'Securing Internet of Medical Things: An Advanced Federated Learning Approach', *ijacsa*, vol. 16, no. 2, 2025, doi: 10.14569/IJACSA.2025.01602129.

[75] D. Torre, A. Chennamaneni, J. Jo, G. Vyas, and B. Sabrsula, 'Toward Enhancing Privacy Preservation of a Federated Learning CNN Intrusion Detection System in IoT: Method and Empirical Study', *ACM Trans. Softw. Eng. Methodol.*, vol. 34, no. 2, pp. 1–48, Feb. 2025, doi: 10.1145/3695998.

[76] W. W. Lo, S. Layeghy, M. Sarhan, M. Gallagher, and M. Portmann, 'E-GraphSAGE: A Graph Neural Network based Intrusion Detection System for IoT', in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, Budapest, Hungary: IEEE, Apr. 2022, pp. 1–9. doi: 10.1109/NOMS54207.2022.9789878.

Article Information Form

Author Contributions: Onur Ceran contributed to conceptualization and writing the original draft. Erdal Özdoğan contributed to the methodology and formal analysis. Mevlüt Uysal contributed to the literature review and provided guidance particularly results section. Each author played a vital role in developing this work, ensuring its quality and accuracy.

Artificial Intelligence Statement: The authors declare that they have not used any generative AI or AI-assisted technologies in this paper.

Plagiarism Statement: This article has been scanned by iThenticate.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
http://saucis.sakarya.edu.tr/
e-ISSN: 2636-8129
Publisher: Sakarya University
Vol. 8, No. 2, 245-259, 2025
DOI: 10.35377/saucis...1646658
SAUCIS
SAKARYA UNIVERSITY
RESEARCH ARTICLE

DeepInsulin-Net: A Deep Learning Model for Identifying Drug Interactions Leading to Specific Insulin-Related Adverse Events

Muhammed Ali PALA[1,2]

[1]Department of Electrical and Electronics Engineering, Faculty of Technology, Sakarya University of Applied Sciences, Sakarya, Türkiye; ORCID: 0000-0002-8153-7971
[2]Biomedical Technologies Application and Research Center (BIYOTAM), Sakarya University of Applied Sciences, Sakarya, Türkiye.

Corresponding author:

Muhammed Ali Pala,
Department of Electrical and Electronics Engineering,
Faculty of Technology,
Sakarya University of Applied Sciences,
Sakarya, Türkiye
pala@subu.edu.tr

Article History:
Received: 25.02.2025
Revised: 21.05.2025
Accepted: 11.06.2025
Published Online: 16.06.2025

ABSTRACT
Predicting clinical adverse effects resulting from drug-drug interactions is a critical research area for drug safety and patient health. Specifically, predicting adverse effects associated with insulin is crucial for clinical decision support systems and pharmacovigilance applications. This study proposes a deep learning-based model with high accuracy to predict adverse effects caused by drug interactions. In the literature, 17 different clinical side effects commonly associated with the hormone insulin have been identified. The properties of the drug molecules causing these interactions were calculated through MACCS, Morgan fingerprints and RDKit descriptors. These features are filtered by the variance thresholding method and optimized to improve classification performance. The model is built on a 1D CNN architecture that handles drug pairs as parallel inputs and a class weighting technique is used to eliminate class imbalance. Experimental results show that the model achieves 99.66% accuracy in training and 94.03% in validation, with training loss decreasing to 0.01 and validation loss stabilizing at 0.22. The ROC-AUC metric is above 0.99, indicating that the model can predict infrequent adverse events. The developed model provides a scalable, computationally efficient and highly reliable approach to predict the clinical consequences of drug interactions.

Keywords: Artificial intelligence, Personalized medicine, Drug-Drug interactions, Adverse drug events, Insulin, Deep learning

1. Introduction

In modern medical approaches, the use of drugs in treating and managing diseases is gradually increasing. In particular, chronic diseases are becoming widespread worldwide due to population aging and many other environmental and genetic influences [1]. This situation brings with it various challenges in the provision of health services and the treatment of diseases. Many alternative treatment methods and medicines are being developed to overcome these challenges [2]. In particular, polypharmacy, i.e., the regular use of more than one medication simultaneously, has become an essential step in today's treatment processes [3], [4]. Polypharmacy has the potential to treat many diseases in individuals simultaneously. However, while it has the potential to increase treatment success and manage comorbidities, it has also brought with it a significant increase in the side effects of drug-drug interactions that occur as a result of this multiple drug use [5], [6]. Drug-drug interactions occur when one drug alters another drug's pharmacokinetic or pharmacodynamic properties. These interactions can lead to a decrease in treatment efficacy, making it difficult to control the disease, as well as significantly increasing the risk of adverse drug reactions (ADRs) [7]. For example, one drug may decrease the efficacy of another by accelerating its metabolism, while another drug may increase its toxicity by slowing its metabolism. In clinical practice, this can manifest as unexpected treatment failures, increased side effects, the emergence of new symptoms or exacerbation of existing diseases [8]. The clinical burden of DDIs not only adversely affects treatment outcomes but also poses a serious threat to patient safety and a significant economic burden on healthcare systems [9]. Therefore, accurately predicting the clinically relevant consequences of drug interactions, mitigating risks, and developing effective management strategies have become critical global priorities to improve patient care quality, ensure patient safety, and reduce healthcare expenditures. In this context, there is a growing need for innovative approaches to detect drug interactions early and accurately, support clinical decision-

Cite as: M. A. Pala. (2025). DeepInsulin-Net: A deep learning model for identifying drug interactions leading to specific insulin-related adverse events. *Sakarya University Journal of Computer and Information Sciences*. vol. 8, no. 2, pp. 245-259, 2025 doi: 10.35377/saucis...1646658

making and improve patient outcomes.

Insulin is a vital hormone with pleiotropic effects in the human body. Produced by the beta cells of the pancreas, insulin primarily plays a central role in regulating glucose metabolism. It controls blood glucose levels by ensuring glucose is taken into cells and used for energy production [10], [11]. However, the effects of insulin are not limited to glucose metabolism. It also plays essential roles in many physiological processes, such as protein and fat metabolism, cell growth and differentiation, vascular function and even neuronal activity [12]–[14]. This wide range of physiological effects makes insulin important in maintaining the body's homeostatic balance [15]. This vital role of insulin and its wide range of actions are directly related to various clinical conditions resulting from its deficiency, resistance or interactions with other drugs [16]. These conditions are not limited to a generalized picture of insulin deficiency or insulin resistance but can lead to a wide range of specific and clinically identifiable diseases and disorders [17], [18]. These diseases and conditions range from hypoglycemia to hyperglycemia, from diabetic ketoacidosis to chronic complications such as diabetic nephropathy, diabetic neuropathy and diabetic retinopathy, from gestational diabetes to neonatal hypoglycemia [19], [20]. Even less well-known but essential clinical conditions, such as glucose intolerance, abnormal liver function tests (LFTs), polydipsia, polyuria and peripheral vascular disease, can result from insulin dysfunction or drug interactions [21]. The diversity and complexity of these clinical outcomes make it necessary to consider these conditions and diseases as insulin-related and carefully examine drug interactions in this area.

Each of these insulin-related diseases and conditions implies different levels of risk and treatment management challenges for patients. For example, hypoglycemia is an acute condition that can be serious, even fatal, especially in elderly and frail patients [22]. Chronic complications, such as diabetic nephropathy and retinopathy, are conditions that significantly reduce the quality of life in the long term and cause severe morbidity and mortality [23]. Conditions such as gestational diabetes and neonatal hypoglycemia are critical for both maternal and infant health and require specialized treatment approaches [24]. Therefore, accurately predicting and classifying the potential clinical consequences of drug interactions in the context of these insulin-related diseases and conditions is vital to personalize treatment strategies, reduce patient risk and support clinical decision-making [9]. The clinical diversity, importance, and potential impact of drug interactions on these conditions are the primary motivations for our study, which focuses on insulin-related drug interactions and their clinical consequences.

Timely and accurate detection of drug-drug interactions is critical to ensure patient safety and optimize treatment success [25], [26]. Traditional DDI prediction methods rely heavily on pharmacological studies, clinical observations, adverse event reports and expert knowledge. While these methods are valuable in identifying interactions between specific drug pairs, they are inadequate in the face of today's rapidly expanding drug knowledge and the increasing reality of polypharmacy [27]. Traditional methods often struggle to uncover complex interaction patterns that require large-scale data analysis and are limited in generating reliable predictions for new drug combinations. Furthermore, these methods usually assess the overall interaction risk and do not provide detailed information on the specific clinical consequences of the interaction [28]. This makes it difficult for clinicians to personalize treatment decisions and minimize patient risk.

Artificial intelligence, especially deep learning methods, has revolutionized drug discovery and development processes in the biomedical field in recent years [29]–[31]. Through multilayer artificial neural networks, deep learning can automatically extract meaningful features and patterns from large and complex data sets. This ability has made deep learning a powerful tool for solving complex problems like drug-drug interaction prediction. Deep learning models have shown promising results in predicting DDIs by integrating information from various data sources such as chemical structures of drugs, genetic data, biological pathway information, electronic health records and scientific literature [25], [32]. In the literature, deep learning-based DDI prediction models have achieved higher accuracy, sensitivity and specificity than traditional methods. These models can detect known interactions and predict novel and potential interactions. Deep learning also offers significant advantages in modeling multidrug interactions, predicting the effects of drug combinations and elucidating the mechanisms underlying interactions [33], [34]. However, most existing deep learning-based DDI prediction studies often focus on generic drug interactions and do not provide in-depth analyses of specific therapeutic areas or clinical outcomes. This limits the potential to provide more meaningful and actionable insights in clinical practice. This shortcoming reveals a significant research gap for better understanding and management of drug interactions in patients with diabetes and insulin therapy.

Among deep learning architectures, Convolutional Neural Networks (CNNs) are emerging as a notable alternative in predicting drug-drug interactions [35]. Although CNNs were first recognized for their superior performance in image processing, especially one-dimensional Convolutional Neural Networks (1D CNNs), they can be applied successfully to one-dimensional data encoding the molecular properties of drugs. 1D CNNs can automatically learn local patterns and hierarchical relationships in the data by performing a convolution operation through a convolutional filter acting on the input data. This capability significantly reduces the need for manual feature engineering and allows the model to delve deeper into complex interaction mechanisms that have not been previously identified. This offers a critical advantage for modeling the subtle and complex biochemical processes underlying drug interactions.

2. Proposed Method

This study presents a novel and advanced deep-learning approach that aims to classify potential insulin-related clinical side effects caused by drug-drug interactions with high accuracy and specificity. The flowchart of the proposed method is given in Figure 1. The technique, DeepInsulin-Net, examines 17 clinical side effects most commonly associated with insulin-related adverse events found in the literature. This multi-class classification approach allows the model to learn more complex interaction patterns and make more accurate predictions. The representation of drug molecules through SMILES (Simplified Molecular Input Line Entry Specification) using a feature set enriched with various fingerprints and identifiers allows the 1D CNN model to consider different chemical and structural properties. The parallel 1D CNN architecture of the model enables the features of both drugs to be processed independently and then combined to predict the probability of interaction. This approach reduces the computational cost and increases the scalability of the model. By combining a customized 1D CNN architecture for the multiclass classification problem, an enriched molecular feature set, and a focus on insulin-related drugs, the study presents a novel and technically powerful method in the field of DDI prediction.

The proposed method provides a multi-stage, integrated pipeline that combines the power of a 1D CNN architecture with a rich feature set representing drug molecular structures and properties. In the first stage of this pipeline, data on insulin and related drug pairs and their interactions were obtained from TWOSIDES, a comprehensive drug interaction database. This dataset was subjected to rigorous pre-processing, including identifying and cleaning conflicting labels. The interaction labels were converted into a numerical format according to the requirements of the model, and the dataset was divided into training and validation sets to evaluate the model's generalization ability. An extensive feature engineering process was applied to capture the complex properties of drugs at the molecular level and improve the model's classification performance. In this process, molecular fingerprints representing the chemical structures of drugs and various molecular descriptors were computed using the RDKit library. These features provide rich and multidimensional information about the chemical structures, physicochemical properties and potential interaction mechanisms of drugs. These features obtained from different sources were combined into a single feature vector by concatenation and scaled in the range [0, 1] to make them suitable for the model's input. To reduce the model's complexity and select the most informative features, feature selection was performed using the variance threshold method.

The model's architecture consists of 1D convolutional layers parallel to each drug pair, which learn the latent and hidden representations of the drugs by taking molecular feature vectors as input. These representations for the two drugs are combined through a concatenation layer and transferred to a standard dense layer. This layer is connected to a softmax output layer that estimates the interaction probability based on 17 different classes of clinical outcomes. The performance of the developed model is extensively evaluated on an independent validation dataset. The overall performance of the model and its ability to predict different clinical outcomes were analyzed using various performance metrics such as accuracy, AUC, precision, recall, and confusion matrix. The clinical outcomes the model predicts with higher accuracy, in which classes it struggles, and the possible sources of errors are analyzed in detail. As one of the first studies to focus on classifying the clinical outcomes of insulin-related drug interactions, this study makes a unique and valuable contribution to the DDI prediction literature. Integrating the developed model into clinical decision support systems has the potential to optimize medication management, reduce the risk of adverse events and improve patient safety in patients with diabetes and insulin therapy.

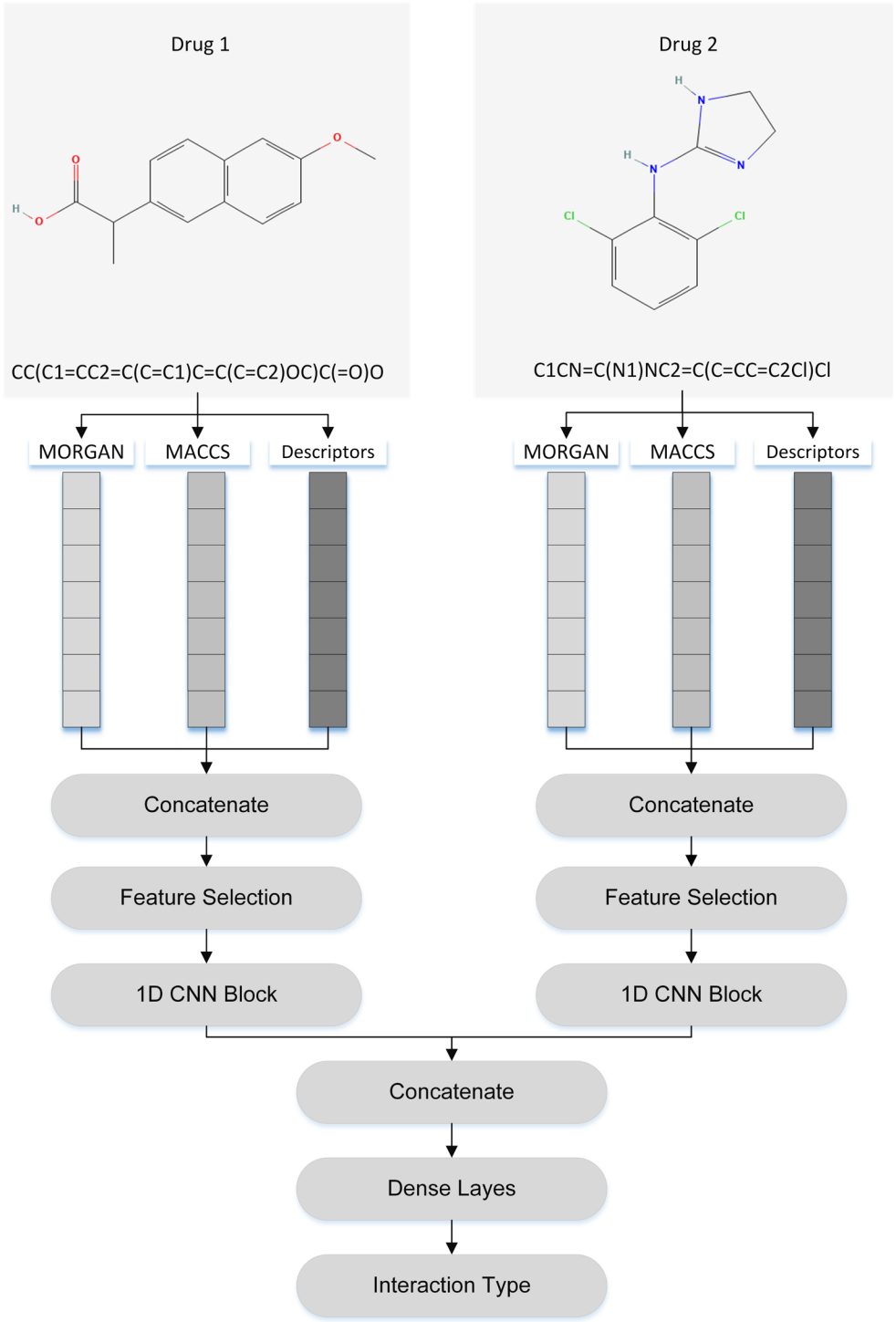


Figure 1. Flowchart of the DeepInsulin-Net.

3. Method

3.1 Data set and pre-processing

This study used the TWOSIDES dataset, a comprehensive and reliable drug interaction database derived from clinical data, to classify the clinical outcomes of insulin-related drug-drug interactions [36]. The TWOSIDES database contains 868,221 significant associations and 1,301 adverse events for 59,220 different drug pairs. These adverse events include conditions that cannot be attributed to any single drug. Furthermore, the database contains 3,782,910 significant associations of drug pairs, and the adverse effect scores of these interactions were higher than those of the individual drugs involved. In our study, we mainly focused on developing models that predict adverse drug reactions resulting from synergistic drug-drug interactions. The fact that TWOSIDES covers a wide range of drugs and drug interactions ensures that interactions of insulin

and related drugs are adequately represented in the database, providing a rich and reliable data source suitable for the study. The TWOSIDE dataset includes the SMILES representation of each drug pair to identify drug interactions and the type of drug interaction as label information.

A series of systematic steps were followed to select insulin-related side effects from the TWOSIDES database. First, insulin-related side effects and interactions were carefully identified through a literature search. Then, from the TWOSIDES database, all drug pairs containing these identified drugs and adverse events resulting from their interactions were eliminated from the entire dataset using text-mining steps. These adverse events were classified into 17 different clinical outcome classes, which were the focus of the study. These classes comprehensively included diabetic complications (diabetic nephropathy, diabetic retinopathy, diabetic neuropathy), disorders of glycemic control (hypoglycemia, hyperglycemia), gestational diabetes, abnormal liver function tests and other related conditions. This raw data was transformed into a structured dataset, with each row indicating a drug pair and the clinical outcome resulting from their interaction. The DDI interactions analyzed in the study are given in Table 1. In total, 93626 interaction pairs were used for the proposed method. The dataset was randomly split into training and evaluation in a 7:3 ratio.

Table 1. Number of insulin-related data in the DDI dataset.

Type	**Number of Interaction**
Hyperglycaemia	18779
Diabetes	12718
Hypoglycaemia	12443
Abnormal LFTs	10960
Polydipsia	5934
Diabetic neuropathy	5168
Peripheral vascular disease	5165
Diabetic acidosis	3523
Polyuria	3366
Insulin-dependent diabetes mellitus	3335
Diabetic Retinopathy	2982
Glucosuria	2894
Glucose intolerance	2107
Ketoacidosis	1362
Diabetic Nephropathy	1264
Hypoglycaemia neonatal	1206
Gestational diabetes	420

Furthermore, to ensure that the performance and reliability of the model are maximized, a comprehensive pre-processing process was applied to the resulting dataset. This process primarily involved identifying and cleaning records that reported multiple and different clinical outcomes for the same drug pair, i.e., conflicting labels. Removing these conflicting records from the dataset improved the consistency and learning ability of the model. The main reason for eliminating conflicting data is that the same drug pair can lead to multiple and different clinical outcomes, complicating the model's learning process and leading to misleading results. Therefore, SMILES strings, in this case, were removed from the dataset. Figure 2 shows the data distribution plot for 17 different insulin-related drug interactions. As can be seen, the dataset has an unbalanced class distribution. The proposed method uses a class weighting strategy to address the unbalanced class distribution. In this method, weights are assigned to each class based on the class distribution in the training data. In this way, it aims to take more imbalanced classes into account by the model and improve its learning process on imbalanced datasets. The obtained weights are integrated into the model's loss function to minimize the performance problems caused by class imbalance.

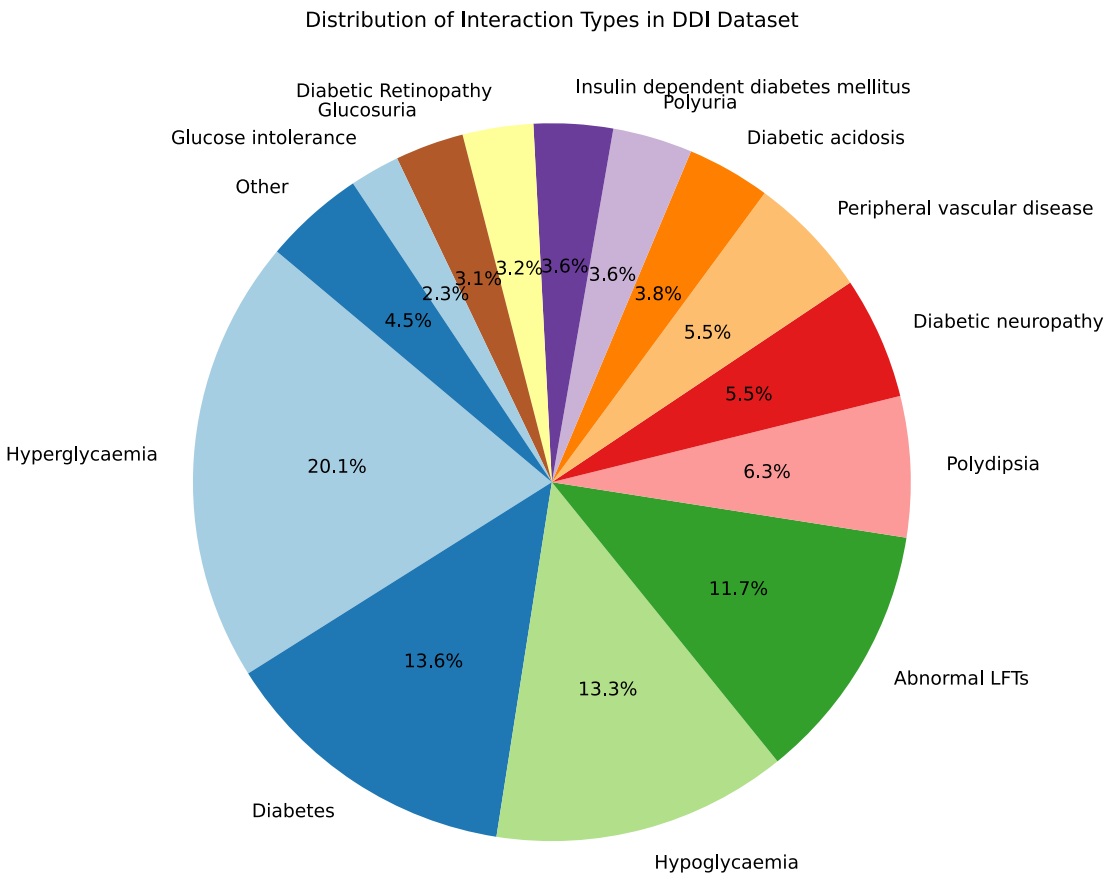


Figure 2. Data distribution

3.2 Fingerprints and Descriptors

Accurate, reliable and clinically meaningful prediction of drug-drug interactions is, without a doubt, critically dependent on representing the molecular structures of drugs and the properties derived from these structures in a way that is appropriate, informative and serves the learning capacity of the model. The transformation of drugs in molecular form into numerical vectors and abstract mathematical objects allows machine learning algorithms to learn the subtle but deeply complex relationships between drugs, make generalizations from these relationships, and ultimately predict interactions for unknown drug pairs. This study used different representation methods, namely two different molecular fingerprints and molecular descriptors, to represent the molecular properties of drugs in a way that encompasses both structural and physicochemical dimensions, i.e., by providing multidimensional and complementary information, thus maximizing the model's classification performance. These approaches offer vital information about the chemical structures, three-dimensional conformations, physicochemical properties and potential interaction mechanisms of the drugs, which are crucial for the model to classify DDIs successfully. MACCS and Morgan fingerprints of drug molecules, as well as RdKit descriptors, were used in this study. SMILES representations of the molecules were used to calculate the fingerprints and identifiers.

MACCS fingerprints are a well-established type of fingerprinting developed by MDL Information Systems, including 166 predefined structural keys [37]. These structural keys are carefully selected structural motifs representing functional groups, ring systems, and specific atomic numbers commonly found in drug molecules. The MACCS fingerprint of a molecule consists of a 166-bit array that indicates whether these 166 keys are present in the molecule. MACCS fingerprints have the advantages of covering an ample chemical space, being relatively interpretable and straightforward, and being computationally efficient. Morgan fingerprints, also known as circular fingerprints, are a type of fingerprint that takes into account the atom-centered peripheral properties of the molecule [38]. Starting from each atom, Morgan fingerprints form circular substructures that contain all neighboring atoms and bonds within a given radius. These substructures are converted into unique integer values using a hash function, and these integers are used to set the corresponding bits in the fingerprint vector to 1. The most crucial feature of Morgan fingerprints is their ability to capture in detail the local structural features of the molecule and the neighborhood and bonding patterns of atoms and, thus, more precisely, identify subtle structural similarities and differences between different molecules. In this study, a rich and detailed representation of the molecular structure of drugs was obtained using Morgan fingerprints with a length of 2048 bits and a radius of 2. Molecular descriptors

are another critical and complementary category of properties that express drug molecules' physicochemical, topological, electronic and geometrical properties in numerical values. Descriptors quantitatively characterize a wide range of properties of the molecule, such as size, shape, surface area, polarity, hydrophobicity, hydrogen bonding capacity, dipole moment, ionization potential and electronic charge distribution, which directly affect the pharmacokinetic and pharmacodynamic properties of the drug. In this study, using the open-source and widely used RDKit chemoinformatics library, four basic molecular descriptors, namely molecular weight (MolWt), LogP (octanol-water partition coefficient), number of hydrogen bond donors (NumHDonors) and number of hydrogen bond acceptors (NumHAcceptors), were calculated, which are frequently used in the literature and play an essential role in predicting the biological behavior of drugs [39].

These molecular fingerprints and descriptors calculated for each drug molecule were combined into a single, long feature vector by concatenation. The concatenated feature matrices resulting from this process, which are performed separately for each drug interaction pair, are applied as input to the CNN model. This combined feature vector represents the structural and physicochemical properties of drugs together and comprehensively, providing a rich and informative input that allows the model to classify drug-drug interactions more accurately, precisely and reliably.

3.3 Feature Selection

Feature selection is essential in data analysis and modeling processes because it can affect the model's learning capacity as the data's size and complexity increase. In deep learning algorithms, too many features can increase the training time of the model, cause overfitting and limit the model's generalization ability [40]. For this reason, only essential and meaningful features should be included in the model. Especially in high-dimensional datasets, some features may lead to performance loss instead of improving the model's accuracy. Therefore, feature selection is critical to ensure the model runs efficiently and accurately. The variance threshold feature selection method is applied to the combined feature matrix created with fingerprints and descriptors [41]. This way, the feature matrices of high-dimensional drug molecules are expressed in a more meaningful matrix format. Therefore, this reduced the computational burden of the model and improved the generalization performance of the model.

The Variance Threshold method calculates the variance of each attribute and removes features with variance below a specific threshold value. Variance is a statistical measure of how variable the values of an attribute are and is calculated as follows:

$$\text{Variance}(X_j) = \frac{1}{n-1}\sum_{i=1}^{n}(X_{ij} - \mu_j)^2 \qquad (1)$$

Where X_j represents the values of the feature across all samples. μ_j is the mean of the $j-th$ feature and n is the number of the samples. If a feature's variance is lower than a specified threshold (θ), that feature is removed from the dataset. The variance thresholding method usually aims to extract features with low variance in the dataset, i.e., features with similar values in all instances. Such features do not benefit the model or contribute to the learning process.

3.4 CNN Model

This study developed a novel and advanced deep learning model based on a 1D Convolutional Neural Network architecture to classify the clinical outcomes of insulin-related drug-drug interactions with high accuracy and specificity. CNNs first proved themselves in image processing, especially in object recognition, image classification and face recognition. However, the basic principles of CNNs, particularly the advantages of convolution, have made them successfully applicable to various fields such as natural language processing, text classification, sentiment analysis, time series prediction, anomaly detection, gene sequence analysis and protein structure prediction.

1D CNNs are a subtype of CNNs specialized for application to one-dimensional data. 1D data is data with a sequential structure, such as texts, audio signals, time series, gene sequences and, as in this study, molecular fingerprints and identifiers. 1D CNNs perform a convolution on such data using a sliding filter. This filter is a small matrix of weights that moves step by step along with the input data with a given stride size. At each step, an element-wise multiplication is performed between the input region covered by the filter and the filter weights, and the sum of the resulting products forms the corresponding element of a feature map [42]. This process enables the automatic detection of local patterns in the input data. The architecture of 1D CNNs typically consists of an input layer, one or more convolutional layers, activation functions, pooling layers, fully connected layers and an output layer. The input layer receives the raw data that the model will process. In this study, the input layer receives the combined feature vectors obtained during the feature extraction stage of drug feature engineering, which is dimensionally reduced by feature selection. Convolutional layers are the basic building blocks of 1D CNNs and produce feature maps by performing a convolution on the input data. Each convolutional layer uses different filters, each learning to capture different local patterns in the input data. Nonlinearity is introduced into the model by applying activation functions to the outputs of the convolutional layers. Activation functions enable the model to learn nonlinear relationships and solve more complex problems. Dense layers are usually located at the end of the CNN architecture and connect all neurons from the previous layers. These layers allow the local features learned by the convolutional layers to be combined to form higher-level and abstract representations. The output layer is where the model produces its final predictions. In this study, the output

layer computes the probabilities of 17 different clinical outcomes for each drug pair using the softmax activation function. The architecture of the 1D CNN model developed for classifying insulin-related DDI pairs is given in Figure 3.

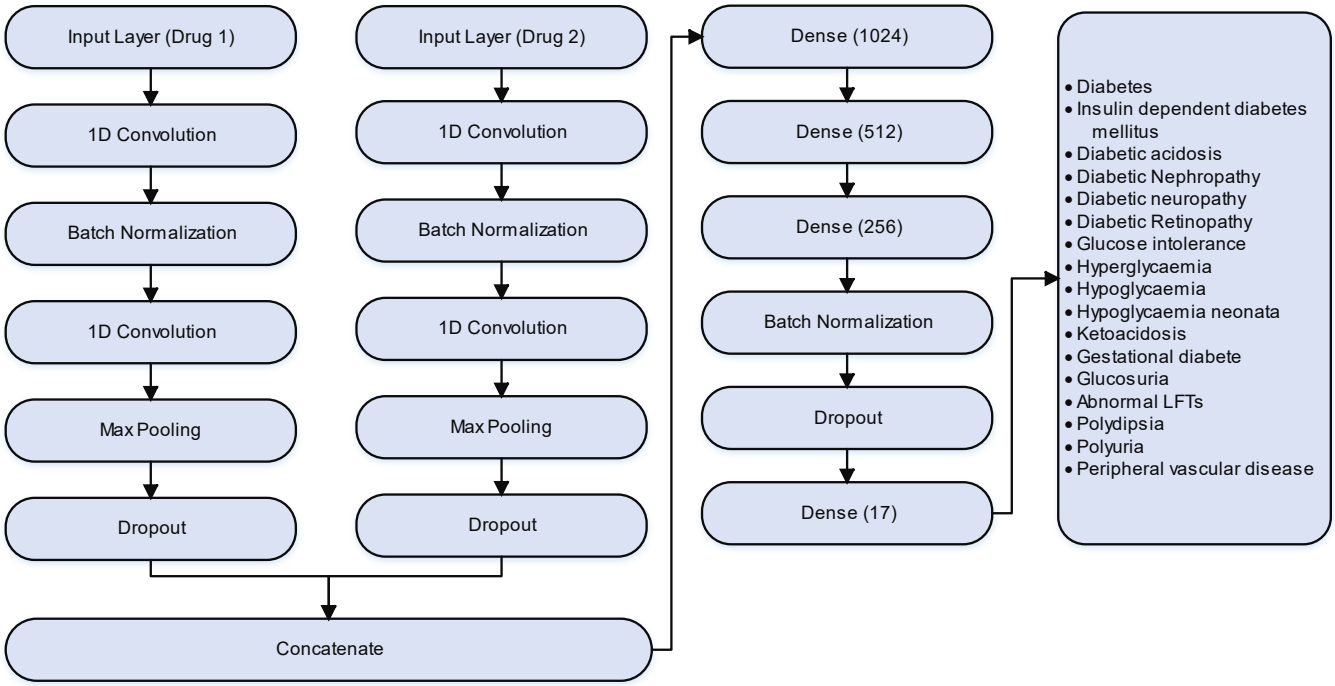


Figure 3. DeepInsulin-Net CNN model architecture.

The model proposed in this paper has a dual-input structure. It uses a parallel processing architecture to process two different drug feature matrices (Drug1 and Drug2) separately and then combine them to obtain a feature representation. In the model's design, each drug's data features are processed independently and combined to get the final classification result. Such a parallel structure allows for more efficient drug interaction modeling while ensuring that each drug's data features are considered separately. This approach offers significant advantages, especially in solving complex biological problems like drug interactions. The model independently processes feature vectors from two drugs to create high-level representations and combines these representations to perform the final classification. The parallel input structure processes the molecular features of both drugs through separate convolutional blocks, aiming to obtain independent but meaningful representations of each drug in a common feature space. This architecture offers higher generalization capacity when analyzing the potential for different drugs to interact with each other.

In the proposed model, there is an independent 1D CNN block for each drug. First, each input is processed with a convolutional layer. In the first convolutional block, feature extraction is performed with a 3D kernel using 128 filters. This is followed by a batch normalization process to provide more balanced learning in the model's training process. Then, the convolution layer is used again. After each convolution block, a dropout layer of 0.1 is added to avoid overfitting the model. After the first convolution block, a second convolutional layer extracts higher-level features with 64 filters. Similarly, this layer undergoes batch normalization, pooling, and dropout to complete the separate processing of each input. In the final stage, after the convolutional processing of each input is completed, it is converted into vector form through the smoothing layer and made ready for processing with dense layers.

In the fusion stage of the model, the feature vectors obtained from both inputs are combined using the concatenate layer. The concatenated vector was processed using the ReLU activation function in dense layers with 1024, 512 and 256 neurons, respectively. Batch normalization was applied to the output of this layer. In the output layer of the model, a dense layer with a softmax activation function was used to consider the multiclass classification problem. The model was compiled with the Adam optimization algorithm (learning rate: 0.0005), and categorical cross-entropy was used as the loss function. During the training process, an early stopping mechanism was added to monitor the model's performance and to activate the early stopping mechanism, which terminates training if the validation loss does not improve for 100 epochs. In the training process, a class weighting method was used to make the model learn in a more balanced way. If the class distribution is unbalanced, class weights are calculated from the training data and integrated into the model's loss function to increase the model's sensitivity to rare classes. The training process was completed after 50 epochs as it met the desired criteria. The hyperparameters of the designed model are given in Table 2.

Table 2. Hyperparameters of the proposed model

Hyperparameters	**Value**
Number of Filters	128, 64
Kernel Size	3
Pooling Size	2
Dropout Rate	0.1
Optimization Algorithm	Adam
Learning Rate	0.0005
Batch Size	64
Epoch	50
Activation Function	ReLU, Softmax
Loss Function	Categorical Crossentropy

The performance evaluation metrics of the developed model are crucial for assessing its overall accuracy and the effectiveness of the learning process. This is particularly important when working with datasets that exhibit an unbalanced distribution, as it necessitates using multiple metrics during the training and evaluation phases. Consequently, various performance metrics were employed to gauge the comprehensive success of the model. In this context, we calculated the ROC-AUC (Receiver Operating Characteristic - Area Under Curve), recall, accuracy, and log loss.

4. Experimental Results

The model's performance is analyzed with various metrics, and the effects of different data representation techniques on the performance are examined. In addition, statistical analysis of the prediction results is performed to understand the decision mechanism of the model better. The multi-class dataset labels prepared in the study were converted into binary strings using the one-hot encoding method. The labels were transformed with this method and used as the target variable of the model. A structure was created to compare the predicted probabilities with the actual labels. This approach also helped reduce potential problems, such as unbalanced class distribution in the dataset and ensured stable learning during the model's training process. The experiments used a NVIDIA GeForce GTX 3070 GPU with Intel Core i7-11700H CPU @ 4.90 GHz and 32 GB RAM.

ROC-AUC is a critical metric that measures how well the model can distinguish between positive and negative classes at different thresholds. An AUC value close to 1 indicates that the model is highly discriminative, while an AUC value close to 0.5 indicates that the model performs at the level of random guessing. In the evaluation step, attention was paid to including only samples with at least two different classes in the analysis. The AUC metric is a vital indicator for understanding the overall generalization capacity of the model trained and evaluated with unbalanced data, and it reveals the level of accuracy of the model and the stability of the decision function. In addition, a log loss metric was used to determine the model's error rate. The categorical cross-entropy loss function is adopted in the model's training, and the log loss value provided by this function is used to analyze how well the probabilities generated by the model match the actual labels. The log loss metric offers a detailed insight into how reliable the estimated probability distribution is, showing the extent to which the model manages uncertainty. The early stopping method was applied to prevent the model from being overfit. This mechanism stops training and restores the best weights if there is no improvement in the verification loss for a specific period. In our study, the validation loss metric was monitored, and if there was no improvement in 100 epochs, the model training was terminated early. The trained model weights were recorded at each epoch to make this process more efficient. In this way, the weights with the lowest validation loss obtained at the end of the training process were reloaded, and the model returned to its best performance. The early stopping mechanism prevented the model from unnecessarily over-training, increasing the validation error and overall generalization capacity. The model trained with this strategy allowed the training and evaluation process to be completed in 50 steps.

As a result, the data processing techniques, metrics, and early stopping mechanisms applied to improve the model's performance and make the training process more efficient have ensured that the experimental results are reliable. The overall success rate of the model was examined in detail within the framework of the specified metrics, and the optimization processes were improved with the validation data at each stage of the performance improvements. Thus, it is proved that the proposed model works effectively in classification tasks and achieves high accuracy rates. The performance metrics of the proposed model during training and validation are given in Figure 4.

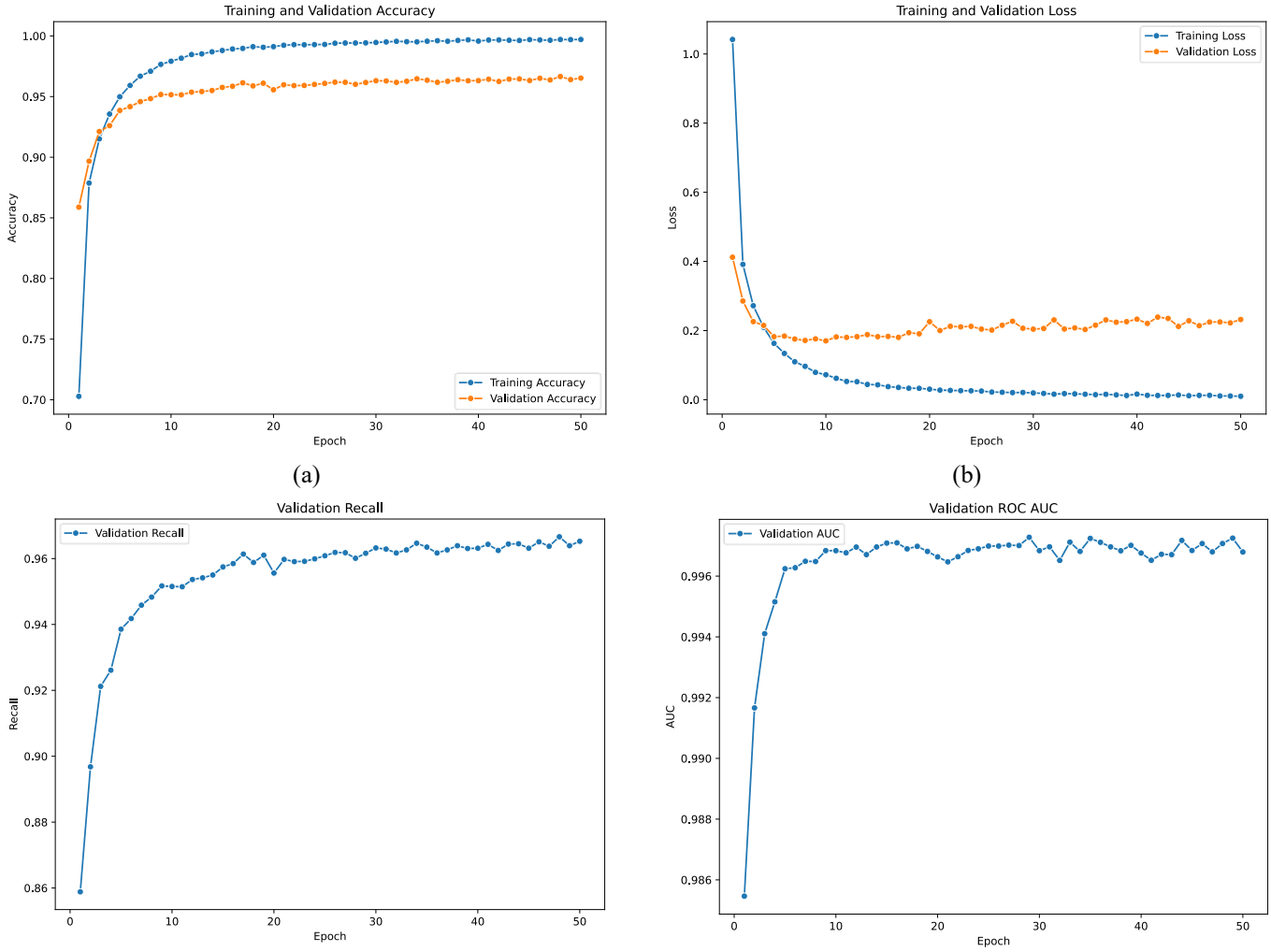


Figure 4. Performance metrics of the model were measured during training and evaluation

The performance of the developed 1D CNN model was comprehensively evaluated on the training and validation datasets through various metrics calculated for each epoch. The model's training runs for 50 epochs; the metrics obtained are presented in detail in Figure 3. The variation of the model's training and validation losses over epochs is visualized in Figure 3-b, while the variation of training and validation accuracies over epochs is visualized in Figure 3-a. The training loss of the model started from 1.04 in the first epoch, decreased steadily as the training process progressed and decreased to 0.01 in the last epoch. This shows that the model has successfully learned the patterns in the training dataset. Similarly, the training accuracy increased from 70.3% in the first epoch to 99.66% in the last. This high accuracy value shows that the model fits the training dataset well.

However, the model's performance on the validation dataset is the most important for assessing its generalization ability. Although the validation loss decreases in parallel with the training loss, it does not reach values as low as the loss. While the validation loss was 0.41 in the first epoch, it reached its lowest value at epoch 10 (0.17) and stabilized around 0.22 with slight fluctuations. The validation accuracy, however, increased from 85.9% in the first epoch to 96.1% in the 18th epoch and then stabilized around 96%. These results show that the model learned the patterns in the training dataset and successfully generalized them to the validation dataset, which it had not seen before. It is clear that the early stopping mechanism does not stop training before 50 epochs due to the halting of the improvement in the validation loss, and the model is trained to its optimum capacity. We also examined the AUC and recall values on the validation dataset to analyze the model's classification performance further. The validation AUC value is above 0.99 for all epochs, indicating that the model can successfully distinguish between positive and negative classes. The validation sensitivity values are also generally above 95%, indicating that the model's false-positive and false-negative predictions are very low. These results show that the model works with high precision, meaning that it both makes accurate predictions and does not miss actual positive samples. The confusion matrix, which shows a detailed overview of the model's correct and incorrect predictions for each clinical outcome class, is presented in Figure 5. The confusion matrix analysis allows for identifying which classes the model predicts better, which classes it struggles with, and which classes it confuses with each other. This analysis enables a better understanding of the strengths and weaknesses of the model.

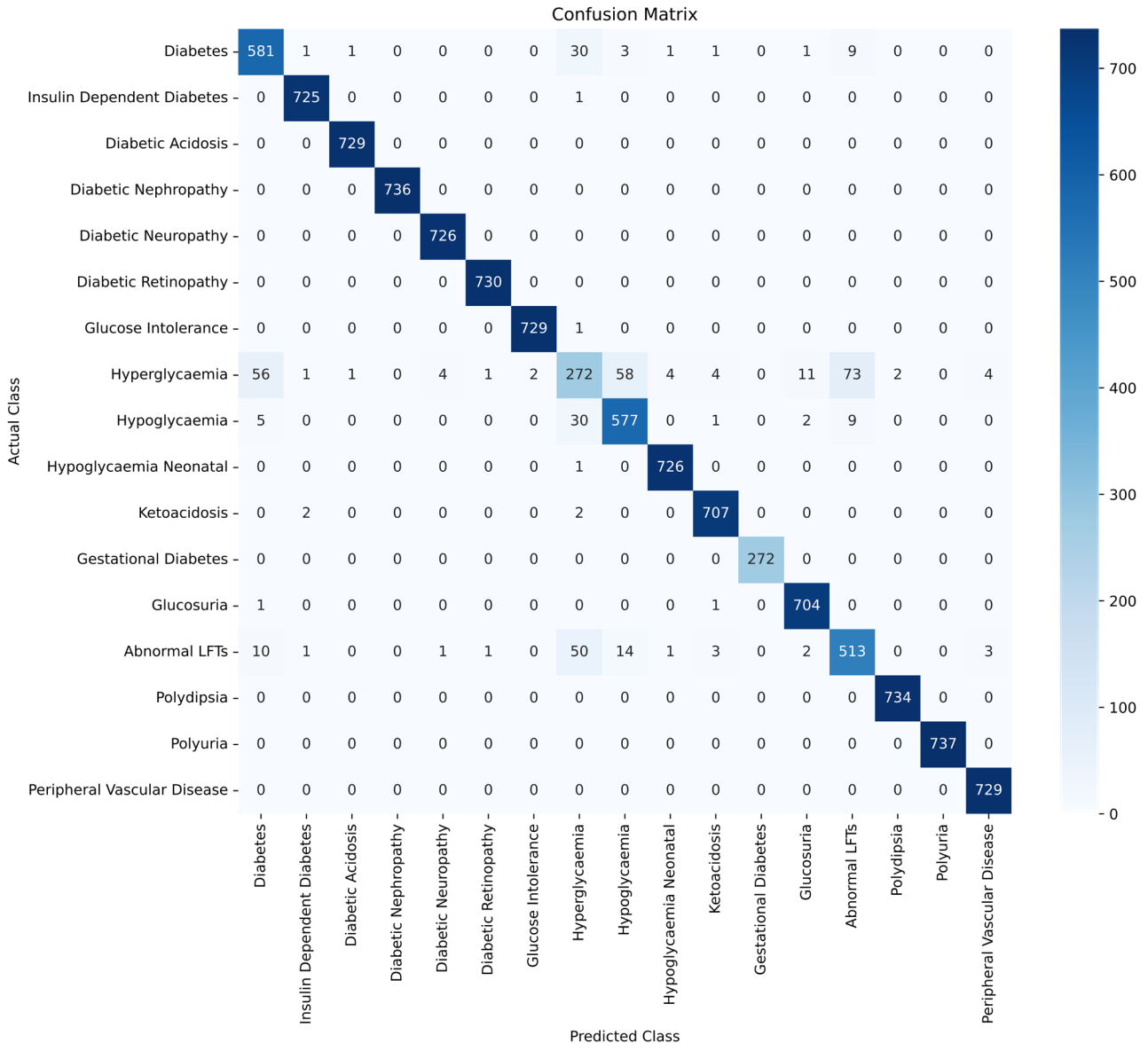


Figure 5. A confusion matrix of the proposed method was obtained during the validation.

The confusion matrix is widely used to evaluate the model's performance in multiclass classification problems. This matrix shows the relationship between the actual classes and the classes predicted by the model. The matrix rows represent the exact classes, and the columns represent the predicted classes. The cells on the diagonal represent the number of instances correctly classified by the model, while the cells off the diagonal represent the number of misclassified cases. As can be seen in Figure 5, the developed model has shown a highly successful classification performance in many different classes. The results show that the developed 1D CNN model can successfully classify the clinical outcomes of adverse effects of insulin-related drugs with high accuracy, sensitivity, and AUC values.

5. Ablation Study and Comparative Performance Evaluation

Several ablation analyses were conducted to evaluate the performance of the DeepInsulin-Net model and to measure the impact of pre-processing steps on classification performance. The performance of the developed model was systematically evaluated to assess the impact of data preprocessing strategies, such as feature selection and class weighting, on the predictive capabilities of various machine learning and deep learning models. The experimental evaluations were performed on a diabetes-related subset of the dataset containing clinically relevant interactions, and the data was stratified and split 7:3 for each scenario as training and validation, maintaining the class distribution. Once the molecular fingerprints and RDKit-based chemical identifiers of the drugs were calculated, four basic preprocessing scenarios were set up: an approach where feature selection and class weighting were used first; a case where only class weighting was applied, and all features were used; a scenario where only feature selection was performed, and class weighting was neglected; and an ablation analysis where no preprocessing steps were applied, and raw features were used. In addition to the proposed CNN model, these cases were

tested using Multilayer Perceptron (MLP), Support Vector Machines (SVM), and Extreme Gradient Boosting (XGBoost) algorithms. Model performances were measured for training and evaluation steps using accuracy, AUC, Cohen's Kappa and Matthews Correlation Coefficient (MCC). The results obtained from the ablation studies are given in Table 3.

Table 3. Performance of ablation analysis and other classification algorithms.

		Train				Validation			
Pre-Processing	Model	Accuracy	AUC	Kappa	MCC	Accuracy	AUC	Kappa	MCC
Feature Selection + Class Weights	CNN	**0,9966**	**0,9990**	**0,9964**	**0,9964**	**0,9403**	**0,9921**	**0,9259**	**0,9259**
	MLP	0,8319	0,8492	0,7702	0,8202	0,8080	0,7922	0,7423	0,7025
	SVM	0,8052	0,8479	0,7417	0,7917	0,7370	0,7387	0,5606	0,5233
	XGBoost	0,8288	0,8490	0,7669	0,8169	0,9111	0,7926	0,7456	0,7059
Class Weights	CNN	0,8136	0,8485	0,7507	0,8007	0,7470	0,7419	0,5712	0,5336
	MLP	0,8203	0,8487	0,7578	0,8078	0,7401	0,7410	0,5638	0,5260
	SVM	0,6733	0,8246	0,6016	0,6516	0,7924	0,7664	0,6195	0,5806
	XGBoost	0,8324	0,8492	0,7707	0,8207	0,9137	0,7929	0,7483	0,7087
Feature Selection	CNN	0,8436	0,8497	0,7826	0,8326	0,9168	0,7943	0,7516	0,7117
	MLP	0,6689	0,8233	0,5969	0,6469	0,7898	0,7654	0,6166	0,5781
	SVM	0,5323	0,7824	0,4518	0,5018	0,6519	0,7205	0,4701	0,4325
	XGBoost	0,6350	0,8140	0,5609	0,6109	0,7568	0,7543	0,5816	0,5439
Non-Pre-processing	CNN	0,6330	0,8122	0,5588	0,6088	0,7557	0,7548	0,5804	0,5425
	MLP	0,5323	0,7824	0,6518	0,6541	0,6519	0,7205	0,6301	0,6325
	SVM	0,6330	0,8122	0,7588	0,7607	0,7557	0,7548	0,7404	0,7425
	XGBoost	0,6689	0,8233	0,7969	0,7982	0,7898	0,7654	0,7766	0,7781

Analyzing the ablation results, it is clear that data preprocessing steps significantly impact classification performance. In the case of no pre-processing, they generally exhibit the weakest validation metrics, demonstrating the challenges posed by the class imbalance and high feature size in the raw data. In this baseline scenario, the XGBoost and SVM models performed slightly better than CNN and MLP, especially on the Kappa and MCC metrics, but their overall performance was limited. This shows that even without pre-processing, some traditional models can capture a specific level of patterns, but the potential for improvement is high. Several changes in the performance of the models were observed when only the feature selection strategy was applied. The CNN model significantly improved in this scenario, increasing the validation accuracy to 91.68% and the Kappa value to 0.7516. This shows that CNN can benefit from a more focused feature set and generalize better with reduced noise. For MLP, SVM, and XGBoost, feature selection alone improved some metrics but showed results more sensitive to imbalance, especially for Kappa and MCC. This indicates that feature selection alone cannot solve the class imbalance problem. Comparisons of different models and pre-processing steps during training are given in Figure 6.

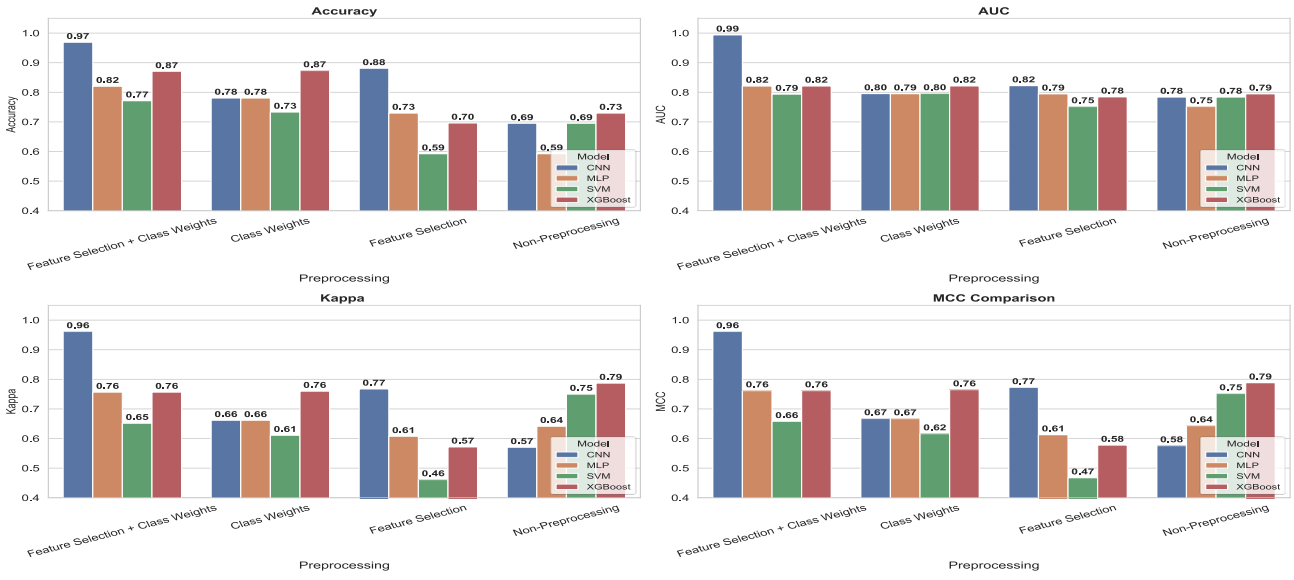


Figure 6. Performance results of different models obtained in training.

Applying only the class weighting strategy significantly increased all models' Kappa and MCC values, especially in the validation set. This indicates that class weighting improves the model's overall predictive stability and reliability by encouraging better learning of minority classes in the dataset. Therefore, it shows that class imbalance in the data is an essential step in classification performance, and the class weighting method applied overcomes this challenge. In the case of class weighting and feature selection, the most comprehensive pre-processing approach and the most fundamental components of the DeepInsulin-Net method, the models showed the highest performance. Under this combined strategy, CNN achieved extremely high and balanced performance metrics such as 94.03% validation accuracy, 0.9921 AUC, 0.9259 Kappa and 0.9259 MCC. On a model-by-model basis, the proposed CNN architecture exhibited a clear superiority over all other tested algorithms, especially when combined with pre-processing strategies. CNN's high training set performance shows that the model can learn data patterns efficiently. In contrast, the high metrics in the validation set indicate a strong generalization capability. XGBoost performed quite competitively and was closest to CNN in all scenarios, especially when class weighting was included. MLP yielded acceptable results, especially with combined pre-processing, while SVM generally underperformed compared to the other models on this dataset and problem definition.

6. Conclusion

This study presents a deep learning-based approach to predicting clinical side effects caused by drug-drug interactions. The proposed method uses molecular fingerprinting and physicochemical properties of drugs to more precisely model the adverse impact of interactions. Seventeen common clinical side effects associated with insulin are selected from the TWOSIDE dataset using a text mining method. Drug pairs were chosen from the dataset, and their molecular properties were calculated using MACCS, Morgan fingerprints and RDKit descriptors. These calculated features using SMILES strings of drug molecules were combined into a single matrix format for each drug pair. In this step, the Variance Thresholding method was applied to the attribute matrix to select the more significant attributes for classification. Drug pairs were used as inputs to a 1D CNN model designed as parallel inputs to predict drug interactions. In the parallel model intended to classify the interactions with high performance, the calculated attributes of the drug pairs are applied to the model as input from different layers, and these attribute matrices are combined in the deep layers of the model and fed to the output layer. The proposed method uses class weighting to balance the unbalanced class distribution. Experimental results show that the proposed model accurately predicts clinical side effects. The model reaches 99.66% accuracy in the training phase, while the validation accuracy is 96%. The training loss decreased to 0.01, while the validation loss stabilized at 0.22. The ROC-AUC metric remained above 0.99 throughout all epochs, and the model demonstrated a strong generalization capability, especially in detecting rare adverse events. These results show that the model can predict common and specific clinical side effects successfully. The developed deep learning architecture allows the features of each drug to be processed independently and combined to predict the probability of occurrence of a specific clinical side effect caused by a drug-drug interaction. This approach increases the model's scalability and reduces the computational burden. The proposed approach can be essential in drug safety assessments and clinical decision support systems. Integrating the model into clinical practice can provide data-driven support for clinicians to make more informed decisions in drug prescribing processes. Implementing these proposed improvements and research directions can significantly improve the current model's performance, robustness and interpretability for predicting adverse drug-drug interactions. Integrating more comprehensive and diverse datasets, applying advanced feature engineering techniques, and exploring innovative model architectures will enable higher accuracy and generalizability, especially in detecting complex and unique interactions. Transparency of the model's decision mechanisms through the integration of explainable artificial intelligence methods will increase clinical acceptability while considering critical factors such as polypharmacy and dose dependency, making predictions more compatible with clinical practice. Future work could develop a more comprehensive solution by integrating the proposed method with different data types, analyzing multiple interacting drug combinations, and its applicability in real-time clinical environments.

References

[1] A. Agustí, E. Melén, D. L. DeMeo, R. Breyer-Kohansal, and R. Faner, “Pathogenesis of chronic obstructive pulmonary disease: understanding the contributions of gene–environment interactions across the lifespan,” *lancet Respir. Med.*, vol. 10, no. 5, pp. 512–524, 2022.

[2] S. R. Chowdhury, D. C. Das, T. C. Sunna, J. Beyene, and A. Hossain, “Global and regional prevalence of multimorbidity in the adult population in community settings: a systematic review and meta-analysis,” *EClinicalMedicine*, vol. 57, 2023.

[3] R. W. Hoel, R. M. G. Connolly, and P. Y. Takahashi, “Polypharmacy management in older patients,” in *Mayo Clinic Proceedings*, Elsevier, 2021, pp. 242–256.

[4] F. Pazan and M. Wehling, “Polypharmacy in older adults: a narrative review of definitions, epidemiology and consequences,” *Eur. Geriatr. Med.*, vol. 12, pp. 443–452, 2021.

[5] J. Wolff *et al.*, “Polypharmacy and the risk of drug–drug interactions and potentially inappropriate medications in hospital psychiatry,” *Pharmacoepidemiol. Drug Saf.*, vol. 30, no. 9, pp. 1258–1268, 2021.

[6] M. R. Mohamed *et al.*, “Association of polypharmacy and potential drug-drug interactions with adverse treatment

outcomes in older adults with advanced cancer," *Cancer*, vol. 129, no. 7, pp. 1096–1104, 2023.

[7] M. Zitnik, M. Agrawal, and J. Leskovec, "Modeling polypharmacy side effects with graph convolutional networks," *Bioinformatics*, vol. 34, no. 13, pp. i457–i466, 2018.

[8] H. Askr, E. Elgeldawi, H. Aboul Ella, Y. A. M. M. Elshaier, M. M. Gomaa, and A. E. Hassanien, "Deep learning in drug discovery: an integrative review and future challenges," *Artif. Intell. Rev.*, vol. 56, no. 7, pp. 5975–6037, 2023.

[9] M. F. Rasool *et al.*, "Assessment of risk factors associated with potential drug-drug interactions among patients suffering from chronic disorders," *PLoS One*, vol. 18, no. 1, p. e0276277, 2023.

[10] P. Rorsman and E. Renström, "Insulin granule dynamics in pancreatic beta cells," *Diabetologia*, vol. 46, pp. 1029–1045, 2003.

[11] N. J. Hogrebe, K. G. Maxwell, P. Augsornworawat, and J. R. Millman, "Generation of insulin-producing pancreatic β cells from multiple human stem cell lines," *Nat. Protoc.*, vol. 16, no. 9, pp. 4109–4143, 2021.

[12] M. A. Hill *et al.*, "Insulin resistance, cardiovascular stiffening and cardiovascular disease," *Metabolism*, vol. 119, p. 154766, 2021.

[13] M. S. Rahman *et al.*, "Role of insulin in health and disease: an update," *Int. J. Mol. Sci.*, vol. 22, no. 12, p. 6403, 2021.

[14] A. Kleinridders, H. A. Ferris, W. Cai, and C. R. Kahn, "Insulin action in brain regulates systemic metabolism and brain function," *Diabetes*, vol. 63, no. 7, pp. 2232–2243, 2014.

[15] J. J. Holst, L. S. Gasbjerg, and M. M. Rosenkilde, "The role of incretins on insulin function and glucose homeostasis," *Endocrinology*, vol. 162, no. 7, p. bqab065, 2021.

[16] J. Blahova, M. Martiniakova, M. Babikova, V. Kovacova, V. Mondockova, and R. Omelka, "Pharmaceutical drugs and natural therapeutic products for the treatment of type 2 diabetes mellitus," *Pharmaceuticals*, vol. 14, no. 8, p. 806, 2021.

[17] R. A. Haeusler, T. E. McGraw, and D. Accili, "Biochemical and cellular properties of insulin receptor signalling," *Nat. Rev. Mol. cell Biol.*, vol. 19, no. 1, pp. 31–44, 2018.

[18] G. Wilcox, "Insulin and insulin resistance," *Clin. Biochem. Rev.*, vol. 26, no. 2, p. 19, 2005.

[19] D. Soumya and B. Srilatha, "Late stage complications of diabetes and insulin resistance," *J Diabetes Metab*, vol. 2, no. 9, p. 1000167, 2011.

[20] O. I. Aruoma, V. S. Neergheen, T. Bahorun, and L.-S. Jen, "Free radicals, antioxidants and diabetes: embryopathy, retinopathy, neuropathy, nephropathy and cardiovascular complications," *Neuroembryology Aging*, vol. 4, no. 3, pp. 117–137, 2007.

[21] C. Argano, L. Mirarchi, S. Amodeo, V. Orlando, A. Torres, and S. Corrao, "The role of vitamin D and its molecular bases in insulin resistance, diabetes, metabolic syndrome, and cardiovascular disease: state of the art," *Int. J. Mol. Sci.*, vol. 24, no. 20, p. 15485, 2023.

[22] J.-M. Guettier and P. Gorden, "Hypoglycemia," *Endocrinol. Metab. Clin.*, vol. 35, no. 4, pp. 753–766, 2006.

[23] L. Sacchetta *et al.*, "Synergistic effect of chronic kidney disease, neuropathy, and retinopathy on all-cause mortality in type 1 and type 2 diabetes: a 21-year longitudinal study," *Cardiovasc. Diabetol.*, vol. 21, no. 1, p. 233, 2022.

[24] B. T. Alemu, O. Olayinka, H. A. Baydoun, M. Hoch, and M. Akpinar-Elci, "Neonatal hypoglycemia in diabetic mothers: A systematic review," *Curr. Pediatr. Res.*, vol. 21, no. 1, 2017.

[25] T. Zhang, J. Leng, and Y. Liu, "Deep learning for drug–drug interaction extraction from the literature: a review," *Brief. Bioinform.*, vol. 21, no. 5, pp. 1609–1627, 2020.

[26] Y. Deng, X. Xu, Y. Qiu, J. Xia, W. Zhang, and S. Liu, "A multimodal deep learning framework for predicting drug–drug interaction events," *Bioinformatics*, vol. 36, no. 15, pp. 4316–4322, 2020.

[27] K. Han *et al.*, "A review of approaches for predicting drug–drug interactions based on machine learning," *Front. Pharmacol.*, vol. 12, p. 814858, 2022.

[28] N.-N. Wang, B. Zhu, X.-L. Li, S. Liu, J.-Y. Shi, and D.-S. Cao, "Comprehensive Review of Drug–Drug Interaction Prediction Based on Machine Learning: Current Status, Challenges, and Opportunities," *J. Chem. Inf. Model.*, vol. 64, no. 1, pp. 96–109, 2023.

[29] A. Akgul, Y. Karaca, M. A. Pala, M. E. Çimen, A. F. Boz, and M. Z. Yildiz, "Chaos Theory, Advanced Metaheuristic Algorithms And Their Newfangled Deep Learning Architecture Optimization Applications: A Review," *Fractals*,

vol. 32, no. 03, p. 2430001, 2024.

[30] M. A. Pala and M. Z. Yıldız, "Improving cellular analysis throughput of lens-free holographic microscopy with circular Hough transform and convolutional neural networks," *Opt. Laser Technol.*, vol. 176, p. 110920, 2024.

[31] M. A. Pala, "Graph-Aware AURALSTM: An Attentive Unified Representation Architecture with BiLSTM for Enhanced Molecular Property Prediction," *Mol. Divers.*, Apr. 2025, doi: 10.1007/s11030-025-11197-4.

[32] X. Li, Z. Xiong, W. Zhang, and S. Liu, "Deep learning for drug-drug interaction prediction: A comprehensive review," *Quant. Biol.*, vol. 12, no. 1, pp. 30–52, 2024.

[33] E. Kim and H. Nam, "DeSIDE-DDI: interpretable prediction of drug-drug interactions using drug-induced gene expressions," *J. Cheminform.*, vol. 14, no. 1, p. 9, 2022.

[34] Z. Li, X. Tu, Y. Chen, and W. Lin, "HetDDI: a pre-trained heterogeneous graph neural network model for drug–drug interaction prediction," *Brief. Bioinform.*, vol. 24, no. 6, p. bbad385, Nov. 2023, doi: 10.1093/bib/bbad385.

[35] M. Asfand-E-Yar, Q. Hashir, A. A. Shah, H. A. M. Malik, A. Alourani, and W. Khalil, "Multimodal cnn-ddi: using multimodal cnn for drug to drug interaction associated events," *Sci. Rep.*, vol. 14, no. 1, p. 4076, 2024.

[36] N. P. Tatonetti, P. P. Ye, R. Daneshjou, and R. B. Altman, "Data-driven prediction of drug effects and interactions.," *Sci. Transl. Med.*, vol. 4, no. 125, p. 125ra31, Mar. 2012, doi: 10.1126/scitranslmed.3003377.

[37] A. Cereto-Massagué, M. J. Ojeda, C. Valls, M. Mulero, S. Garcia-Vallvé, and G. Pujadas, "Molecular fingerprint similarity search in virtual screening," *Methods*, vol. 71, pp. 58–63, 2015.

[38] S. Zhong and X. Guan, "Count-based morgan fingerprint: A more efficient and interpretable molecular representation in developing machine learning-based predictive regression models for water contaminants' activities and properties," *Environ. Sci. Technol.*, vol. 57, no. 46, pp. 18193–18202, 2023.

[39] G. Landrum *et al.*, "rdkit/rdkit: 2024_09_5 (Q3 2024) Release." Zenodo, 2025. doi: 10.5281/zenodo.14779836.

[40] X. Ying, "An overview of overfitting and its solutions," in *Journal of physics: Conference series*, IOP Publishing, 2019, p. 22022.

[41] M. A. F. A. Fida, T. Ahmad, and M. Ntahobari, "Variance threshold as early screening to Boruta feature selection for intrusion detection system," in *2021 13th International Conference on Information & Communication Technology and System (ICTS)*, IEEE, 2021, pp. 46–50.

[42] M. A. PALA, M. E. ÇİMEN, M. Z. YILDIZ, G. ÇETİNEL, E. AVCIOĞLU, and Y. ALACA, "CNN-Based Approach for Overlapping Erythrocyte Counting and Cell Type Classification in Peripheral Blood Images," *Chaos Theory Appl.*, 2022, doi: 10.51537/chaos.1114878.

Conflict of Interest Notice

The author declares that there is no conflict of interest regarding the publication of this paper.

Artificial Intelligence Statement

Artificial intelligence tools were used solely for language and grammar correction during the preparation of this manuscript. No AI tools were involved in the research design, data analysis, interpretation of results, or writing of the scientific content. All intellectual contributions and scholarly work were carried out exclusively by the author.

Plagiarism Statement

This article has been scanned by iThenticate ™.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
http://saucis.sakarya.edu.tr/
e-ISSN: 2636-8129
Publisher: Sakarya University
Vol. 8, No. 2, 260-272, 2025
DOI: 10.35377/saucis..1637290
SAKARYA UNIVERSITY
RESEARCH ARTICLE

Feature Enhancement of TUM-RGBD Depth Images and Performance Evaluation of Gaussian Splatting-Based SplaTAM Method

Cemil Zeyveli[1*], Ali Furkan Kamanlı[2]

[1]Department of Electrical and Electronics Engineering, Faculty of Engineering, Karabük University, Karabük, Türkiye
[2]Department of Electrical and Electronics Engineering, Faculty of Technology, Sakarya University of Applied Science, Sakarya, Türkiye

Corresponding author:
Cemil Zeyveli,
Department of Electrical and Electronics Engineering, Faculty of Engineering, Karabük University,
Karabük, Türkiye
cemilzeyveli@karabuk.edu.tr

Article History:
Received: 11.02.2025
Revised: 02.05.2025
Accepted: 05.06.2025
Published Online: 16.06.2025

ABSTRACT
Simultaneous Localization and Mapping (SLAM) methods are used in autonomous systems to determine their locations in unknown environments and map these environments. Autonomous systems need to act autonomously without external intervention. These methods are widely used in robotics and AR/VR applications. Gaussian Splatting SLAM is a Visual SLAM method that performs mapping and localization using depth and RGB images and uses Gaussian structures for scene representation. Popular datasets such as TUM-RGBD, Replica, and Scannet++ are used in the performance evaluation and testing of the visual SLAM methods. However, the depth images in the TUM-RGBD dataset are of lower quality than other datasets. This problem negatively affects the depth data's accuracy and reduces the quality of mapping results. In this study, to increase the quality of depth images, the features of depth images were corrected using the median filter, which is the depth smoothing method, and a cleaner depth dataset was obtained. The new dataset obtained was processed using the Gaussian Splatting SLAM method, and better metric results (PSNR, SSIM, and LPIPS) were obtained compared to the original dataset. As a result, in the dataset with corrected features, an improvement of 8.08% in the first scene and 4.69% in the second scene was achieved according to metric values compared to the original dataset.

Keywords: SLAM, Gaussian Splatting, Median Filter

1. Introduction

Simultaneous Localization and Mapping (SLAM) is the problem of simultaneous mapping and positioning autonomous systems in an unknown environment [1]. SLAM is widely used in indoor and outdoor environments, underwater and aerial vehicles, and computer vision applications [2]. The solution methods for the SLAM problem are based on the work conducted by Smith and Cheeseman in 1986. The focus of this study was estimating the robot's position using sensor data [3]. In subsequent studies, algorithms have been developed to create 2D and 3D maps using various sensors [4]. SLAM methods are divided into two main groups, according to the types of sensors used: laser-based SLAM and vision-based SLAM methods [5]. In the first studies on SLAM, laser-based systems were preferred due to the cameras' low resolution and high cost. However, in recent years, the low cost of cameras, ease of integration and advantages of providing richer environmental information compared to lidar-based systems have led to the preference for monocular, stereo and RGB-D camera-based visual SLAM methods [6].

Visual SLAM methods can be divided into two main types, direct (intensive) and feature-based, depending on how the information extracted from the images is used [7]. While direct methods estimate camera motion using the intensity and color values of each pixel in the images, feature-based methods perform camera pose estimation and mapping by matching corner, line and plane features in multiple images [8].

Dense Visual SLAM methods generally use explicit scene representation. These representations include voxel grids, where the three-dimensional space is discretized by dividing it into regular small cubes, point clouds that represent the surfaces in the scene with a series of discrete point clusters, signed distance fields that provide surface information by calculating the distance of each point in space to the nearest surface and the positive or negative value of this distance, and neural scene representations created by encoding scene features through neural networks [9,10]. Although these scene representations have made progress, particularly in localization, they still face difficulties obtaining dense maps with high levels of detail [11].

To overcome these problems, the SplaTAM [10] method, based on the 3D Gaussian Splatting method, was developed in

Cite as: C. Zeyveli & A. F. Kamanlı. (2025). Feature enhancement of TUM-RGBD depth images and performance evaluation of gaussian splatting-based SplaTAM method. *Sakarya University Journal of Computer and Information Sciences*. vol. 8, no. 2, pp. 260-272, 2025 doi: 10.35377/saucis...1637290

2024. SplaTAM is a method that uses 3D Gaussians as scene representation and performs location optimization with scene geometry with these structures [10]. The Gaussian Splatting method provides better noise management. Compared to other methods, it can create dense maps by offering advantages such as continuous surface modeling, low memory consumption and fast image creation [12]. With these features, SplaTAM stands out as a strong alternative for Dense SLAM applications. Popular datasets such as TUM-RGBD are widely used in the literature to compare the performance of visual SLAM methods and evaluate them according to metric values. The TUM-RGBD dataset consists of RGB and depth images. Visual SLAM methods perform camera pose estimation and 3D reconstruction using these datasets. Therefore, the quality of RGB and depth images in the datasets is important for the method's performance. However, low-quality sensors were used to create the TUM-RGBD dataset, and the depth information contained in the depth images became weak and limited [10].

In this study, the depth images of two sub-scenes in the TUM-RGBD dataset were improved using the median filter, the depth smoothing method. The newly obtained datasets were processed using the SplaTAM method, and compared to the original datasets, 3D image quality and structural similarity improvements of 8.08% were achieved in the first scene and 4.69% in the second scene.

2. Related Work

2.1. Dense V-SLAM and Gaussian Splatting-Based SLAM Methods

Dense SLAM is a visual SLAM method that creates a high-resolution map by extracting dense information from each scene pixel. DTAM [13] is the first method that performs dense 3D reconstruction using all pixel information. However, although it provides high accuracy, it has a high computational cost. Kinect Fusion [14] performs camera tracking using the ICP algorithm and scene representation using the TSDF method. Elastic Fusion [15] is a method that optimizes memory usage and increases accuracy in large-scale environments with its loop closure feature. BAD-SLAM [16] performs dense scene representation with the Bundle Adjustment approach but has a high computational cost. Droid-SLAM [17] is a deep learning-based Dense SLAM method that provides successful results, especially in dynamic scenes.

Neural Radiance Field (NeRF) based methods have recently been developed for scene representations. iMAP [18] is a method that represents the scene with a single MLP network. However, this method suffers from catastrophic forgetting problems in large-scale scenarios. NICE-SLAM [19] and Vox-Fusion [20] provide high accuracy in large-scale scenes by optimizing implicit scene representation methods. Co-SLAM [21] combines hash-grid and one-blob encoding methods to achieve high-quality scene reconstruction. However, NeRF-based methods are limited in real-time performance due to high computational cost and memory requirements.

Gaussian Splatting-based SLAM methods use 3D Gaussian structures for scene representation. This method uses differential rasterization to perform scene modeling with high accuracy and low cost. MonoGS [22] can accurately reconstruct small and transparent objects, while GS-SLAM [23] performs optimized camera pose estimation using 3D Gaussian representations with RGB-D data. SplaTAM provides an explicit volumetric model using 3D Gaussian structures for scene representation. This method calculates the photometric losses more efficiently than other dense visual SLAM (V-SLAM) methods. It determines the gaps in the scene using a silhouette mask and thus performs a high-accuracy and high-quality mapping process. In addition, the map can be expanded by adding Gaussian components to the existing map. This feature allows SplaTAM to work effectively in large-scale environments. The SplaTAM method simplifies system setup and allows wide application areas to work with a single unposed RGB-D camera [10].

2.2. Datasets and Depth Smoothing Methods

Visual SLAM methods are evaluated using benchmark datasets such as Replica [24], ScanNet [25], ScanNet++ [26], and TUM-RGBD [27]. The ScanNet++ dataset is large and contains high-quality DSLR and iPhone RGB-D images. The Replica dataset provides photo-realistic indoor scenes with HDR renderings and dense 3D mesh structures. The TUM-RGBD dataset is generated using the Microsoft Kinect v1 sensor and is typically used only for camera tracking evaluations due to its low-quality depth data.

In computer vision systems, environmental factors and hardware limitations cause distortions in-depth images and low-resolution image problems. These problems make it difficult to represent fine details in-depth images accurately [28]. Although traditional linear filtering methods (Gaussian [29], Mean [30], Wiener [31]) are effective in eliminating problems in-depth images, they are insufficient in preserving critical features such as edges and corners [32]. Therefore, the nonlinear Median Filter [33] method is preferred to reduce distortions at the edges of object features in-depth images and improve image quality. The median filter performs this operation using the median value of the pixels, thus providing the advantage of preserving the edge and sharp structure features in the image compared to other filtering methods [34].

3. Method

The SplaTAM method is a dense SLAM method that can perform precise camera tracking and high-precision map reconstruction in challenging real-world scenes. The online optimization method uses an explicit volumetric representation approach called 3D Gaussian Splatting. This representation can process color and depth information with high accuracy and speed.

3.1. Gaussian Scene Representation

In the SplaTAM method, the scene map is represented by a set of 3D Gaussians. Each Gaussian structure consists of color (RGB), center position (μ), radius (r) and opacity (o). The Gaussian function below in Equation 1 calculates the contribution of each 3D Gaussian structure used for scene representation at a specific point in the scene.

$$\mathrm{f(x)} = o \cdot \exp\left(-\frac{\| x - \mu \|^2}{2\mathrm{r}^2}\right) \tag{1}$$

In this function, the distance of a given point in the scene of the 3D Gaussian distribution is calculated to determine the area of influence of the Gaussian distribution. The obtained distance value is normalized with the radius, and an exponential function is applied. Consequently, points close to the center of the distribution are assigned to a higher degree of influence, and distant points are assigned to a lower degree of influence.

3.2. Rendering Process of RGB, Depth and Silhouette Images

The SplaTAM method generates RGB, depth and silhouette images for each camera frame by representing the scene representation with 3D Gaussian structures. This method optimizes the scene map and camera parameters by applying a differential rendering process. When processing RGB images, 3D Gaussian structures are arranged from front to back according to their distance from the camera (depth value). Each Gaussian structure is projected onto a two-dimensional plane in the following step. Subsequently, the color and opacity contributions of each Gaussian in the pixel plane are calculated. A pixel's final color is formed by combining the colors and opacities of the Gaussian structures overlapping that pixel.

The pixel plane's color, depth, and silhouette values are obtained using the common formula of Equation 2. The coefficient v_i in the equation represents the weight value specific to the processed image. The value $f_i(p)$ indicates the opacity coefficient of the *i*-th Gaussian component at pixel p. The inverse opacity effect of previously processed Gaussian structures is calculated in the remaining part of the equation.

$$G(p) = \sum_{i=1}^{n} v_i f_i(p) \prod_{j=1}^{i-1} \Big(1 - f_j(p)\Big) \tag{2}$$

The color of a pixel is calculated using the following Equation 3. Each Gaussian component is represented by a color value ($v_i = c_i$). In the Gaussian function, the first Equation 1, the contribution of the Gaussian component to the pixel is obtained by multiplying the calculated opacity value with the color value. The inverse effect of the opacities of all Gaussian components preceding the relevant Gaussian representation in the current pixel is calculated. Therefore, the color value of each Gaussian component is weighted by the opacity effect in the current pixel and the inverse effect of the opacities of the previous structures, and the sum of the contributions of all structures determines the final color of the pixel.

$$C(p) = \sum_{i=1}^{n} c_i f_i(p) \prod_{j=1}^{i-1} \Big(1 - f_j(p)\Big) \tag{3}$$

The positions of the 3D Gaussian structures are calculated as their distances from the camera reference frame. A depth map representing scene geometry is then created using this information. Equation 4 is used to calculate the depth value for each pixel. This depth value is calculated by weighting the depth value ($v_i = d_i$) of the corresponding Gaussian component with its opacity value and the inverse effect of the opacities of its previous components.

$$D(p) = \sum_{i=1}^{n} d_i f_i(p) \prod_{j=1}^{i-1} \Big(1 - f_j(p)\Big) \tag{4}$$

A silhouette image is rendered to verify the presence of information in a pixel within the current map and its representation in the current scene map. The contribution of each Gaussian component to the silhouette is weighted by its opacity value and aliasing effects of the preceding components in Equation 5. This process identifies map gaps and underrepresented areas while optimizing camera tracking and map update processes.

$$S(p) = \sum_{i=1}^{n} f_i(p) \prod_{j=1}^{i-1} \Big(1 - f_j(p)\Big) \tag{5}$$

3.3. SLAM System

The SLAM system is based on 3D Gaussian representation and differential rendering methods. The system consists of camera tracking, Gaussian densification and map update steps. In the first frame, the camera position is not optimized. The camera position is determined as the starting position. This position is regarded as a reference point for the pose information that is subsequently provided. Similarly, new Gaussian structures were recreated using all pixels in the first frame. The camera

position is optimized according to the current map information in the novel frames after the initial frame. Camera tracking is based on a loss function calculated with RGB and depth information. This loss function takes the difference between the rendered RGB, depth, and silhouette images and the real RGB-D images. Using the loss function, the camera position is determined.

The following loss function in Equation 6 is developed to measure the difference between the rendered depth and color values and the real values on pixels sufficiently represented in the scene map ($S(p) > 0.99$). For each pixel, the depth difference $L_1(D(p))$ and color difference $0.5L_1(C(p))$ are calculated and summed. This total value is used to optimize the camera position. The 0.5 coefficient applied to the color difference term was determined experimentally by observing that depth difference values typically range between [0.002, 0.006] while color difference values range between [0.01, 0.03]. This scaling factor balances the different value ranges, preventing the color term from dominating the loss function during optimization [10].

$$L_t = \sum_p (S(p) > 0.99)\left(L_1(D(p)) + 0.5L_1(C(p))\right) \tag{6}$$

The Gaussian densification step is a process that provides a more detailed and dense representation of the map by adding new Gaussians to the scene for new frames. The densification mask in Equation 7 is used to identify regions in the map that are weak or underrepresented.

$$M(p) = (S(p) < 0.5) + \left(D_{\mathrm{GT}}(p) < D(p)\right)\left(L_1(D(p)) > \lambda\mathrm{MDE}\right) \tag{7}$$

If the silhouette value of the pixel is below the threshold value in the formula and the input depth value $(D_{\mathrm{GT}}(p))$ is smaller than the rendered depth $(D(p))$, a new Gaussian structure is added to the pixel. Additionally, if the absolute difference between the rendered and actual depth is greater than the median depth error (MDE represents the median value of all absolute differences between the rendered and actual depth values across the image), a Gaussian structure is created based on this error. Finally, in the map update step, the current 3D Gaussian map is optimized using the differential rendering process and gradient-based optimization method, starting from fixed camera poses. In this process, only the keyframe and previous frames that overlap with the current frames are considered, thereby reducing the processing load. All pixels are optimized without using a silhouette mask. Consequently, a more efficient and accurate map representation is provided. The operational system of the SLAM method is illustrated in Figure 1.

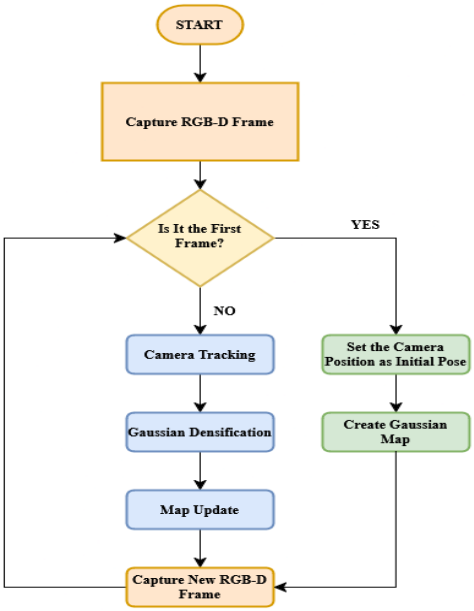


Figure 1. The Operational System of the SLAM Method

3.4. TUM-RGBD Dataset

The TUM-RGBD dataset was developed using a Microsoft Kinect v1 sensor to objectively evaluate the performance of visual odometry and SLAM systems and compare them with different methods. The dataset contains synchronized RGB and depth images and camera position information in PNG format with a resolution of 640*480. Camera position information is used in SLAM systems to measure the error rate of camera pose estimations and evaluate the generated map's accuracy.

RGB images are recorded in 8-bit format, and depth images are in 16-bit single-channel monochrome (grayscale) format. Depth images contain problems caused by low-quality sensors, ambient light and environmental conditions. This negatively affects the accuracy of depth maps and the performance of SLAM systems.

For instance, the distortions edges of the depth image of the joystick object in Figure 2 and the distortions in the regions marked in Figure 3 increase the error rate of the SplaTAM method in processing the depth map.

Figure 2: Distortions on the Edges of the Joystick Object in the Depth Image

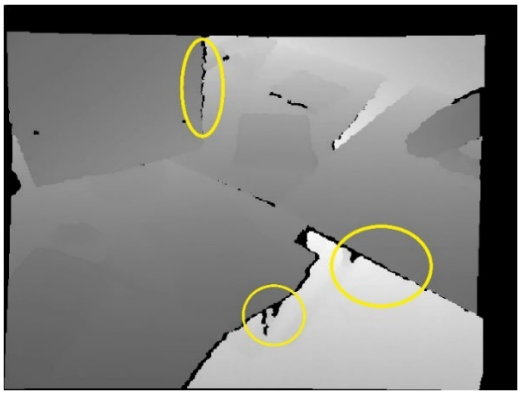


Figure 3: Edge Distortions in Depth Image

Because of these problems in the depth dataset, the TUM-RGBD dataset is generally not preferred for comparing metrical results of visual SLAM methods. To solve these problems, the median filter method improved objects' edge and corner features in-depth images.

3.5. The Median Filter

The median filter is a non-linear filter widely used in image processing. It is especially used to preserve and enhance sharp features such as edges and corners in images. Although different types of filters perform this function, the key advantage of the median filter is that it preserves the quality of the features in the image during processing and prevents blurring. Therefore, the median filter is preferred in applications where sharpness and details must be preserved.

The median filter is applied by operating all pixels in an image. Figure 4 shows the working principle of the median filter. The median filtering process consists of the following steps for each pixel in the image:

1. **Determining the Kernel Size:** A kernel size is determined around the relevant pixel to which the filter is applied.
2. **Sorting Pixel Values:** All pixel values within the kernel size are sorted from smallest to largest, and these values' median (median value) are calculated.
3. **Updating Center Pixel:** The value of the pixel located in the center of the kernel size is replaced with the calculated median value.

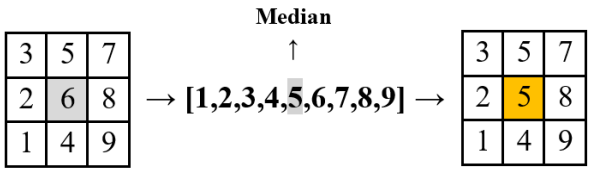


Figure 4: Median Filter Working Principle

In this study, images were processed using the median filter function of the SciPy library. This function uses size, mode and cval parameters to control the median filter.

The size parameter determines the kernel size used during the filtering. The mode parameter determines how to fill empty values when the specified kernel size is outside the image. Reflect, Constant, Nearest, Mirror and Wrap modes are mode parameters that can be used for the median filter.

The Reflect mode fills the pixel values outside the image borders in the specified kernel size by symmetrically reflecting the pixel values at the image borders (d c b a | a b c d | d c b a).

The Constant mode fills the pixel values outside the image boundaries in the specified kernel size with a constant value determined by the cval parameter (k k k k | a b c d | k k k k).

The Nearest mode fills the pixels that are outside the image borders in the specified kernel size with the value of the nearest border pixel to the relevant pixel (a a a a | a b c d | d d d d).

The Mirror mode operates similarly to the Reflect mode. The values in an inner pixel from the boundary pixel values are reflected to the empty pixels using the “mirroring” method (d c b | a b c d | c b a).

The Wrap mode places pixel values at the image boundaries by wrapping them around pixels outside the kernel size (a b c d | a b c d | a b c d).

All median filter modes explained above were applied to the depth images in the TUM-RGBD dataset, and new depth images were obtained. The kernel size used in the median filter is an important parameter affecting the filtering process's effectiveness and quality.

Determining the kernel size too large effectively removes image distortions. However, it causes the loss of fine details in the image. On the contrary, very small kernel sizes preserve edge and corner features in the depth image. However, it is insufficient to improve the features of the image. Therefore, determining the optimum kernel size is important for improving the features of in-depth images and increasing the mapping quality obtained with the SplaTAM method.

To evaluate the effect of kernel size and filter modes on the performance of depth images, kernel sizes were set to 5, 9, 11, 13, 15 and 19, respectively, and new datasets were created using each filter mode. To ensure balance and optimal results, the minimum kernel size was set to 5 and the maximum to 19.

Figure 5 shows the effect of the median filter on the distortions at the object's boundaries in the depth image by applying it to different kernel sizes.

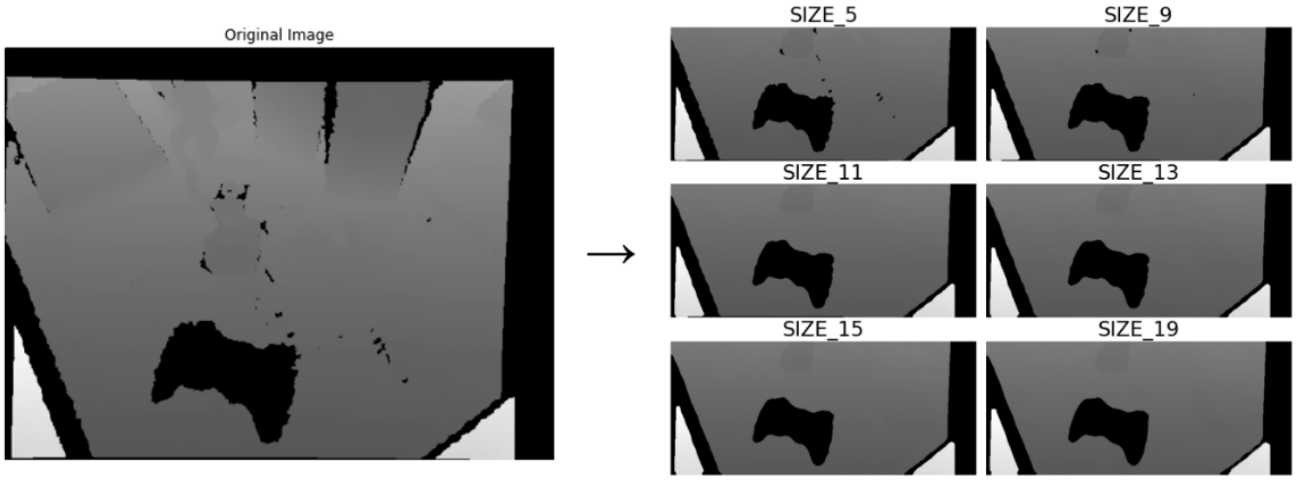


Figure 5: The Effect of The Median Filter on Depth Images at Different Kernel Sizes

Figure 6 shows the effect of the median filter on the distortions at the edge of the table in the depth image by applying it to different kernel sizes.

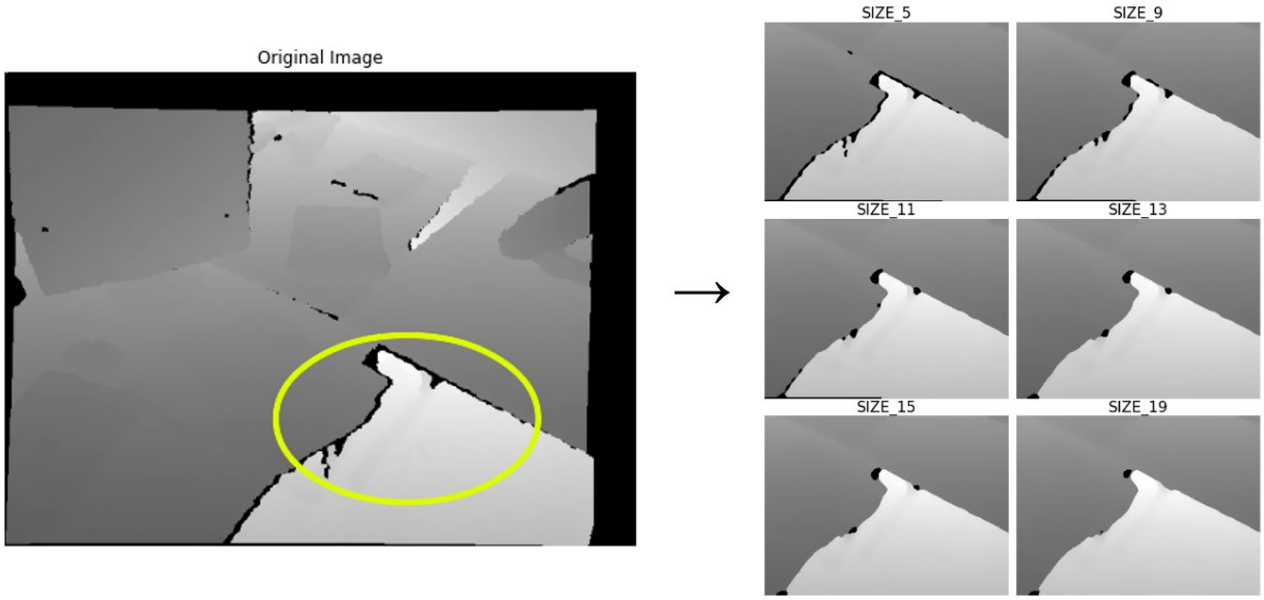


Figure 6: The Effect of The Median Filter on Depth Image Edge Distortions at Different Kernel Size

4. Result and Discussion

This section compares the results of median filtering applied to depth images of two scenes in the TUM-RGBD dataset. The effects of new depth datasets created as a result of the filtering process on the rendering quality in the SplaTAM method are examined according to Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS) metrics.

The PSNR measures the deviation of the rendered image from the original image on a pixel basis. A high PSNR value means that the rendered image is closer to the original image and that the image quality is high [35]. SSIM value provides an evaluation method based on the human visual system. It compares the structural similarities of images by considering their brightness, contrast and structural components. A high SSIM value indicates that the original image and the rendered image are structurally similar to each other [36]. LPIPS value is a metric that uses a deep learning-based method to measure the perceptual similarity between two images. It extracts features from images using a convolutional neural network model previously trained on image classification. LPIPS value takes values between zero and one, and images with values closer to zero indicate higher similarity to the original image [36].

The sub-dataset named “fr1/desk” of the TUM-RGBD dataset consists of 613 RGB images and 595 depth images. When this dataset's original depth and RGB images are rendered with the SplaTAM model, the PSNR value is 21.78, the SSIM value is 0.849, and the LPIPS value is 0.241.

Table 1 shows the metric results of the new depth datasets obtained by applying the median filter to the depth images in the "fr1/desk" sub-dataset of the TUM-RGBD dataset. The table presents the median filter results separately for each filter mode and different kernel sizes. The coloring on the table highlights the best results among the results obtained with different kernel sizes in each filter mode. Green, yellow, and orange represent the best, second-best, and third-best results, respectively.

The PSNR value of the original dataset is 21.78, and applying the median filter to the depth images resulted in a general increase in the PSNR values. The highest PSNR value is 23.54 in the Wrap mode (kernel size 13). This value shows a significant improvement compared to the original dataset. Furthermore, the second and third highest PSNR values are 23.35 in Mirror mode (kernel size 15) and 23.32 in Nearest mode (kernel size 19). If the SSIM values are compared, the SSIM result of the original dataset is 0.849, which is a similar improvement to the PSNR results. After applying the median filter, the highest SSIM value is 0.907 in Wrap mode (kernel size 13). The second highest SSIM value is 0.897 in Mirror mode (kernel size 15), and the third best value is 0.895 in Nearest mode (kernel size 19). Finally, when the LPIPS results are evaluated, the LPIPS value of the original dataset is 0.241. The best LPIPS value is 0.178 in Wrap mode (kernel size 13). The second lowest value is 0.199 in Nearest mode (kernel size 19), and the third best value is 0.200 in Mirror mode (kernel size 15).

Overall, the results show that the median filtered dataset has higher metric results than the original dataset. However, increasing the kernel size above a certain value leads to losing fine details and feature loss of in-depth images. When the table is analyzed, it is seen that although there are exceptions in some modes, the general trend is that when the kernel size is increased excessively, the metric results decrease. This shows that selecting the optimum kernel size is critical in processing depth images. Detailed results are shown in Table 1.

The processing applied to the first scene of the TUM-RGBD dataset was also applied to the second scene, “fr1/desk2”. The PSNR value obtained from the original depth image of the second scene is 20.30. According to the median filter applied, the highest PSNR value is 21.30 in Mirror mode (kernel size 11). The second-best value is 21.07 in the Constant mode (kernel size 15), and the third-best value is 20.84 in the Nearest mode (kernel size 5). According to the original dataset results, the SSIM value is 0.781. When the Median filter was applied, the highest SSIM value is 0.830 in Mirror mode (kernel size 11). The second highest value is 0.813 in Constant mode (kernel size 15), and the third highest is 0.806 in Reflect mode (kernel size 15). The LPIPS value is 0.271 in the original dataset. It can be observed that the LPIPS values obtained after applying the median filter are not higher than the original result.

Compared to the first scene, the second scene of the TUM-RGBD dataset consists of RGB and depth images with a wider perspective, even though the dataset was created from the same scene environment. This suggests that the depth images in the scene “fr1/desk2” contain less detail than in the first scene. Therefore, although the median filter applied to the depth images yielded favorable results regarding PSNR and SSIM, it performed worse than the original dataset regarding the LPIPS parameter measuring perceptual similarity. This suggests that in less detailed scenes, the effect of the median filter on perceptual similarity is limited, and therefore, LPIPS values increase. Table 2 shows the effects of the median filter on PSNR, SSIM and LPIPS metrics different parameters.

The Bilateral Filter was also tested to improve the depth of images. The Bilateral Filter is a filtering method for preserving edges and fine details. The results of the Bilateral Filter applied to the “fr1/desk” sub-dataset of the TUM-RGBD dataset are shown in Table 3. When the results were examined, a decrease rather than an improvement in metric values was observed compared to the original dataset. The decrease in PSNR and SSIM values and the increase in LPIPS values as the filter window size increased indicated that this filter was not a suitable method for the study's objectives.

Table 1: The Metrical Results on TUM-RGBD “fr1/desk” Sub-dataset

Kernel Size	Metrics			Constant	Mirror	Nearest	Reflect	Wrap
Original	PSNR	↑	21.78					
	SSIM	↑	0.849	-	-	-	-	-
	LPIPS	↓	0.241					
Size 5	PSNR	↑		22.55	21.38	22.75	22.32	22.48
	SSIM	↑	-	0.869	0.844	0.882	0.856	0.869
	LPIPS	↓		0.217	0.243	0.219	0.230	0.211
Size 9	PSNR	↑		22.01	21.84	22.27	22.45	22.91
	SSIM	↑	-	0.843	0.856	0.871	0.891	0.880
	LPIPS	↓		0.257	0.246	0.222	0.205	0.218
Size 11	PSNR	↑		22.12	23.23	21.83	22.55	22.49
	SSIM	↑	-	0.856	0.884	0.857	0.862	0.865
	LPIPS	↓		0.250	0.213	0.241	0.240	0.235
Size 13	PSNR	↑		22.50	21.74	22.38	22.83	23.54
	SSIM	↑	-	0.866	0.856	0.892	0.873	0.907
	LPIPS	↓		0.245	0.255	0.207	0.218	0.178
Size 15	PSNR	↑		22.57	23.35	21.69	23.02	21.94
	SSIM	↑	-	0.877	0.897	0.858	0.876	0.881
	LPIPS	↓		0.233	0.200	0.253	0.228	0.229
Size 19	PSNR	↑		21.97	22.08	23.32	21.65	22.45
	SSIM	↑	-	0.864	0.872	0.895	0.866	0.882
	LPIPS	↓		0.246	0.244	0.199	0.241	0.220

Table 2: The Metrical Results on TUM-RGBD “fr1/desk2” Sub-dataset

Kernel Size	Metrics			Constant	Mirror	Nearest	Reflect	Wrap
Original	PSNR	↑	20.30					
	SSIM	↑	0.781	-	-	-	-	-
	LPIPS	↓	0.271					
Size 5	PSNR	↑		19.76	19.35	20.84	20.04	20.11
	SSIM	↑	-	0.765	0.761	0.797	0.800	0.780
	LPIPS	↓		0.305	0.310	0.281	0.287	0.298
Size 9	PSNR	↑		20.11	19.79	19.80	20.06	20.79
	SSIM	↑	-	0.791	0.783	0.768	0.773	0.773
	LPIPS	↓		0.301	0.314	0.305	0.302	0.287
Size 11	PSNR	↑		18.57	21.30	20.72	20.04	20.58
	SSIM	↑	-	0.734	0.830	0.787	0.799	0.777
	LPIPS	↓		0.330	0.278	0.293	0.301	0.303
Size 13	PSNR	↑		19.23	19.42	20.67	20.28	19.61
	SSIM	↑	-	0.760	0.760	0.796	0.772	0.765
	LPIPS	↓		0.329	0.336	0.308	0.313	0.323
Size 15	PSNR	↑		21.07	20.05	20.64	20.35	20.48
	SSIM	↑	-	0.813	0.770	0.775	0.806	0.790
	LPIPS	↓		0.282	0.310	0.291	0.303	0.296
Size 19	PSNR	↑		20.69	19.81	20.62	20.61	20.78
	SSIM	↑	-	0.800	0.785	0.792	0.784	0.764
	LPIPS	↓		0.300	0.315	0.304	0.319	0.314

Table 3: The Bilateral Filter Metrical Results on TUM-RGBD “fr1/desk” Sub-dataset

Metrics		Original	Size 5	Size 9	Size 11	Size 13	Size 15	Size 19
PSNR	↑	21.78	21.20	17.24	15.85	13.35	13.65	13.95
SSIM	↑	0.849	0.823	0.679	0.617	0.502	0.469	0.500
LPIPS	↓	0.241	0.269	0.415	0.469	0.563	0.565	0.511

Although the main focus of the study was to examine the effects of depth image enhancements on 3D visual quality metrics (PSNR, SSIM, and LPIPS), the effect of filtered depth images on the camera-pose estimation ATE RMSE (Absolute Trajectory Error Root Mean Square Error) of the SplaTAM method was also investigated. Tables 4 and 5 show the camera pose estimation errors processed with the Median Filter method using different kernel sizes and filter modes for the "fr1/desk" and "fr1/desk2" sub-datasets, respectively. The ATE RMSE value for the original dataset is 3.35 for the "fr1/desk" sub-dataset and 6.54 for "fr1/desk2". When examining the results, it can be observed that the Median Filter process improves visual quality and enhances camera pose estimation accuracy.

Table 4: Camera-Pose Estimation Results on TUM-RGBD “fr1/desk” Sub-dataset

Kernel Size		Constant	Mirror	Nearest	Reflect	Wrap
Original	3.35	-	-	-	-	-
Size 5	-	3.35	3.36	3.32	3.33	3.39
Size 9	-	3.50	3.29	3.30	3.33	3.33
Size 11	-	3.42	3.34	3.33	3.29	3.29
Size 13	-	3.41	3.32	3.32	3.31	3.31
Size 15	-	3.39	3.33	3.31	3.30	3.28
Size 19	-	3.31	3.28	3.29	3.32	3.30

Table 5: Camera-Pose Estimation Results on TUM-RGBD "fr1/desk2" Sub-dataset

Kernel Size		Constant	Mirror	Nearest	Reflect	Wrap
Original	6.54	-	-	-	-	-
Size 5	-	6.63	6.59	6.48	6.50	6.20
Size 9	-	6.64	6.43	6.43	6.44	6.30
Size 11	-	6.53	6.32	6.62	6.62	6.18
Size 13	-	6.64	6.33	6.50	6.58	6.07
Size 15	-	6.56	6.13	6.60	6.26	6.03
Size 19	-	6.70	6.45	6.31	6.34	6.22

The results of the positive effects of the median filter are presented in Figure 7. Figure 7 shows the depth and rendered images of two scenes of the TUM-RGBD dataset. In the first row with the joystick object, when the lowest PSNR value is compared to the best one, it is seen that there is less distortion on the edges of the game controller, and the transitions are smoother. Focusing on the ends of the papers on the edge of the table in the second row, the image with the lowest PSNR shows significant distortion on the edges and corners. In the image with the best PSNR value, the corners of the papers are clearer, and the transitions are sharp.

The images in the third row are obtained from the second scene "fr1/deks2" dataset and present a more general and wide-angle scene than the first. Comparing the images with the lowest and highest PSNR, it can be seen that there is less distortion in the objects on the table and the details on the monitors.

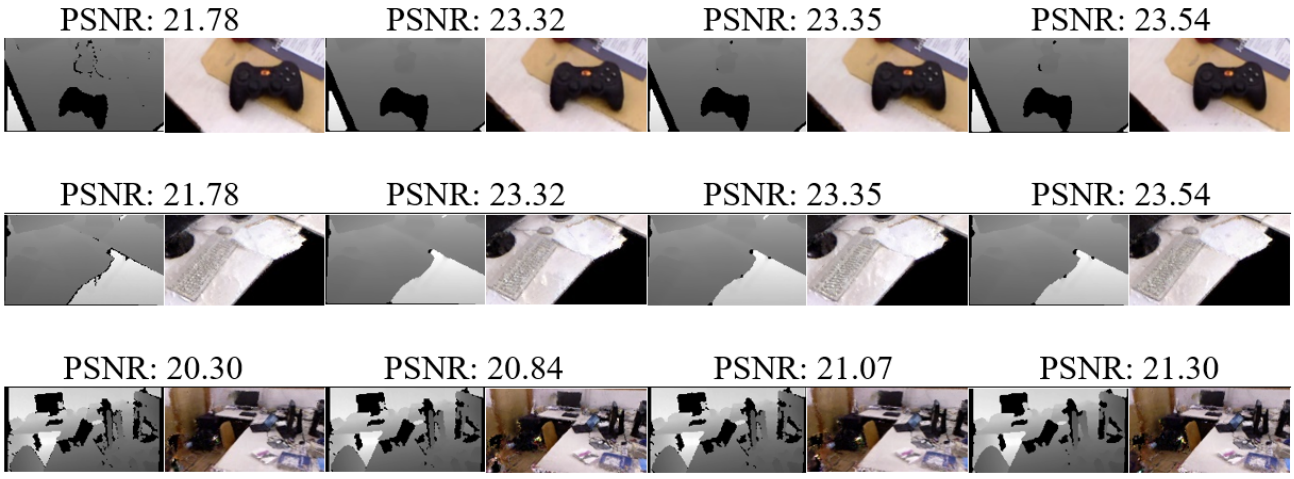


Figure 7: The Effect of Median Filter on Rendering Results

5. Conclusion

In this study, focusing on the problems in the quality of depth images in the TUM-RGBD dataset, the median filter method is used to improve the features of in-depth images, and the SplaTAM method, which uses Gaussian structure in scene representation, is used to improve 3D reconstruction performance. The TUM-RGBD, Replica and Scannet datasets are used to evaluate the performance of visual SLAM methods. Compared to Replica and Scannet datasets, which are high-quality datasets, the TUM-RGBD dataset contains lower-quality depth data with poor feature information. The original SplaTAM study [10] mentioned that the TUM-RGBD dataset consists of depth images with low image quality. Therefore, it was determined that the TUM-RGBD dataset is not preferred in studies representing visual quality. This study's contribution has demonstrated that the quality of depth images directly affects the performance of the Gaussian Splatting-based SplaTAM method. The improvements have significantly increased SplaTAM's mapping quality, thus highlighting the critical importance of depth image quality in SLAM systems.

In this study, the original depth images of two scenes in the TUM-RGBD dataset were processed with the SplaTAM method and PSNR, SSIM and LPIPS metric results were extracted. Then, a median filter is applied to the existing depth images to improve their properties. In the first scene representation, an 8.08% improvement was achieved compared with the original result, and in the second scene, a 4.69% improvement was achieved. These results show that the median filter effectively improves the weak features of in-depth images and its performance when applied to the Gaussian Splatting-based SplaTAM method.

Based on this study, in future work, the effects of depth data sets used in visual SLAM methods on 3D reconstruction performances can be examined more comprehensively using different filtering methods or deep learning-based techniques.

Moreover, since the SplaTAM method depends on depth data, generating the depth data can be improved. In this context, using deep learning-based models to generate metric depth maps from RGB images can obtain RGB and depth data using only a monochrome camera. Thus, the hardware requirements of SLAM systems can be reduced and integrated into wider application areas.

References

[1] H. Durrant-Whyte, D. Rye, and E. Nebot, "Localization of Autonomous Guided Vehicles," Robotics Research, pp. 613–625, 1996, doi: 10.1007/978-1-4471-1021-7_69.

[2] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robotics and Automation Magazine*, vol. 13, no. 2, pp. 99–108, Jun. 2006, doi: 10.1109/MRA.2006.1638022.

[3] R. C. Smith and P. Cheeseman, "On the Representation and Estimation of Spatial Uncertainty," *The international journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, Dec. 1986, doi: 10.1177/027836498600500404.

[4] H. Taheri and Z. C. Xia, "SLAM; definition and evolution," *Engineering Applications of Artificial Intelligence*, vol. 97, p. 104032, Jan. 2021, doi: 10.1016/J.ENGAPPAI.2020.104032.

[5] T. J. Chong, X. J. Tang, C. H. Leng, M. Yogeswaran, O. E. Ng, and Y. Z. Chong, "Sensor Technologies and Simultaneous Localization and Mapping (SLAM)," *Procedia Computer Science*, vol. 76, pp. 174–179, Jan. 2015, doi: 10.1016/J.PROCS.2015.12.336.

[6] W. Chen *et al.*, "An Overview on Visual SLAM: From Tradition to Semantic," *Remote Sensing 2022, Vol. 14, Page 3010*, vol. 14, no. 13, p. 3010, Jun. 2022, doi: 10.3390/RS14133010.

[7] A. R. Sahili *et al.*, "A Survey of Visual SLAM Methods," *IEEE Access*, vol. 11, pp. 139643–139677, 2023, doi: 10.1109/ACCESS.2023.3341489.

[8] A. Macario Barros, M. Michel, Y. Moline, G. Corre, and F. Carrel, "A Comprehensive Survey of Visual SLAM Algorithms," *Robotics 2022, Vol. 11, Page 24*, vol. 11, no. 1, p. 24, Feb. 2022, doi: 10.3390/ROBOTICS11010024.

[9] E. Sandström, Y. Li, L. van Gool, M. R. Oswald, E. Zürich, and K. Leuven, "Point-SLAM: Dense Neural Point Cloud-based SLAM." pp. 18433–18444, 2023.

[10] N. Keetha *et al.*, "SplaTAM: Splat Track & Map 3D Gaussians for Dense RGB-D SLAM." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 21357–21366, 2024.

[11] C. Yan *et al.*, "GS-SLAM: Dense Visual SLAM with 3D Gaussian Splatting." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 19595–19604, 2024.

[12] B. Kerbl, G. Kopanas, T. Leimkuehler, and G. Drettakis, "3D Gaussian Splatting for Real-Time Radiance Field Rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, p. 14, Aug. 2023, doi: 10.1145/3592433.

[13] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2320–2327, 2011, doi: 10.1109/ICCV.2011.6126513.

[14] R. A. Newcombe *et al.*, "KinectFusion: Real-time dense surface mapping and tracking," *2011 10th IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2011*, pp. 127–136, 2011, doi: 10.1109/ISMAR.2011.6092378.

[15] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "ElasticFusion: Real-time dense SLAM and light source estimation," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1697–1716, Sep. 2016, doi: 10.1177/0278364916669237.

[16] T. Schops, T. Sattler, and M. Pollefeys, "BAD SLAM: Bundle Adjusted Direct RGB-D SLAM." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 134–144, 2019.

[17] Z. Teed and J. Deng, "DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras," *Advances in Neural Information Processing Systems*, vol. 20, pp. 16558–16569, Aug. 2021.

[18] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "iMAP: Implicit Mapping and Positioning in Real-Time," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6209–6218, 2021, doi: 10.1109/ICCV48922.2021.00617.

[19] Z. Zhu *et al.*, "NICE-SLAM: Neural Implicit Scalable Encoding for SLAM," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2022-June, pp. 12776–12786, 2022, doi: 10.1109/CVPR52688.2022.01245.

[20] X. Yang, H. Li, H. Zhai, Y. Ming, Y. Liu, and G. Zhang, "Vox-Fusion: Dense Tracking and Mapping with Voxel-

based Neural Implicit Representation," *Proceedings - 2022 IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2022*, pp. 499–507, Oct. 2022, doi: 10.1109/ISMAR55827.2022.00066.

[21] H. Wang, J. Wang, and L. Agapito, "Co-SLAM: Joint Coordinate and Sparse Parametric Encodings for Neural Real-Time SLAM," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2023-June, pp. 13293–13302, Apr. 2023, doi: 10.1109/CVPR52729.2023.01277.

[22] H. Matsuki, R. Murai, P. H. J. Kelly, and A. J. Davison, "Gaussian Splatting SLAM." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18039–18048, 2024.

[23] C. Yan *et al.*, "GS-SLAM: Dense Visual SLAM with 3D Gaussian Splatting," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* pp. 19595–19604, 2024.

[24] J. Straub *et al.*, "The Replica Dataset: A Digital Replica of Indoor Spaces," Jun. 2019, Accessed: Jan. 24, 2025. [Online]. Available: https://arxiv.org/abs/1906.05797v1

[25] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Niessner, "ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5828–5839, 2017.

[26] C. Yeshwanth, Y.-C. Liu, M. Nießner, and A. Dai, "ScanNet++: A High-Fidelity Dataset of 3D Indoor Scenes," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12–22, 2023.

[27] J. Sturm, W. Burgard, and D. Cremers, "Evaluating Egomotion and Structure-from-Motion Approaches Using the TUM RGB-D Benchmark," *Proc. of the Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RJS International Conference on Intelligent Robot Systems (IROS)*, vol. 13, 2012.

[28] Q. Yang, R. Yang, J. Davis, and D. Nistér, "Spatial-depth super resolution for range images," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007, doi: 10.1109/CVPR.2007.383211.

[29] G. Deng and L. W. Cahill, "Adaptive Gaussian filter for noise reduction and edge detection," *IEEE Nuclear Science Symposium & Medical Imaging Conference*, no. pt 3, pp. 1615–1619, 1994, doi: 10.1109/NSSMIC.1993.373563.

[30] I. Pitas and A. N. Venetsanopoulos, "Nonlinear Mean Filters in Image Processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 3, pp. 573–584, Jun. 1986, doi: 10.1109/TASSP.1986.1164857.

[31] M. Kazubek, "Wavelet domain image denoising by thresholding and Wiener filtering," *IEEE Signal Processing Letters*, vol. 10, no. 11, pp. 324–326, Nov. 2003, doi: 10.1109/LSP.2003.818225.

[32] P. Jain and V. Tyagi, "A survey of edge-preserving image denoising methods," *Information Systems Frontiers*, vol. 18, no. 1, pp. 159–170, Feb. 2016, doi: 10.1007/S10796-014-9527-0/TABLES/1.

[33] T. S. Huang, G. J. Yang, and G. Y. Tang, "A Fast Two-Dimensional Median Filtering Algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 1, pp. 13–18, 1979, doi: 10.1109/TASSP.1979.1163188.

[34] A. Ravishankar, S. Anusha, H. K. Akshatha, A. Raj, S. Jahnavi, and J. Madhura, "A survey on noise reduction techniques in medical images," *Proceedings of the International Conference on Electronics, Communication and Aerospace Technology, ICECA 2017*, vol. 2017-January, pp. 385–389, 2017, doi: 10.1109/ICECA.2017.8203711.

[35] F. Artuğer and F. Özkaynak, "Görüntü Sıkıştırma Algoritmalarının Performans Analizi İçin Değerlendirme Rehberi," *International Journal of Pure and Applied Sciences*, vol. 8, no. 1, pp. 102–110, Jun. 2022, doi: 10.29132/IJPAS.1012013.

[36] S. Ghazanfari, S. Garg, P. Krishnamurthy, F. Khorrami, and A. Araujo, "R-LPIPS: An Adversarially Robust Perceptual Similarity Metric," Jul. 2023.

Authors' Contribution

Study conceptualization, design: C. Z., A. F. K. Supervision: A. F. K. Data collection and analysis: C. Z. Literature review: C. Z., A. F. K. Manuscript writing: C. Z. Final review: A.F.K.

Conflict of Interest Notice

The authors declare that there is no conflict of interest regarding the publication of this paper.

Artificial Intelligence Statement

No artificial intelligence tools were used while writing this article.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
http://saucis.sakarya.edu.tr/
e-ISSN: 2636-8129
Publisher: Sakarya University
Vol. 8, No. 2, 273-285, 2025
DOI: 10.35377/saucis..1589506
RESEARCH ARTICLE

Assessing the Role of Software in Sustainability: A Survey of Industry Practices and Research Trends

Enes Bajrami[1*]
[1] Ss. Cyril and Methodius University in Skopje, Faculty of Computer Science and Engineering, Ruger Boskovik 16, 1000 Skopje, Republic of North Macedonia

Corresponding author:
Enes Bajrami, Ss. Cyril and Methodius University
enes.bajrami@students.finki.ukim.mk

Article History:
Received: 21.11.2024
Revised: 09.04.2025
Accepted: 09.04.2025
Published Online: 17.06.2025

ABSTRACT
The ever-increasing demand for complex software applications has turned the entire ICT resource and energy consumption into a significant environmental concern. Several research studies have concentrated on making hardware sustainable; however, the environmental aspects of software remain underexplored. This paper discusses the contribution of software to Green Computing, with a special emphasis on energy efficiency throughout the software development life cycle. The study has pointed out the guiding principles of Sustainable Software Engineering: the efficiency of energy and resources, sustainable lifecycle management, and design centered on the users. The analysis also reveals that environmental effects, such as carbon footprint and energy consumption, call for targeted software component improvements. The study also examines the hurdles in implementing green software engineering, such as legacy system challenges, regulatory issues, and economic viability. This research integrates insights from climate science, hardware optimization, and software engineering to contribute to developing eco-friendly software systems, thus charting future directions for sustainability in this field.
Keywords: Green Computing, Sustainable Software Engineering, Energy Efficiency, Resource Optimization

1. Introduction

The rising demand for increasingly complex software applications has led to a significant negative environmental impact from Information and Communication Technology (ICT), primarily due to its escalating consumption of resources and energy [1] [2]. Approximately 97% of climate scientists concur that the observed global warming trends over the past century are most likely attributable to human activities [3]. The impact of Information and Communication Technology (ICT) on sustainable development, particularly software, has become a prominent focus in Green Computing. Sustainable development involves utilizing resources to meet human needs while considering the ecological, economic, and societal consequences [4] [5]. Although recent efforts in ICT have sought to develop environmentally efficient solutions, it remains uncertain whether ICT's energy and resource savings will outweigh its overall resource consumption [6] [7]. A considerable body of research on Green ICT has primarily concentrated on environmental sustainability concerning computer hardware. However, addressing the energy consumption issues associated with software is crucial for advancing green computing. Software features contribute to CO_2 emissions just as much as hardware components [8] [9]. Software exerts an indirect environmental impact by managing and operating the underlying hardware. Certain software solutions can optimize resource utilization, while others are designed to be sustainable enough to reduce the need for additional hardware following updates [10]. Unfortunately, a notable lack of models and research focused on software and software development processes exists. Recently, significant efforts have been made to develop green software. Some initiatives aim to create sustainable software, while others design software development processes that guide stakeholders in producing environmentally friendly software products [11]. Additional efforts focus on developing tools that measure the environmental impact of software and the energy efficiency of application development environments [12]. There is also an emphasis on enhancing operating systems to better manage applications' power consumption [13]. The software dimensions of Green IT have not been extensively explored due to its intangible nature and its indirect impact on the environment [14]. However, researchers are increasingly recognizing software's direct and indirect environmental impacts. Energy efficiency and sustainability throughout the software life cycle are essential, though these factors have traditionally been overlooked in conventional software development processes [15]. The main challenges in integrating sustainability into software development include the unclear extent of software's contribution to overall hardware energy consumption, the uncertain role of software engineering in promoting sustainability, and the lack of a clear conceptual foundation [16]. Artificial intelligence (AI) and machine learning (ML) have gained attention for their potential to enhance sustainability efforts, particularly in optimizing energy consumption and improving software efficiency [17]. However, the increasing complexity of AI models, including large language models (LLMs), has

Cite as: E. Bajrami. (2025). Assessing the role of software in sustainability: A survey of industry practices and research trends. *Sakarya University Journal of Computer and Information Sciences*. vol. 8, no. 2, pp. 273-285, 2025 doi: 10.35377/saucis...1589506

also raised concerns about their energy consumption and carbon footprint [18]. While AI-powered solutions hold promise for green software development, this study focuses on existing software tools used in industrial settings for sustainability. Future work could explore how AI-driven optimization techniques contribute to improving energy efficiency in industrial applications [19] [20].

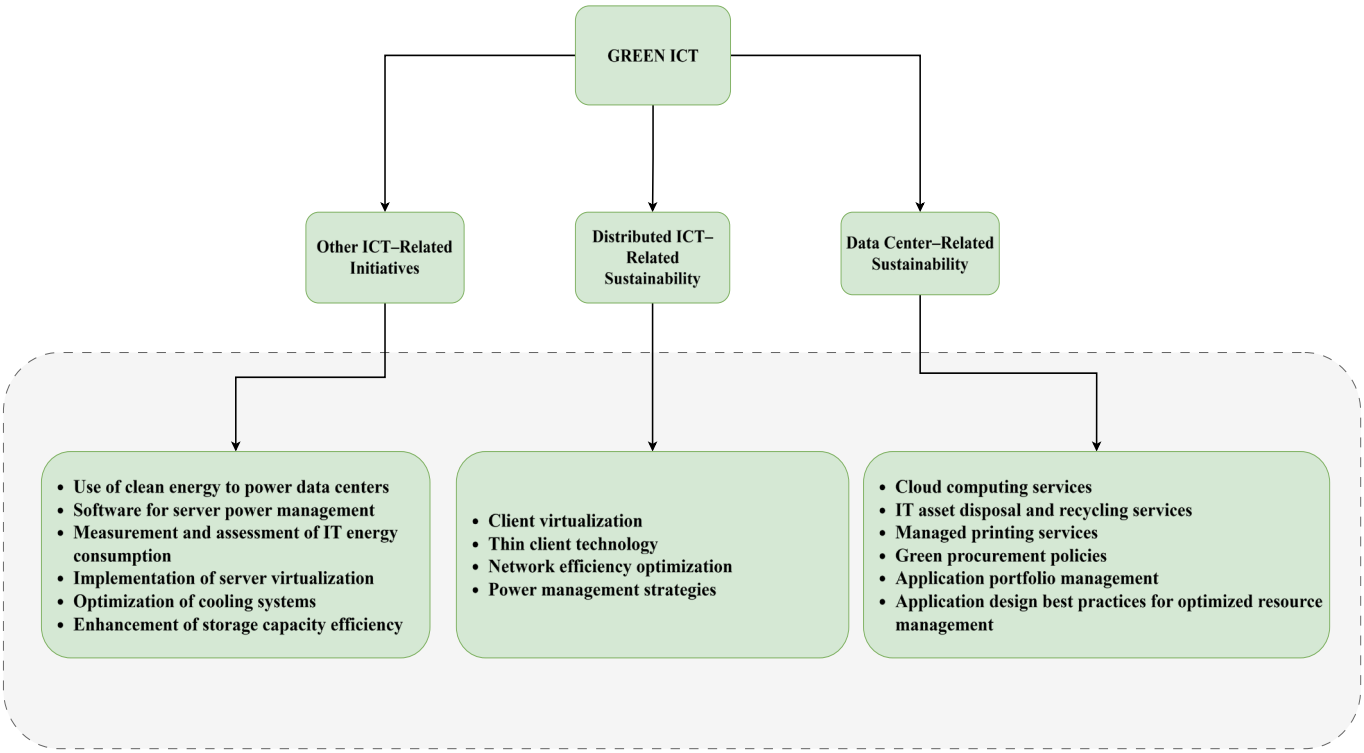


Figure 1: Taxonomy of Green ICT

Figure 1 presents a structured overview of Green ICT practices, categorized into three core domains: Data Center–Related Sustainability, Distributed ICT–Related Sustainability, and Other ICT–Related Initiatives. The author developed this taxonomy to conceptually organize key sustainability actions across different layers of ICT systems. In the first category, Data Center–Related Sustainability includes using clean energy, server power management, virtualization, improved cooling mechanisms, and enhanced storage capacity efficiency, all aimed at reducing the environmental impact of large-scale computing infrastructures. Distributed Sustainability focuses on client-side practices, such as client virtualization, thin client technology, network efficiency optimization, and power management strategies, which contribute to minimizing energy consumption at the end-user level. The final category, Other ICT–Related Initiatives, addresses broader organizational measures such as cloud computing services, IT asset recycling, managed printing, green procurement, application portfolio management, and sustainable application design. These elements provide a comprehensive foundation for understanding how sustainability is integrated into ICT environments.

2. Literature Review

Several studies have been carried out to collect, analyze, and assess the evidence available in Green Software Engineering. These investigations focus on identifying effective strategies and approaches that enhance software systems' sustainability, underscoring software's critical role in reducing environmental impact. In [21], the author explores the principles and practices of green software engineering, emphasizing its environmental impact and identifying best practices in the field. The study specifically focuses on energy efficiency, resource optimization, and eco-friendly development methodologies, all aimed at reducing the carbon footprint of software engineering. Additionally, the paper addresses the challenges and regulatory constraints associated with implementing sustainable practices. The research underscores the importance of a collective commitment to energy efficiency and sustainability within the software engineering community. In [22], a systematic literature review (SLR) was conducted to map the state-of-the-art in Sustainable Software Engineering (SE), focusing on existing models, guidelines, practices, and related proposals. However, this review was limited to 36 works published until 2010 and addressed only three research questions. The study explored the most frequently cited guidelines and models in Sustainable SE, tracked the evolution of interest in the field, and identified key authors and venues contributing to this area of research. In [23], a review study was conducted on green computing, focusing on five key areas: software

engineering, cloud computing, mobile computing, data centers, and the educational sector. The study performed a systematic literature review for each area, detailing current research trends and limitations. However, it lacked discussion on using datasets, their characteristics, testing mechanisms, and their contributions to various perspectives on current technology applications within these fields. Since green computing is closely linked with multiple domains, exploring these associations is crucial for developing environmentally friendly modern computing systems. In [24], a comprehensive review was conducted on energy management techniques to achieve green IoT. The study presents the current challenges in green IoT related to energy consumption and examines various approaches utilized by different studies for energy management. It offers a thorough overview of recent energy management systems within the IoT ecosystem. It outlines current research trends and future perspectives for energy management in the context of green IoT. Reviewing these implementation studies provides valuable insights, identifies existing challenges, and suggests potential directions for future research, not only within green computing but also across other related fields. In [25], the authors discuss the growing significance of software in the twenty-first century, emphasizing its indirect impact on hardware energy consumption and carbon emissions. While previous studies have predominantly focused on models and tools for measuring power consumption and energy efficiency from a hardware perspective, less attention has been given to the role of software development in energy optimization. The study highlights that energy consumption can be reduced by implementing green software practices throughout all phases of the development lifecycle. However, existing green software process models primarily concentrate on environmental and economic aspects without adequately integrating waste management in the development phase. To address this gap, a qualitative study was conducted involving interviews with eight informants from Malaysia's public and private sectors. The study aimed to (i) examine the current industry practices in green software processes, (ii) identify waste elements in software development, and (iii) determine key green factors influencing the software process. Thematic analysis was conducted using Atlas. ti 8 revealed three key themes: best practices in software processes, nine categories of software waste (e.g., building the wrong feature, rework, unnecessary complexity, cognitive overload, psychological distress, waiting, knowledge loss, ineffective communication, and delays), and six green factors (resources, people, organizational aspects, technical aspects, environmental concerns, and technology). The findings indicate that integrating best practices, green methodologies, and software technologies at every stage of development is crucial for achieving a sustainable and environmentally friendly software process. The study underscores the role of advancing computing technologies in maintaining a continuously updated and green software development framework. In [26], the authors explore the emerging research area of sustainability in Software Engineering, highlighting that while sustainability has been widely discussed in academia, it remains underrepresented in the software industry. The study emphasizes that incorporating sustainability into software design and development can provide significant societal benefits, but this requires increased awareness and knowledge among software professionals. To address this gap, the research examines sustainability knowledge, its perceived importance, and industry support from the perspective of South Asian software professionals. The study investigates key questions such as how professional software developers perceive sustainability, how the software industry identifies sustainability requirements, and how developers incorporate sustainability parameters during software development. A survey was conducted among 221 industry practitioners working on software projects in banking, finance, and management applications. The results indicate that while 91% of practitioners recognize the importance of sustainability, a substantial knowledge gap exists in its practical application. Notably, 48% of professionals misinterpret "Green software" as "sustainable software." Moreover, the technical aspect of sustainability is regarded as the most important factor by 67% of professionals and 77% of companies. A critical finding of the study is that 92% of software practitioners cannot identify sustainability requirements for software applications. The study contributes by proposing sustainability guidelines for specific software applications and a catalog to assist in identifying sustainability requirements. The findings provide an initial perspective on how sustainability is understood and addressed within the South Asian software industry. In [27], the authors highlight the critical role of the electronics sector in modern industry, emphasizing its contribution to clean technology, dry processes, and efficient design, which align with Industry 4.0 and sustainability principles. However, the rapid obsolescence of electronic devices has led to a significant increase in electronic waste. To mitigate this issue, the study proposes a novel edge computing structure, the AIFC, which operates independently of specific systems and leverages existing computing infrastructures. The AIFC is built on an enterprise service bus (ESB) and implemented using decentralized microservices, reducing reliance on conventional cloud computing models. The study adopts an action research approach involving collaboration between researchers and industry practitioners and tests the proposed structure in six different scenarios. These scenarios simulate small and medium-sized enterprise (SME) environments and encompass various stages, including proof of concept, prototyping, minimum viable product, scalability, and a roadmap for implementation. The developed microservices facilitate data filtering, processing, storage, querying, and sensor data acquisition while maintaining low latency and, in some cases, improving performance compared to traditional cloud-based architectures. Furthermore, the findings demonstrate that the approach eliminates the need for hardware or communication structure upgrades—key contributors to electronic waste and rapid obsolescence. Following the AIFC development process, the study presents a sustainable roadmap supporting Industry 4.0 initiatives and SME digital transformation efforts.

3. Methodology

3.1 Methodology of Research

This study employs a mixed-method approach, combining a literature-based analysis with a questionnaire-based survey to explore green software engineering practices and their adoption. While this study does not follow a Systematic Literature Review (SLR) methodology, a structured approach was applied to ensure a comprehensive and relevant literature search. The literature review was conducted by searching for peer-reviewed journal articles, industry reports, and case studies in IEEE Xplore, ACM Digital Library, SpringerLink, ScienceDirect, and Google Scholar databases. Search terms included "Green Software," "Sustainable Computing," "Software Energy Efficiency," and "Eco-Friendly Software Engineering." Studies were selected based on their relevance to sustainability in software engineering, with priority given to recent publications from 2015 to 2024. This review provided the foundation for understanding key themes, methodologies, and trends in the field. The survey-based study was designed to complement the findings of the literature by assessing how green software practices are adopted in industrial settings. The structured questionnaire included 15 Yes/No questions focusing on energy optimization, predictive maintenance, resource management, and sustainability metrics. A pilot test was conducted with five factory managers to ensure clarity and relevance.

3.2 Data Collection

The study incorporates both qualitative and quantitative data sources. Qualitative data was gathered from peer-reviewed journals, conference proceedings, and industry reports better to understand green software engineering practices and their practical implications. The questionnaire-based survey was conducted to gather quantitative data on green software adoption across different industrial sectors. The survey targeted six types of factories in the Polog[1] region, including refrigerator, railing, door and window, candle, oil, and outdoor tile factories, chosen for their operational diversity. Since I live in the Polog region, this location was selected due to its accessibility. It allowed direct communication with factory managers and ensured a higher response rate and more reliable data collection. The survey responses were collected from 6 factories using stratified random sampling to ensure a balanced representation across industries. Data was collected through in-person interviews and online survey forms, ensuring broad participation. Python was used to analyze the responses and to process and visualize the data. Figures 3–6 were generated using Python-based tools such as Matplotlib and Seaborn, providing graphical insights into green software adoption trends, industry-wise variations, and key challenges in sustainability implementation. The combination of literature analysis and survey responses provides a holistic view of sustainability practices in software engineering. This mixed-method approach ensures that the study captures theoretical insights and practical realities, highlighting the factors influencing the adoption of green practices and offering actionable recommendations for future research and policy development.

4. The Principles of Sustainable Software Engineering

Sustainable Software Engineering is a developing field that blends climate science with software, hardware, energy markets, and data center design [28]. It encompasses a fundamental set of skills required to create, develop, and operate software applications in an environmentally responsible manner. A primary principle in green software engineering is energy efficiency [29]. This principle centers on optimizing software to minimize energy consumption during its operation. Electricity usage is decreased by making software more energy-efficient, and hardware longevity is enhanced due to reduced strain [30] [31]. Achieving energy efficiency involves employing techniques such as optimizing code, utilizing efficient data structures, and implementing algorithms that lower computational demands. Resource efficiency is another crucial principle, focusing on efficiently using computing resources like CPU and memory. Resource-efficient software operates effectively within the limitations of the hardware it runs on, avoiding unnecessary resource use [32]. This approach reduces the environmental impact and reduces companies' operational expenses. Load balancing, virtualization, and efficient resource management are key to enhancing resource efficiency [33]. Managing the software lifecycle sustainably is also a key principle. This entails considering the environmental impact of software from its design and development phases to deployment, maintenance, and eventual decommissioning. Sustainable lifecycle management advocates using renewable energy in data centers, adopting energy-efficient hardware, and recycling electronic waste [34]. Moreover, it promotes the creation of software that can be easily maintained and updated, prolonging its lifespan and lessening the need for frequent replacements. The principle of scalability is significant in ensuring that software can grow to handle more workload without a corresponding increase in resource use. Scalable software can efficiently manage more users or data without substantially increasing energy use [35]. Cloud computing and distributed systems are often utilized to achieve scalable, sustainable software. User-centered design is another important aspect. Software should be designed to focus on essential functionalities, ensuring it meets user needs without including superfluous features that might increase energy consumption [36]. By creating simple user interfaces and focusing on necessary features, developers can produce energy-efficient and user-friendly software, contributing to the software's overall sustainability [37]. Finally, sustainability metrics and reporting are vital for green software engineering. Developers and organizations must measure and report on the environmental impacts of their

[1] The Polog Statistical Region is one of eight statistical regions of the Republic of North Macedonia. Polog, located in the northwestern part of the country, borders Albania and Kosovo. Internally, it borders the Southwestern and Skopje statistical regions.

software [38]. Standard metrics like carbon footprint and energy usage help organizations assess and improve their software's sustainability. Regular reporting and transparency in these practices can bolster a company's reputation and commitment to sustainability [39].

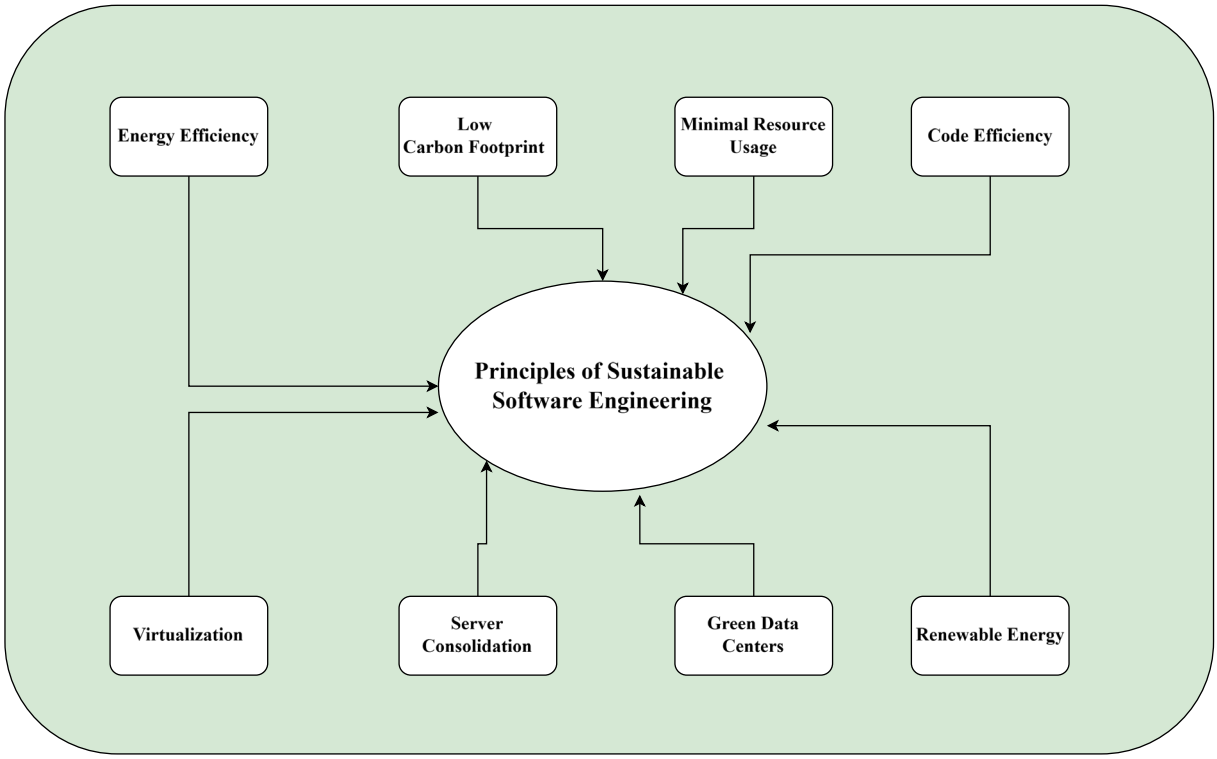


Figure 2: Green Software Engineering Principles

Figure 2 illustrates a visual summary of the fundamental principles of sustainable software engineering inspired by the work of Sivkumar Mishra and Namita Dehury [40] and redrawn by the author to align with the focus of this study. The diagram presents eight essential principles that guide the development of environmentally responsible software systems. These include energy efficiency, which involves reducing power consumption during software execution; low carbon footprint, which refers to minimizing greenhouse gas emissions linked to computing infrastructure; and minimal resource usage, which focuses on optimizing the consumption of processing power, memory, and storage. Code efficiency highlights the importance of writing optimized and maintainable software. Virtualization promotes scalability and resource sharing while reducing dependency on physical hardware. Server consolidation aims to streamline workloads onto fewer servers to enhance resource utilization. Green data centers support the transition to sustainable infrastructure through energy-efficient systems and clean power. Finally, renewable energy promotes the shift toward powering software systems using sustainable energy sources. These principles represent a comprehensive foundation for integrating sustainability into software engineering practices.

5. Evaluation of Environmental Effects

Environmental Impact Assessment (EIA) evaluates the potential effects of a project or policy on the environment. It aims to identify, predict, and mitigate adverse impacts, ensuring informed decision-making and promoting sustainable development by integrating environmental considerations into planning processes [41].

5.1 Measuring Carbon Footprints

This involves assessing the total carbon emissions produced during the Software Development Life Cycle [21]. Doing so helps evaluate the software's environmental impact and identify strategies to mitigate carbon footprints. Carbon footprints are typically quantified in CO2 equivalents, determined by energy usage, hardware, data centers, and user behaviors [42].

5.2 Analyzing Energy Consumption in Software Systems

This is the most crucial phase of the software development life cycle. Detailed observation and monitoring are conducted to reveal energy usage patterns and traits [21]. This process identifies which software components and processes consume the most energy, highlighting areas for optimization. Developers can then focus on creating more efficient algorithms, promoting environmentally friendly software development [43].

5.3 Identifying Hotspots for Improvement

The next step involves pinpointing the most energy-consuming components. After identifying these, we can focus on hotspots where environmental improvements will have the greatest impact. Hotspots vary depending on power consumption and the database resources they use. Once identified, developers can more effectively target these areas for improvement [21].

6. Methods and Utilities

The Methods and Utilities chapter delves into the foundational resources and strategies for sustainable software development. It begins with an overview of the key concepts and examines energy-efficient programming languages and frameworks. The chapter also explores sustainability-focused software development methodologies, comprehensively understanding the tools and techniques supporting eco-friendly software engineering practices.

6.1 Overview

Green software engineering tools are designed to assist developers in creating more energy-efficient and environmentally friendly software. These tools are specifically developed to measure, analyze, and optimize the environmental impacts of software. They enable efficient use of resources and come in various forms, including energy profiling tools, sustainability assessment platforms, resource optimization software, and green code analyzers [44]. A notable example is Google's commitment to sustainable software development, where they utilize machine learning for optimizing data center cooling and resources, leading to significant energy savings and reduced greenhouse gas emissions. Similarly, Meta's Open Compute Project (OCP) has achieved considerable energy efficiency and lowered its carbon footprint in data center operations [45].

6.2 Energy-Efficient Programming Languages and Framework

These frameworks are created to reduce energy and resource consumption, aiding developers in producing software that requires less energy [21]. Key features of energy-efficient languages include optimized use of resources, minimal runtime overhead, avoidance of unnecessary computations, efficient memory management, and simplified algorithmic complexity [46].

6.3 Sustainability-Focused Software Development Methodologies

A security-focused software development approach integrates environmental and optimization considerations, ensuring that software meets both functional requirements and environmental standards [21]. It encompasses the entire software lifecycle and emphasizes eco-design to reduce waste and energy use. Microsoft and Salesforce exemplify this methodology, with both companies achieving significant energy savings and reduced carbon emissions by prioritizing renewable energy, efficient coding, and sustainable data center practices [47].

7. Barriers and Challenges

7.1 Challenges in Implementing Green Software Engineering

The primary challenge in adopting green software engineering lies in the reliance on legacy systems, which are often difficult to adapt to eco-friendly practices. Additionally, successful implementation requires significant awareness and training within organizations [21].

7.2 Regulatory and Policy Challenges

Environmental regulations vary by region and impact software development by addressing issues like energy consumption, e-waste management, and emissions. Compliance with these regulations is essential for green software engineering [48]. This includes adhering to standards for energy efficiency, participating in eco-labeling programs, and aligning with sustainability initiatives. Furthermore, regulations may mandate using renewable energy and proper e-waste disposal, influencing how organizations manage data centers and infrastructure. Compliance costs must be considered, as well as potential tax incentives for adopting sustainable practices [49].

7.3 Economic Factors and Cost Efficiency

Economic viability is crucial for green software engineering practices. Organizations must evaluate the total cost of ownership, including development, operational expenses, and potential savings from energy efficiency. Optimizing resource utilization can reduce costs, and while compliance with environmental regulations may incur costs, non-compliance can result in fines. Transitioning to renewable energy may require upfront investment but can offer long-term savings [50]. Evaluating the return on investment (ROI) involves balancing these costs with benefits such as energy savings and enhanced brand reputation. Economic assessments should also factor in risks like fluctuating energy prices and the positive impact of sustainability on employee productivity. Achieving both sustainability and cost-effectiveness is essential for the long-term success of green software engineering initiatives [21] [51].

8. Questionnaire-Based Assessment of Sustainability Practices

The following section presents the findings of a comprehensive questionnaire distributed to six distinct types of factories in the Polog region. The survey aimed to evaluate the adoption of green software engineering practices and sustainability-focused solutions within these factories. Spanning 15 critical questions, the questionnaire explored various dimensions of sustainable software usage, such as energy optimization, predictive maintenance, resource management, and lifecycle analysis. The responses were analyzed and visualized using multiple charts to provide clear insights into adoption trends, factory-specific patterns, and overall engagement with sustainability practices. These visualizations highlight the strengths and gaps in the current implementation of green technologies, serving as a basis for further discussions and recommendations. This section includes four charts that collectively summarize the results, offering a mix of aggregated and detailed perspectives on the data. Each chart sheds light on a specific survey aspect, enabling a deeper understanding of how factories integrate (or fail to integrate) sustainable software practices in their operations. Figures 3, 4, 5, and 6 were independently developed by the author based on the quantitative analysis conducted in this study and serve as original visual representations of the survey findings.

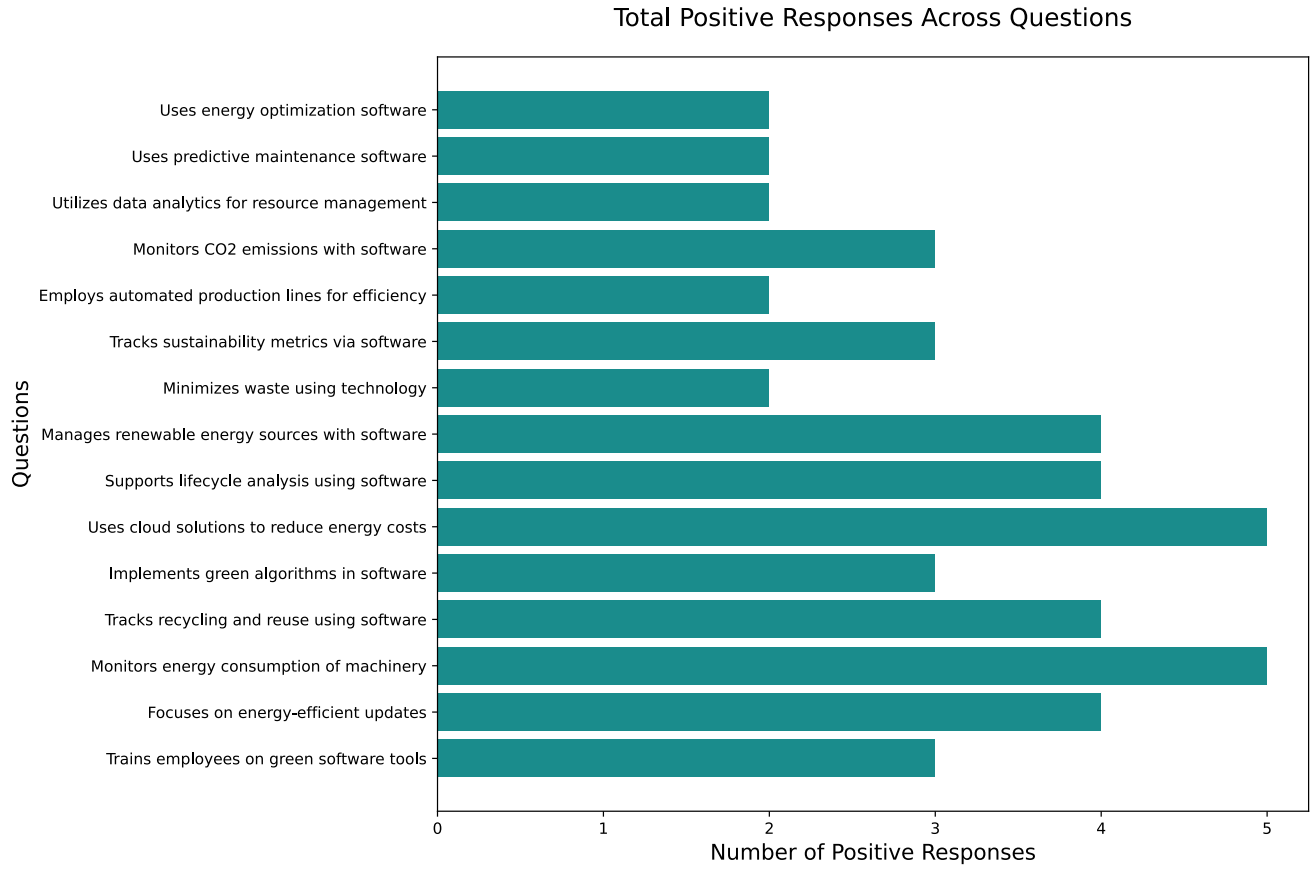


Figure 3: Total Positive Responses Across Questions

This chart visually represents the total number of positive responses received for each question across all factories. The questions are displayed along the y-axis, while the x-axis quantifies the number of positive responses. The chart highlights which areas of green software engineering practices (e.g., energy optimization or CO2 monitoring) are most commonly adopted by the factories overall.

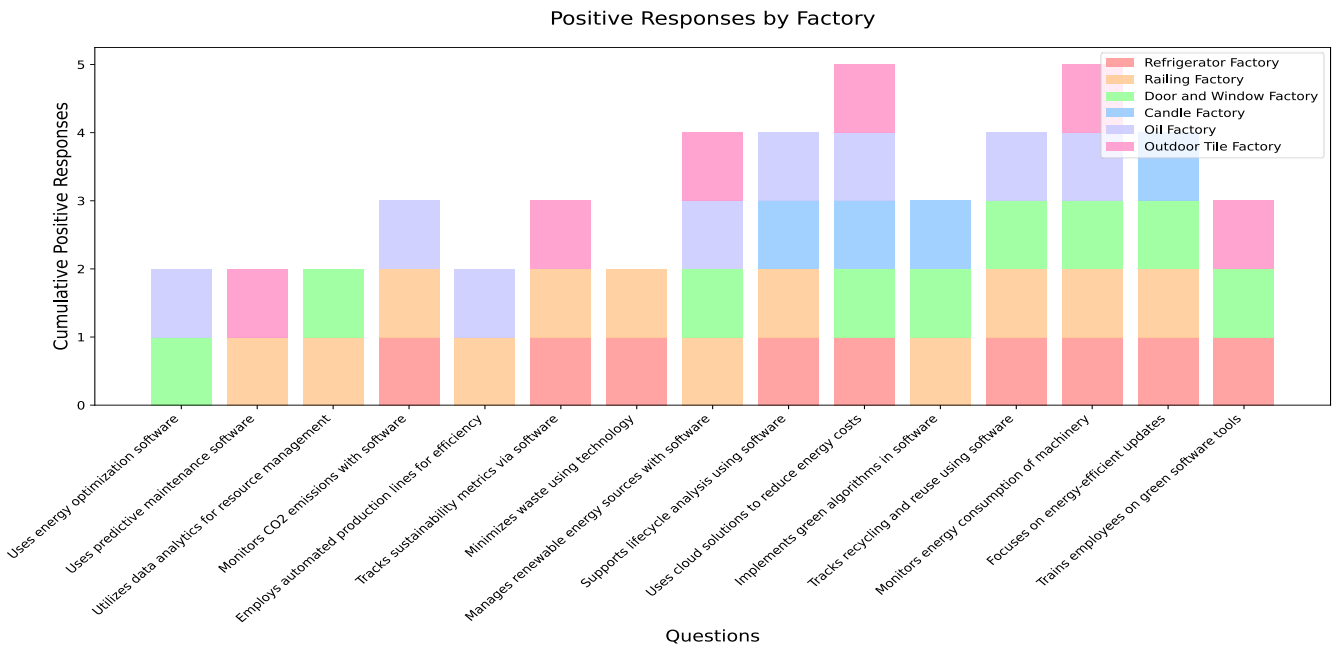


Figure 4: Positive Responses by Factory

This chart breaks down positive responses for each question by factory type. Each bar is segmented into colored sections, where each section represents the contribution of a specific factory type (e.g., Refrigerator Factory, Railing Factory). The chart allows a comparison of how different factory types adopt green software practices across various aspects of sustainability.

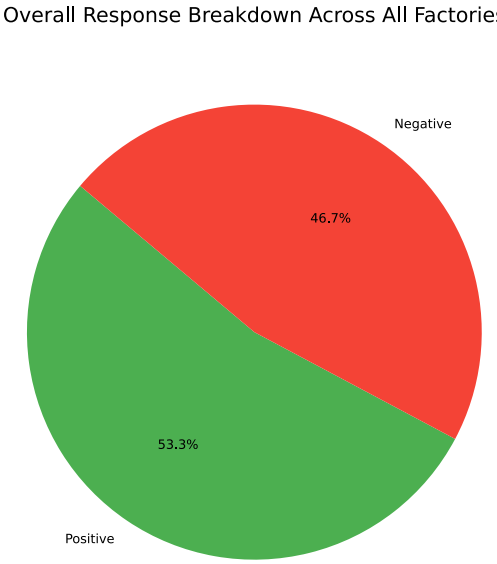


Figure 5: Overall Response Breakdown Across All Factories

This consolidated pie chart combines all responses across all factories and questions into positive and negative categories. The proportions of the two sections reveal the overall state of adoption of sustainability. Positive responses (green) reflect areas where factories have implemented green software engineering practices, while negative responses (red) indicate a lack of adoption.

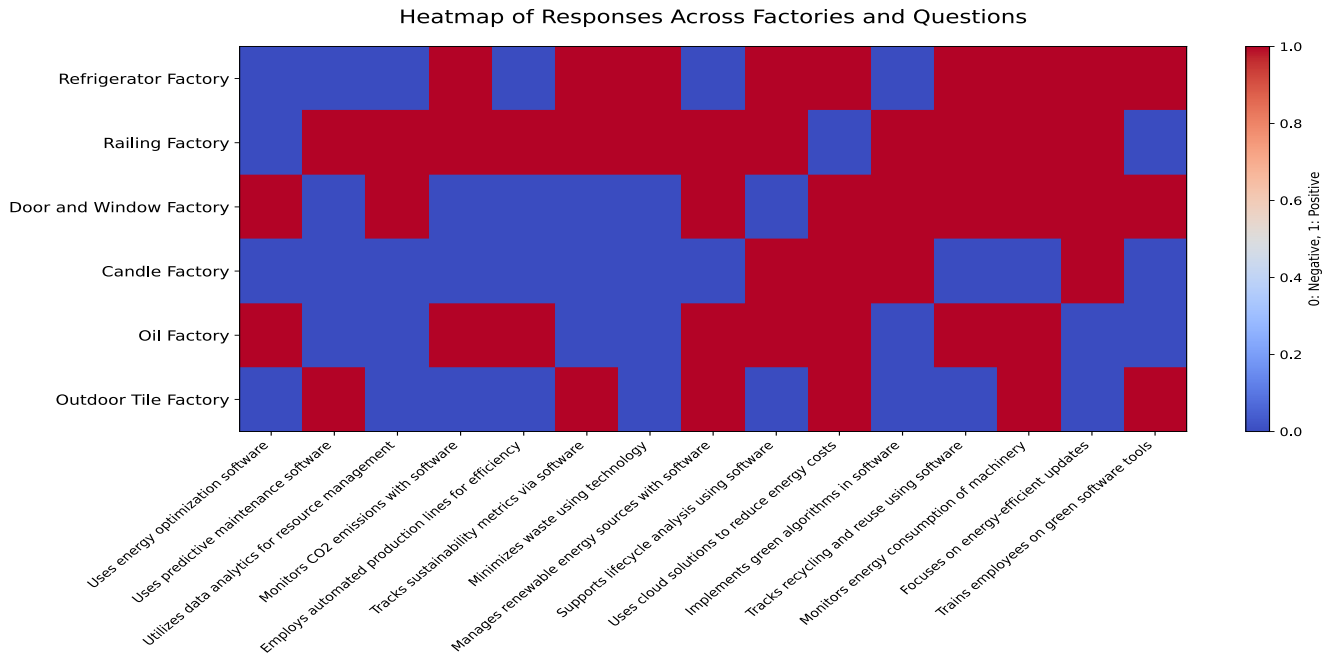


Figure 6: Responses Across Factories and Questions

This chart visualizes the response dataset in a heatmap format, where each cell represents a factory's response to a specific question. Darker shades represent negative responses (0), while lighter shades represent positive responses (1). Rows correspond to factory types, and columns correspond to questions, allowing readers to identify patterns or trends in the data at a glance.

9. Discussion

The study's findings demonstrate substantial deficiencies in implementing green software engineering methods among the examined manufacturers, indicating a widespread absence of sophisticated sustainability initiatives. Although fundamental techniques like predictive maintenance and resource management exhibited somewhat greater adoption rates, sophisticated technology such as CO2 monitoring, energy-efficient upgrades, and green algorithms were predominantly overlooked. Refrigerator, Door, and Window manufacturers showed somewhat superior engagement with specific green practices, possibly attributable to greater operational complexity and the necessity for efficiency. In contrast, Candle and Railing companies persistently fell behind in adopting sustainable software solutions. The findings highlight significant obstacles hindering advancement, such as insufficient awareness and training regarding green technologies, economic limitations that deter smaller factories from investing in these practices, and ineffective regulatory frameworks that do not enforce sustainability standards. These issues underscore the necessity for specific initiatives, including government subsidies, accessible training programs, and the creation of affordable, user-friendly solutions designed for small and medium enterprises (SMEs). The results underscore the necessity of cooperative initiatives among policymakers, industry stakeholders, and software developers to address existing deficiencies and facilitate the extensive implementation of sustainable software practices. The region may progress towards a more environmentally sustainable industrial framework by overcoming these obstacles while improving operational efficiency and cost-effectiveness. A significant finding from the survey indicates that factories employ various bespoke Enterprise Resource Planning (ERP) systems to facilitate their sustainability initiatives; however, these software solutions are tailored to individual factories, created either internally or by third-party vendors, and frequently do not possess a commercial designation. Although tailored to specific needs, these ERP-based systems possess shared features, including resource planning, inventory management, and production optimization, and display analogous user interfaces (UI) across many factories. These ERP solutions function as the principal digital framework for operational management, yet they are constrained in their ability to incorporate sophisticated green software functionalities. Moreover, companies utilize predictive maintenance technologies to enhance machinery performance and minimize energy waste, alongside monitoring systems to assess electricity consumption and carbon footprint. Nevertheless, none of the presently utilized software solutions are driven by artificial intelligence. Numerous industry participants indicated a desire to implement AI-based solutions imminently, especially for optimizing energy efficiency and automating sustainability reporting. This indicates that although custom ERP-based sustainability software is prevalent, there is an increasing push toward AI-driven optimization solutions to improve sustainable operations further. The findings suggest that current sustainability software utilization is operational yet stagnant, as most manufacturers employ rudimentary ERP and monitoring systems devoid of AI-driven analytics or automation capabilities. In light of the growing emphasis on energy efficiency and legal changes favoring environmentally sustainable industrial practices, implementing AI-enhanced sustainability software could represent a pivotal advancement for these sectors. Future research and development should

incorporate AI-driven optimization features into current ERP systems, facilitating the automation of sustainability reporting, improving energy efficiency, and optimizing resource utilization at a more sophisticated level.

10. Conclusion and Future Work

10.1 Conclusion

This study underscores the critical importance of incorporating sustainability into software engineering practices, highlighting that software's environmental impact can be as significant as that of hardware. This research provides a comprehensive understanding of green software engineering principles, adoption trends, and barriers by analyzing literature and real-world survey data. The findings reveal that while some basic green practices, such as predictive maintenance and energy optimization, show moderate adoption, advanced sustainability practices, including CO2 monitoring and green algorithms, remain largely neglected across industries. The questionnaire data provided insights into specific factory types, demonstrating variations in green software adoption. For example, Refrigerator and Door and Window factories displayed higher adoption rates than Candle and Railing factories, which lagged significantly. These results emphasize industries' critical challenges, including economic constraints, lack of awareness, and insufficient regulatory enforcement. Achieving sustainability in software engineering requires technical innovation and collaborative efforts from policymakers, industry stakeholders, and the software community to prioritize eco-friendly practices and develop actionable frameworks. This research highlights the need for a multi-faceted approach that includes education, accessible technologies, and supportive policies to drive the adoption of green practices across industries.

10.2 Future Work

Building on the insights gained from this study, future research should aim to bridge the gaps identified in the literature and the survey data. The next research phase will focus on developing specific research questions (RQs) that address the challenges and opportunities associated with sustainable software engineering. A systematic literature review will explore innovative methodologies and emerging technologies that enhance the sustainability of software systems. In particular, future studies should investigate the role of artificial intelligence (AI) and machine learning (ML) in optimizing energy and resource efficiency. These technologies hold immense potential for automating energy management, predicting resource consumption patterns, and improving the environmental impact of software systems. Additionally, longitudinal studies across diverse industry sectors could provide deeper insights into the evolving adoption of green software practices. Exploring the integration of green engineering standards into software development lifecycles and assessing the effectiveness of government incentives could further enhance our understanding of how to overcome barriers to adoption. Finally, creating open-source sustainability tools and frameworks tailored for small and medium enterprises (SMEs) could empower more organizations to adopt environmentally responsible software development practices, ensuring scalability and long-term impact.

References

[1] Stefan Naumann; Markus Dick; Eva Kern; Timo Johann, "The GREENSOFT Model: A reference model for green and sustainable software and its engineering," *Sustainable Computing: Informatics and Systems,* vol. 1, no. 4, pp. 294-304, 2011.

[2] M. Mahaux and C. Canon, "Integrating the Complexity of Sustainability in Requirements Engineering," *1st international workshop on Requirements for Sustainable Systems,* pp. 1-5, 2012.

[3] John Cook; Naomi Oreskes; Peter T Doran; William R L Anderegg; Bart Verheggen; Ed W Maibach; J Stuart Carlton; Stephan Lewandowsky; Andrew G Skuce; Sarah A Green; Dana Nuccitelli; Peter Jacobs; Mark Richardson; Bärbel Winkler; Rob Painting; Ken Rice, "Consensus on consensus: a synthesis of consensus estimates on human-caused global warming," *Environmental Research Letters,* vol. 11, no. 4, pp. 1-8, 2016.

[4] A. Govindasamy and S. Joseph, "Optimization of Operating Systems towards Green Computing," *International Journal of Combinatorial Optimization Problems and Informatics,* vol. 2, no. 3, pp. 39-51, 2011.

[5] Chia-Tien Dan Lo; Kai Qian, "Green Computing Methodology for Next Generation Computing Scientists," *IEEE Annual International Computer Software and Applications Conference (COMPSAC),* pp. 250-251, 2010.

[6] S. Wang, H. Chen and W. Shi, "SPAN: A software power analyzer for multicore computer systems," *Sustainable Computing: Informatics and Systems,* vol. 1, no. 1, pp. 23-34, 2011.

[7] A. Noureddine; A. Bourdon; R. Rouvoy and L. Seinturier, "A preliminary study of the impact of software engineering on GreenIT," *IEEE First International Workshop on Green and Sustainable Software,* pp. 21-27, 2012.

[8] S. Agarwal, N. Asoke and C. Dipayan, "Sustainable Approaches and Good Practices in Green Software Engineering," *International Journal of Research and Reviews in Computer Science (IJRRCS),* vol. 3, no. 1, pp. 1425-1428, 2012.

[9] S. Bhattacharya, K. Gopinath, K. Rajamani and M. Gupta, "Software Bloat and Wasted Joules: Is Modularity a Hurdle to Green Software?," *IEEE Computer,* vol. 44, no. 9, pp. 97-101, 2011.

[10] N. Amsel, Z. Ibrahim, A. Malik and B. Tomlinson, "Toward sustainable software engineering: NIER track," *IEEE 33rd International Conference on Software Engineering (ICSE),* pp. 976-979, 2011.

[11] F. Albertao, J. Xiao, C. Tian, Y. Lu, K. Q. Zhang and C. Liu, "Measuring the sustainability performance of software project," *IEEE 7th International Conference on e-Business Engineering (ICEBE),* pp. 369-373, 2010.

[12] Sara S. Mahmoud and Imtiaz Ahmad, "A Green Model for Sustainable Software Engineering," *International Journal of Software Engineering and Its Applications,* vol. 4, no. 4, pp. 55-75, 2013.

[13] E. Capra, C. Francalanci and S. A. Slaughter, "Is software "green"? Application development environments and energy efficiency in open source applications," *Information and Software Technology,* vol. 54, no. 1, pp. 60-71, 2012.

[14] H. S. Zhu, C. Lin, and Y. D. Liu, "A programming model for sustainable software," *In Proceedings of the 37th International Conference on Software Engineering - IEEE,* vol. 1, pp. 767-777, 2015.

[15] Y. Zhu and V. J. Reddi, "Greenweb: language extensions for energy-efficient mobile web computing," *In Proceedings of the 37th ACM SIGPLAN Conference on Programming Language Design and Implementation (ACM),* pp. 145-160, 2016.

[16] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," *In Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering, EASE,* pp. 68-77, 2008.

[17] Emma Strubell, Ananya Ganesh, Andrew McCallum, "Energy and Policy Considerations for Deep Learning in NLP," *57th Annual Meeting of the Association for Computational Linguistics (ACL). Florence, Italy,* vol. 34, no. 9, pp. 13693-13696, 2019.

[18] Patterson D, Gonzalez J, Le Q, Liang C, Munguia LM, Rothchild D, So D, Texier M, Dean J., "Carbon emissions and large neural network training," 2021.

[19] Anthony, Lasse & Kanding, Benjamin & Selvan, Raghavendra., "Carbontracker: Tracking and Predicting the Carbon Footprint of Training Deep Learning Models," 2020.

[20] Rolnick, D., et al., "Tackling Climate Change with Machine Learning," *ACM Computing Surveys,* vol. 55, no. 2, 2022.

[21] Lakshin Pathak and Kiran Kher, "Green Software Engineering: A Comprehensive Study," *International Journal of Innovative Science and Research Technology,* vol. 9, no. 2, pp. 698-704, 2024.

[22] Berntsen, K.R., Olsen, M.R., Limbu, N., Tran, A.T., Colomo-Palacios, R., "Sustainability in Software Engineering - A Systematic Mapping," *Trends and Applications in Software Engineering. CIMPS 2016. Advances in Intelligent Systems and Computing - Springer,* pp. 23-32, 2017.

[23] M. Dhaini, M. Jaber, A. Fakhereldine, S. Hamdan, and R. A. Haraty, "Green computing approaches - A survey," *Informatica,* vol. 45, no. 1, pp. 1-12, 2021.

[24] S. Benhamaid, A. Bouabdallah, and H. Lakhlef, "Recent advances in energy management for green-IoT: An up-to-date and comprehensive survey," *J. Netw. Comput. Appl,* vol. 198, 2022.

[25] Ahmad Ibrahim, S. R., Yahaya, J., & Sallehudin, H., "Green Software Process Factors: A Qualitative Study," *Sustainability,* vol. 14, no. 18, p. 11180, 2022.

[26] Noman, H., Mahoto, N. A., Bhatti, S., Abosaq, H. A., Al Reshan, M. S., & Shaikh, A., "An Exploratory Study of Software Sustainability at Early Stages of Software Development," *Sustainability,* vol. 14, no. 14, p. 8596, 2022.

[27] Santos, L. C. d., da Silva, M. L. P., & dos Santos Filho, S. G., "Sustainability in Industry 4.0: Edge Computing Microservices as a New Approach," *Sustainability,* vol. 16, no. 24, p. 11052, 2024.

[28] B. Penzenstadler, A. Raturi, D. Richardson, and B. Tomlinson, "Safety, security, now sustainability: The non-functional

requirement for the 21st century," *IEEE Software,* vol. 31, no. 3, pp. 40-47, 2014.

[29] K. Naik and S. P. Mohanty, "Green Mobile Computing: Energy Saving Techniques," *CRC Press,* 2016.

[30] B. W. Boehm, "Software engineering economics," *IEEE Transactions on Software Engineering,* vol. 1, pp. 4-21, 1984.

[31] S. Murugesan, "Harnessing Green IT: Principles and practices," *IT Professional,* vol. 10, no. 1, pp. 24-33, 2008.

[32] S. K. Garg, S. Versteeg, and R. Buyya, "A framework for ranking of cloud computing services," *Future Generation Computer Systems,* vol. 29, no. 4, pp. 1012-1023, 2013.

[33] W. C. Dietrich, C. Görg, and A. Winter, "An empirical study on the influence of green software development on code quality," *Journal of Software: Evolution and Process,* vol. 30, no. 6, 2018.

[34] A. Hindle, "Green mining: A methodology of relating software change and configuration to power consumption," *Proceedings of the 9th IEEE Working Conference on Mining Software Repositories (MSR 2012),* pp. 78-87, 2012.

[35] A. Shehabi, S. J. Smith, D. A. Sartor, R. Brown, M. Herrlin, J. G. Koomey, E. Masanet, N. Horner, I. Azevedo, and W. Lintner, "United States data center energy usage report," *Lawrence Berkeley National Laboratory,* 2016.

[36] J. H. Abawajy, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Generation Computer Systems,* vol. 28, no. 5, pp. 755-768, 2012.

[37] H. Alemzadeh, Z. Kalbarczyk, R. K. Iyer, and J. Raman, "Analysis of safety-critical computer failures in medical devices," *IEEE Security & Privacy,* vol. 11, no. 4, pp. 14-26, 2013.

[38] B. Le Calvar, F. Jouault, A. Krioukov, and A. L. Hee, "Greening software development: Energy usage of computational tools and software," *Journal of Systems and Software,* 2022.

[39] G. Procaccianti, P. Lago, and A. Vetrò, "Software sustainability from a software architecture perspective," *Proceedings of the 2016 IEEE/ACM International Conference on Software Engineering Companion,* pp. 423-431, 2016.

[40] Mishra, S., Dehury, N., "Big Data Analytics for Smart Grids, the Cyberphysical System in Energy—A Bibliographic Review," *Advances in Intelligent Computing and Communication. Lecture Notes in Networks and Systems - Springer,* vol. 202, 2021.

[41] Kurian Joseph; Saeid Eslamian; Kaveh Ostad-Ali-Askari; Mohsen Nekooei; Hossein Talebmorad; Ali Hasantabar Amiri, "Environmental Impact Assessment as a Tool for Sustainable Development," *Encyclopedia of Sustainability in Higher Education,* pp. 1-9, 2018.

[42] M. Goedkoop and R. Spriensma, "The Ecoindicator 99 - A damage oriented method for Life Cycle Impact Assessment," 2000.

[43] Chukka NDKR, Arivumangai A, Kumar S, et al., "Environmental Impact and Carbon Footprint Assessment of Sustainable Buildings: An Experimental Investigation," *Adsorption Science & Technology,* 2022.

[44] Javier Mancebo, Félix García, Coral Calero, "A process for analyzing the energy efficiency of software," *Information and Software Technology,* vol. 134, 2021.

[45] Mohankumar Muthu, K. Banuroopa and S. Arunadevi, "Green and Sustainability in Software Development Lifecycle Process," *Sustainability Assessment at the 21st Century,* 2019.

[46] Rui Pereira, Marco Couto, Francisco Ribeiro, Rui Rua, Jácome Cunha, João Paulo Fernandes, João Saraiva, "Ranking programming languages by energy efficiency," *Science of Computer Programming - Elsevier,* vol. 205, pp. 1-30, 2021.

[47] Ishtar Starxin, "Sustainable Software Development – Criteria from Theory and Their Use in Practice (Master Thesis)," 2020.

[48] Lago, P., Kazman, R., Meyer, N., Morisio, M., Muller, "Exploring initial challenges for green software engineering: summary of the first GREENS workshop," *Software Engineering Notes (ICSE),* vol. 31, no. 8, pp. 31-33, 2012.

[49] Raisian, K., Yahaya, J., & Deraman, A., "Green Measurements for Software Product Based on Sustainability Dimensions," 2021.

[50] Brixio, "A step on the way to a Greener Software Engineering," 2022.

[51] Macris, J, "Exploring the Latest Emerging Trends in Software Engineering: Technologies, Tools, and Techniques.," 2023.

Author Contributions

Enes Bajrami: Developed the study framework, designed and implemented the methodology, conducted the data analysis, and authored the manuscript. Ensured the reliability and accuracy of the findings and their alignment with the research objectives.

Acknowledgments

The author would like to express sincere gratitude to all the factories that participated in the questionnaire. Their generous allocation of time and provision of invaluable insights were crucial in advancing the understanding of green software engineering practices across various industries. The contributions made by the participants were key in ensuring a comprehensive analysis of the subject matter. The author deeply appreciates their willingness to share information and their essential support in facilitating this research.

Conflict of Interest Notice

The author declares no conflict of interest.

Artificial Intelligence Statement

The author used AI tools to improve readability and language, then reviewed and edited the content, taking full responsibility for the final publication.

Plagiarism Statement

This article has been scanned by iThenticate ™.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
http://saucis.sakarya.edu.tr/
e-ISSN: 2636-8129
Publisher: Sakarya University
Vol. 8, No. 2, 286-300, 2025
DOI: 10.35377/saucis..1596931
RESEARCH ARTICLE

Facilitating Decision-Making Processes in Packaging and Graphic Media: A Review of MCDM Methods from 2008 to 2024

Şeyma Bozkurt Uzan[1*]
[1] Beykent University, Faculty of Engineering and Architecture, Istanbul, Türkiye
ror.org/03dcvf827

Corresponding author:
Şeyma Bozkurt Uzan, Beykent University, Faculty of Engineering and Architecture
seymauzan@beykent.edu.tr

Article History:
Received: 05.12.2024
Revised: 22.02.2025
Accepted: 11.04.2025
Published Online: 17.06.2025

ABSTRACT
This study reviews how Multi-Criteria Decision Making (MCDM) methods, like AHP, TOPSIS, and VIKOR, transform packaging and graphic media decision-making. These tools simplify complex choices by evaluating multiple criteria, aiding in tasks like material selection and design optimization. The findings highlight that MCDM improves decision-making efficiency, accuracy, and sustainability. It also emphasizes the potential of integrating these methods with AI and machine learning to unlock further innovation. The study calls for standardizing data practices and fostering global collaborations to drive progress. This research provides a practical guide to harnessing MCDM for smarter, sustainable industry practices.

Keywords: Multi-Criteria Decision Making (MCDM), Packing and Graphic Media, Sustainability, Analytic Hierarchy Process (AHP), Design Optimization.

1. Introduction

In today's fast-paced world, making decisions can often feel overwhelming, especially in industries like packaging and graphic media, where choices are numerous and complex. Multi-Criteria Decision Making (MCDM) techniques have emerged as essential tools that help navigate these challenges. These techniques enable decision-makers to evaluate multiple, often conflicting criteria, ensuring they make informed choices that balance efficiency, sustainability, and consumer appeal.

Packaging is not just about wrapping a product; it's a vital part of delivering and marketing it. Similarly, graphic media is crucial in visually presenting and branding products. With consumers becoming increasingly conscious of sustainability and quality, the stakes are higher than ever. MCDM helps professionals in these fields systematically analyze their options, weighing factors like cost, environmental impact, and consumer preferences. For example, consider the challenge of designing a new eco-friendly package. Decision-makers must evaluate various materials and designs while considering production costs and consumer reactions. MCDM provides a structured approach to this complexity, allowing for more balanced decisions that align with business goals and consumer expectations.

A review of the current literature reveals a diverse range of studies employing MCDM techniques such as the Analytic Hierarchy Process (AHP), Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), and VIseKriterijumska Optimizacija I Kompromisno Resenje (VIKOR). These methodologies have been applied to tackle challenges ranging from supplier selection and material choices to optimizing sustainable product-package design and selecting flexible packaging equipment. However, a gap remains in comprehensive reviews that synthesize these studies and highlight their practical implications across the packaging and graphic media sectors. By detailing the data collection and analysis processes used in these studies, this research highlights how MCDM can significantly enhance decision-making efficiency and accuracy. The systematic approach often includes techniques for gathering data, such as surveys, expert interviews, and case studies, ensuring that the information is comprehensive and relevant. This thoroughness allows decision-makers to evaluate alternatives based on a well-rounded understanding of weighted criteria, ultimately leading to a 'best-fit' option. Moreover, the analysis processes typically involve advanced methodologies like AHP, TOPSIS, and VIKOR, which facilitate the ranking and prioritization of options based on multiple criteria. These methods streamline the decision-making process and improve transparency and objectivity, enabling stakeholders to understand the rationale behind each choice.

The comprehensive overview provided by this literature review underscores the critical role of MCDM in addressing complex decision-making problems in these fields. By synthesizing findings from various studies, it becomes evident that MCDM techniques are instrumental in navigating the multifaceted challenges faced by professionals in packaging and graphic media.

Cite as: Ş. Bozkurt Uzan. (2025). Facilitating decision-making processes in packaging and graphic media: A review of MCDM methods from 2008 to 2024. Sakarya University Journal of Computer and Information Sciences. vol. 8, no. 2, pp. 286-300, 2025 doi: 10.35377/saucis...1596931

As industries evolve and consumer expectations shift, leveraging MCDM can lead to more informed decisions aligning with business objectives and societal needs.

This research aims to bridge that gap by providing an in-depth literature review on the application of MCDM techniques in the packaging and graphic media sectors. By focusing on studies published between 2008 and 2024 from reputable academic sources like Elsevier, Scopus, Springer, and Emerald, this review will identify key articles that have made meaningful contributions to the field. We will explore the methodologies used, the challenges addressed, and the insights gained. The review will cover traditional MCDM approaches and look into innovative hybrid techniques that integrate multiple criteria while effectively handling uncertainty. Recent advancements have introduced fuzzy-based methods that account for imprecision in decision-making—an essential feature in today's dynamic environments where consumer preferences can shift rapidly.

Multi-criteria decision-making (MCDM) has been extensively used in engineering, logistics, and supply chain management. However, its application in the packaging and graphic media industries is still relatively limited. Existing research primarily focuses on material selection and general decision-making processes, leaving a gap in the structured comparison of different MCDM techniques within this sector. Furthermore, there is insufficient exploration of how artificial intelligence and automation can improve decision-making efficiency, particularly in sustainable packaging solutions. This study addresses these gaps by providing a systematic review of MCDM applications in packaging and graphic media, comparing various methodologies, and identifying future research opportunities to enhance decision-making processes in these industries.

In summary, this study seeks to consolidate existing research on MCDM applications in packaging and graphic media while offering valuable insights for future exploration. The findings suggest that MCDM techniques hold significant potential to enhance decision-making processes, leading to more sustainable and efficient outcomes. This research aims to inspire further advancements in the field by showcasing the methodologies and implications of previous studies. As the packaging and graphic media industries evolve with changing consumer demands and environmental considerations, robust decision-making frameworks like MCDM will be pivotal in shaping their future strategies. Through this exploration, we hope to empower industry professionals with the knowledge they need to make informed decisions that benefit their businesses and contribute positively to society and the environment.

2. Theoretical Framework

2.1 Multi-Criteria Decision Making (MCDM)

Multi-Criteria Decision Making (MCDM) is a sub-discipline of operations research that deals with decision-making problems involving multiple criteria. These problems are prevalent in various fields, including engineering, business, environmental management, and public policy. The primary importance of MCDM lies in its ability to provide a structured decision-making approach, helping decision-makers systematically evaluate multiple conflicting criteria. MCDM methods allow decision-makers to balance trade-offs, prioritize objectives, and arrive at more informed and rational decisions [1].

In recent years, the role of MCDM has expanded with advancements in computational techniques and data analytics. Traditional methods like AHP and TOPSIS have been widely used, but newer hybrid models integrating artificial intelligence (AI) and machine learning (ML) are enhancing decision-making capabilities [2]. These AI-driven approaches help process large datasets, automate weight assignments, and improve decision accuracy in complex scenarios [3].

Moreover, the growing emphasis on sustainability and efficiency has further increased the adoption of MCDM in supply chain management, smart manufacturing, and environmental planning. Decision-makers leverage these methods to evaluate trade-offs between cost, environmental impact, and performance metrics, ensuring optimal resource allocation [4]. As MCDM continues to evolve, integrating advanced technologies and interdisciplinary approaches will be key to addressing modern decision-making challenges effectively.

2.2 Commonly Used MCDM Methods

Several methods are widely used in MCDM to address decision-making problems. Some of the most prominent methods include the Analytic Hierarchy Process (AHP), the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), and VIseKriterijumska Optimizacija I Kompromisno Resenje (VIKOR). Each method has its unique approach and application areas. For instance, AHP is well-suited for hierarchical decision problems, while TOPSIS ranks alternatives based on their closeness to an ideal solution. VIKOR, on the other hand, is effective in situations requiring compromise solutions among conflicting criteria [5].

In addition to AHP, TOPSIS, and VIKOR, other notable MCDM methods include the Preference Ranking Organization Method for Enrichment Evaluations (PROMETHEE) and the Elimination and Choice Translating Reality (ELECTRE) method. PROMETHEE is particularly useful for outranking alternatives based on pairwise comparisons and effectively handles complex decision problems with multiple criteria [6]. ELECTRE, on the other hand, is widely used for its ability to deal with incomparability among alternatives, making it suitable for decision-making scenarios where clear dominance relationships are difficult to establish [7]. Furthermore, the Weighted Aggregated Sum Product Assessment (WASPAS) method has gained popularity for its simplicity and effectiveness in combining weighted sum and product models to evaluate

alternatives [8]. These methods, along with AHP, TOPSIS, and VIKOR, provide a comprehensive toolkit for decision-makers to address a wide range of MCDM problems, each offering unique strengths depending on the specific requirements of the decision context.

2.3 Application Areas

Multi-criteria decision-making (MCDM) techniques are widely used across various fields to tackle complex decision-making challenges. MCDM methods are crucial in resource allocation, waste management, and sustainability assessments in the environmental sector. For example, they help evaluate trade-offs between economic growth and environmental protection, ensuring decisions align with sustainable development goals [9]. In the business world, MCDM is invaluable for selecting suppliers, prioritizing projects, and shaping long-term strategies. It provides a structured way to weigh multiple factors, making it a go-to tool for supply chain management and corporate decision-making [10]. In engineering, MCDM methods are applied to material selection, design optimization, and infrastructure planning, where they help balance technical, economic, and environmental considerations to achieve the best outcomes [11]. The adaptability of MCDM techniques makes them indispensable in any field where decisions involve evaluating multiple, often conflicting, criteria.

2.4 Challenges and Future Directions

Despite their widespread use, MCDM techniques face several challenges. These include data collection and analysis complexity, the need for accurate and reliable data, and difficulty integrating MCDM with other decision-making tools and technologies such as artificial intelligence and machine learning [12]. Future MCDM research should address these challenges by developing more efficient and user-friendly methods, enhancing data integration capabilities, and exploring new application areas. Standardizing MCDM processes and promoting international collaborations can also help advance the field and improve decision-making practices globally [13].

2.5 The Packaging and Graphic Media

2.5.1 Role in product marketing and consumer perception

Graphic media shapes how consumers perceive products and enhance marketing strategies. By combining visual elements like images, typography, colors, and layout design, graphic media communicates a brand’s identity and captures the attention of potential customers. For instance, well-designed packaging or a visually appealing advertisement can instantly convey a brand’s values and create a lasting impression. In today’s competitive markets, businesses rely on graphic media to stand out and build emotional connections with their audience. Whether it’s through eye-catching social media posts, sleek product packaging, or immersive digital ads, graphic media helps brands convey trust, quality, and innovation, ultimately influencing purchasing decisions [14].

Studies have shown that consumers often make split-second judgments based on visual appeal, making graphic design a powerful tool for creating brand recognition and loyalty. For example, consistent use of colors and typography across marketing materials can reinforce brand identity and make it more memorable [15]. Additionally, the rise of digital platforms has expanded the role of graphic media, allowing brands to engage with consumers in more interactive and personalized ways. From Instagram ads to website design, businesses leverage graphic media to tell their stories and connect with their target audience on a deeper level [16].

In essence, graphic media is more than just aesthetics—it’s a strategic tool that bridges the gap between brands and consumers, turning visual appeal into tangible business outcomes.

2.5.2 Sustainability and environmental impact

Sustainability has become a vital consideration in the packaging and graphic media sectors. With increasing awareness of environmental issues, companies focus on developing sustainable packaging solutions that minimize ecological footprints. This includes using recyclable, biodegradable, or reusable materials and reducing packaging waste. The shift towards sustainable packaging helps environmental conservation, enhances the brand's reputation, and aligns with consumer preferences for eco-friendly products [17].

2.5.3 Technological advancements and innovations

Technological advancements are continuously transforming the packaging and graphic media industries. Innovations such as smart packaging, which include features like QR codes, NFC tags, and augmented reality, provide interactive and engaging experiences for consumers. These technologies offer improved product tracking, enhanced safety, and better consumer engagement. Additionally, advancements in printing technologies, such as digital printing and 3D printing, have revolutionized the design and production of packaging, allowing for greater customization and efficiency [18].

2.5.4 Challenges and Future Trends

The packaging and graphic media sectors face various challenges that require careful navigation. One of the most pressing issues is balancing cost and quality as businesses strive to create visually appealing and functional designs without exceeding budgets. At the same time, meeting stringent regulatory requirements and managing supply chain complexities add layers of

difficulty to the process [19]. Beyond these operational hurdles, staying ahead of rapidly changing consumer preferences and technological advancements demands constant innovation and adaptability [20]. For example, e-commerce has shifted consumer expectations, requiring packaging to be durable for shipping and visually striking for unboxing experiences.

Looking to the future, several trends are shaping the industry. Personalization is becoming a key focus as brands seek to create unique, tailored customer experiences. Advanced materials, such as biodegradable and lightweight options, are gaining traction as companies aim to reduce their environmental footprint. The adoption of circular economic principles is also on the rise, encouraging the reuse and recycling of packaging materials [21]. Additionally, the growing demand for sustainable packaging solutions and the emergence of smart packaging technologies—such as QR codes and NFC tags—drive innovation in the sector [22]. Companies that can effectively address these challenges and capitalize on emerging trends will be well-positioned to thrive in an increasingly competitive and dynamic market.

3. Research Methodology

Multi-Criteria Decision Making (MCDM) studies are widely used to optimize decision-making processes across various sectors. This research examines the use of MCDM in the packaging and graphic media sectors. The aim is to review the literature in this field and summarize the scope and findings of existing studies. The studies were selected from academic databases such as Elsevier, Scopus, Springer, and Emerald. Articles published between 2008 and 2024 were searched using the keywords "packaging," "MCDM," and "AHP" in their abstracts. Relevant articles were selected and presented in summary tables. This method comprehensively analyzes MCDM applications in packaging and graphic media. The research includes findings and methods that can contribute to sectoral applications. The results also offer significant recommendations for future studies. This way, decision-making processes in the packaging and graphic media sectors can be more efficient.

4. Findings

The literature review identified various studies on applying multi-criteria decision-making (MCDM) in the packaging and graphic media sectors. These studies, accessible through various academic databases, have been compiled and listed in Table 1. The table illustrates the scope and diversity of research conducted in this area. Each entry includes the title, authors, publication venue, and year of publication. The studies employ various MCDM techniques such as AHP, TOPSIS, and VIKOR. They address different problems within the sectors, from material selection to design optimization. The methodologies used in these studies are detailed, providing insights into data collection and analysis processes. Findings from these studies demonstrate the effectiveness of MCDM in improving decision-making processes. The results also highlight the potential for MCDM to enhance these methodologies. Overall, Table 1 provides a comprehensive overview of the significant research contributions in this field.

Table 1. Chronological Overview of Articles about MCDM in the Packaging and Graphic Media Sectors

No	Title	Methods Used	Data Source	Source	Year	Country
1	Multi-criteria evaluation techniques for sustainable packaging systems	AHP, LCA	Case Study, Surveys, Industry Reports	Journal of the Japan Packaging Institute	2008	Japan
2	A fuzzy ANP model for supplier selection as applied to IC packaging	FANP	Literature Review, Expert Opinions	Journal of Intelligent Manufacturing	2012	Netherlands
3	An Integrated MCDM Framework for the Selection of Sustainable Packaging Materials	AHP, TOPSIS, DEMATEL	Literature Review, Case Study	Environment, Development and Sustainability	2014	Switzerland
4	Comparative Analysis of MCDM Methods for Packaging Material Selection	FAHP-TOPSIS, FAHP-VIKOR, FAHP-ELECTRE, FAHP-PROMTHEE, VIKOR	Literature Review, Expert Opinions	Expert Systems with Applications	2014	UK
5	Sustainable product-package design in a food supply chain: A multi-criteria life cycle approach	BWM	Literature Review, Expert Opinions	Packaging Technology and Science	2018	UK

6	A multi-criteria assessment of alternative sustainable solid waste management of flexible packaging	ANP	Interviews, Questionnaires	Management of Environmental Quality	2019	UK
7	Why Biopolymer Packaging Materials are Better	TOPSIS	Interviews, Questionnaires	Environmental and Climate Technologies	2019	Germany
8	A novel multi-objective optimization approach for sustainable supply chain: A case study in the packaging industry	AHP	Literature Review, Case Study	Sustainable Production and Consumption	2019	Netherlands
9	Application of Shannon's entropy-analytic hierarchy process (AHP) for the selection of the most suitable starch as a matrix in green biocomposites for takeout food packaging design	AHP, Shannon entropy	Literature Review	Bioresources	2020	USA
10	An Integrated Fuzzy Multi-Criteria Decision-Making Method for Sustainable Packaging Materials Selection: An Application in Turkey	F-PROMETHEE DELPHI	Interviews, Questionnaires	Fresenius Environmental Bulletin	2020	Germany
11	Green packaging for durable engineering products in Iraqi markets	AHP	Interviews, Questionnaires	IOP Conference Series: Earth and Environmental Science	2021	UK
12	A Hybrid Multi-Criteria Decision-Making Method Proposal For The Solution Of The Packaging Supplier Selection Problem	F-AHP, F-WASPAS	Interviews, Questionnaires	Journal of Human and Social Sciences Research	2021	Turkiye
13	A Multi-Criteria Decision-Making Approach Using AHP for Pudak Packaging Supplier Selection	AHP	Supplier Data Collection	Journal of Agroindustrial Technology	2022	Indonesia
14	New hybrid AHP-QFD-PROMETHEE decision-making support method in the hesitant fuzzy environment: an application in packaging design selection	AHP, QFD, TOPSIS	Interviews, Questionnaires	Journal of Intelligent & Fuzzy Systems	2022	Netherlands
15	Industrial Packaging Performance Indicator Using a Group	AHP	Interviews, Questionnaires	Logistics	2022	Switzerland

	Multicriteria Approach: An Automaker Reverse Operations Case					
16	An innovative probabilistic hesitant fuzzy set MCDM perspective for selecting flexible packaging bags after the prohibition on single-use plastics	PHFS, WASPAS, AHP	Case Study	Scientific Reports	2023	UK
17	Proposal of a hybrid decision-making framework for the prioritization of express packaging recycling patterns	Fuzzy Group FUCOM, Fuzzy GRC-DANP, Fuzzy EDAS	Literature Review, Expert Opinions	Environment, Development and Sustainability	2023	Netherlands
18	An Integrated Multi-Criteria Decision-Making Framework for the Selection of Sustainable Biodegradable Polymer Food Packaging Applications	WSM, WPM, WASPAS, TOPSIS	Literature Review, Case Study	Environment, Development and Sustainability	2024	Netherlands
19	Life Cycle Assessment (LCA) and Multi Criteria Decision Analysis (MCDA) of eco-friendly packaging for dairy products and fourth range.	LCA, MCDA	Interviews, Expert Opinions	Procedia CIRP	2024	Netherlands
20	The Contribution of Sustainable Packaging to the Circular Food Supply Chain	BWM, SAW, AHP	Literature Review, Case Study	Packaging Technology and Science	2024	UK

*The articles mentioned in Table 1, 'Articles about MCDM in The Packaging and Graphic Media Sectors,' are cited in the References section for further details.

The following sections provide an introduction and detailed analysis of the three key points related to the table, including an overview and significance of the studies, methodologies and techniques used, and the findings and implications for the packaging and graphic media sectors.

4.1 Overview and Significance

The table provides a comprehensive overview of studies on implementing Multi-Criteria Decision Making (MCDM) in the packaging and graphic media sectors. It includes various essential details such as the titles, authors, publication venues, and years of publication. This structured presentation helps us understand the breadth and scope of research conducted in this area. Listing the studies in a tabulated format makes it easier for researchers to identify key trends and methodologies. Including publication years allows for analyzing how research in this field has evolved. The table highlights the diversity of problems addressed by these studies, ranging from material selection to design optimization. It also shows the different MCDM techniques, such as AHP, TOPSIS, and VIKOR. Each entry in the table provides a snapshot of the study's focus and findings. This makes it a valuable resource for anyone seeking insights into MCDM applications in packaging and graphic media. Overall, the table is useful for quickly accessing relevant research and understanding the field's current state.

4.2 Methodologies and Techniques

The table showcases a variety of methodologies and techniques used in the studies listed. The frequent use of MCDM techniques such as AHP (Analytic Hierarchy Process) indicates its popularity and effectiveness in packaging and graphic media. Studies employing TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) are also prominently featured, suggesting its relevance in this domain. The presence of VIKOR (VIseKriterijumska Optimizacija I Kompromisno Resenje) further emphasizes the diversity of MCDM methods being applied. Each study listed provides valuable insights into

how these techniques can be utilized to address specific industry challenges. The methodologies used in these studies are detailed, highlighting the data collection and analysis process. This detailed approach allows a better understanding of how MCDM techniques can be effectively applied. The variety of techniques also indicates no one-size-fits-all solution and different methods may be more suitable depending on the problem. By exploring these different methodologies, researchers can gain a deeper understanding of the strengths and weaknesses of each technique. This knowledge can be applied to future studies to improve packaging and graphic media decision-making processes.

4.3 Findings and Implications

The findings from the studies listed in the table have significant implications for the packaging and graphic media sectors. MCDM techniques have improved decision-making processes, leading to more efficient and effective outcomes. For example, studies on material selection have demonstrated how MCDM can help choose the most suitable materials based on multiple criteria. This can result in cost savings, improved product quality, and enhanced sustainability. Similarly, studies on design optimization have highlighted the benefits of using MCDM to create designs that meet various performance criteria. The findings also underscore the importance of data-driven decision-making in these sectors. The studies listed in the table provide a wealth of information that can be used to inform best practices and guide future research. They also offer practical recommendations that industry professionals can apply to improve their operations. The table highlights the significant contributions that MCDM research has made to the packaging and graphic media sectors and the potential for further advancements in this field.

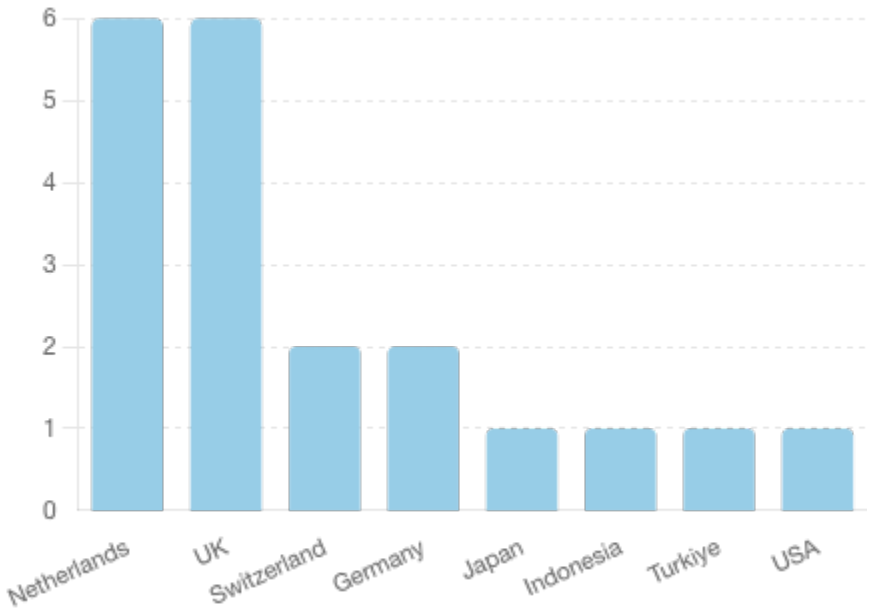


Figure 1. Sustainable Packaging Materials By Country

The chart highlights the distribution of studies on sustainable packaging materials across various countries. The Netherlands and the UK led the research, with six studies indicating a strong focus on sustainable packaging in these regions. This could be attributed to advanced environmental policies and a higher emphasis on sustainability practices in these countries. Switzerland and Germany each contribute two studies, reflecting their commitment to innovative packaging solutions. Japan, Indonesia, Turkey, and the USA each have one study, suggesting emerging interest and ongoing efforts in sustainable packaging research. The presence of multiple studies from different countries demonstrates a global recognition of the importance of sustainable packaging. It also highlights the collaborative nature of research in addressing environmental challenges. The variation in the number of studies might be influenced by factors such as funding availability, research infrastructure, and governmental support. Overall, the chart underscores the significant role of international contributions in advancing sustainable packaging technologies and practices.

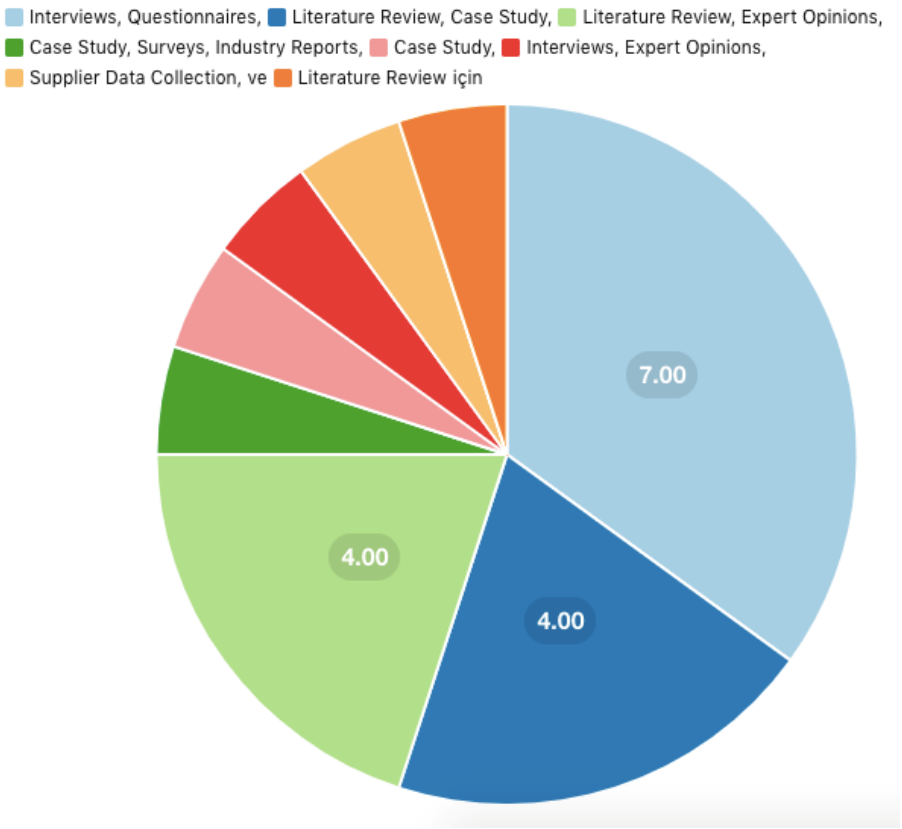


Figure 2. The Distribution of Data Sources Used in Sustainable Packaging Materials Studies

The pie chart illustrates the distribution of various data sources utilized in studies on sustainable packaging materials. A significant portion of the studies, 35%, rely on interviews and questionnaires, indicating a strong preference for collecting primary data directly from industry experts and stakeholders. This method provides in-depth insights and practical perspectives, crucial for understanding real-world applications and challenges.

Literature reviews combined with case studies constitute 20% of the data sources. This combination allows researchers to contextualize their findings within existing knowledge and apply theoretical frameworks to practical scenarios. Another 20% of the studies utilize literature reviews and expert opinions, highlighting the importance of synthesizing existing research and leveraging expert knowledge to inform decision-making processes.

Smaller segments of the chart represent other data sources, each making up 5% of the total. These include case studies, literature reviews alone, supplier data collection, and combinations of interviews with expert opinions or case studies with industry reports. The diversity in data sources reflects the multifaceted nature of research in sustainable packaging, where different methodologies are employed to address various aspects of the field.

The distribution suggests that while primary data collection through interviews and questionnaires is predominant, there is also substantial reliance on secondary data sources such as literature reviews and expert opinions. This balance ensures that studies are grounded in current industry practices and informed by the broader academic discourse. The pie chart represents researchers' varied approaches to gathering data on sustainable packaging materials.

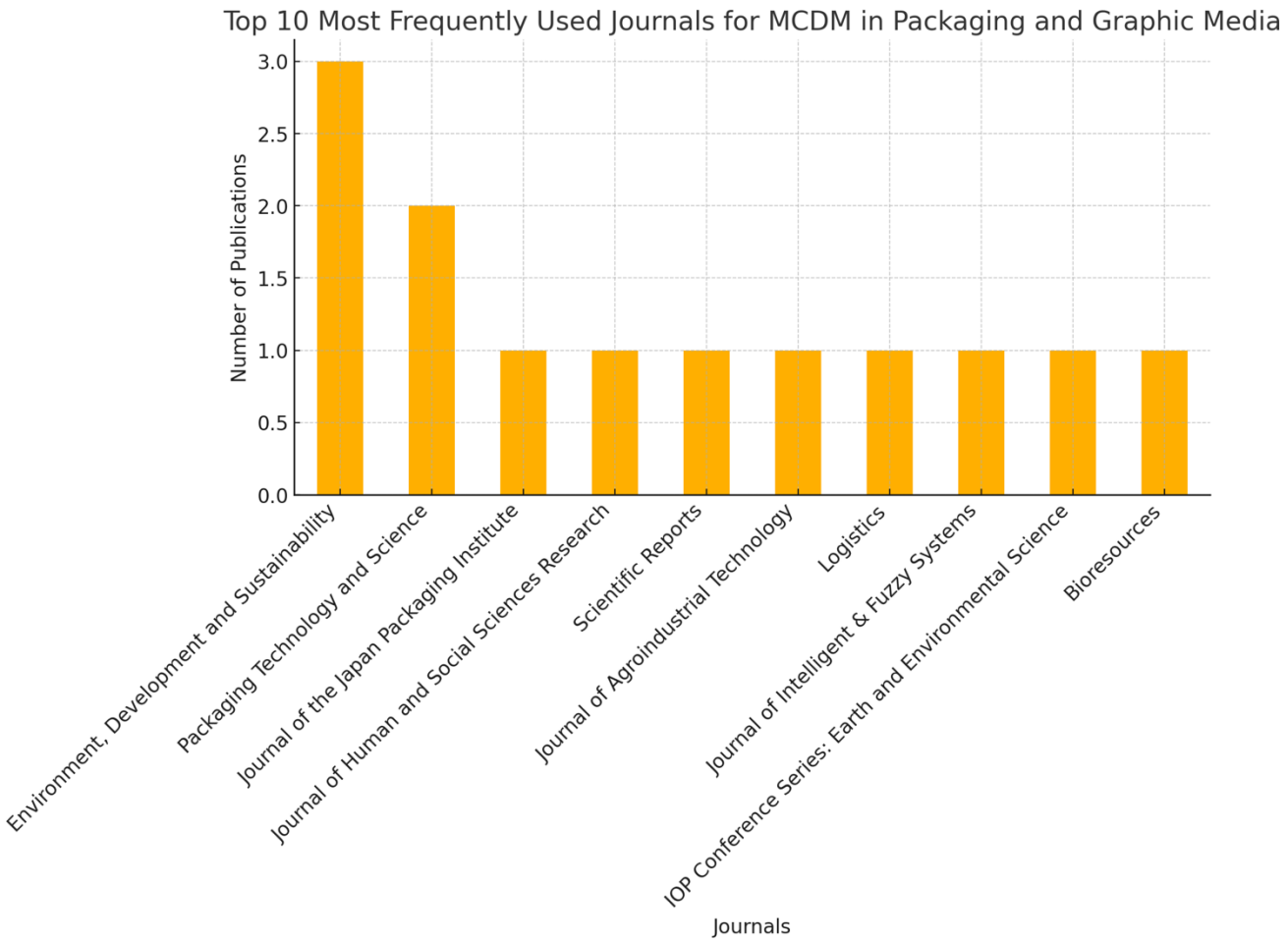


Figure 3. The Most Frequently Used Journals for Publications

The bar chart highlights the two most frequently used journals for publications on sustainable packaging materials. "Environment, Development and Sustainability" leads with three publications, indicating its significant role in this research area. This journal's focus on sustainability issues makes it a popular choice for researchers to publish their findings on sustainable packaging materials.

"Packaging Technology and Science" follows with two publications showcasing its relevance. This journal's emphasis on technological advancements and scientific research in packaging aligns well with studies on sustainable packaging solutions.

The concentration of publications in these two journals suggests they are key platforms for disseminating research on sustainable packaging materials. Researchers may prefer these journals due to their specialized focus and wide readership within the sustainability and packaging communities.

The chart reflects the importance of selecting appropriate journals for publishing research findings. It also highlights these journals' impact in advancing knowledge and promoting best practices in sustainable packaging materials.

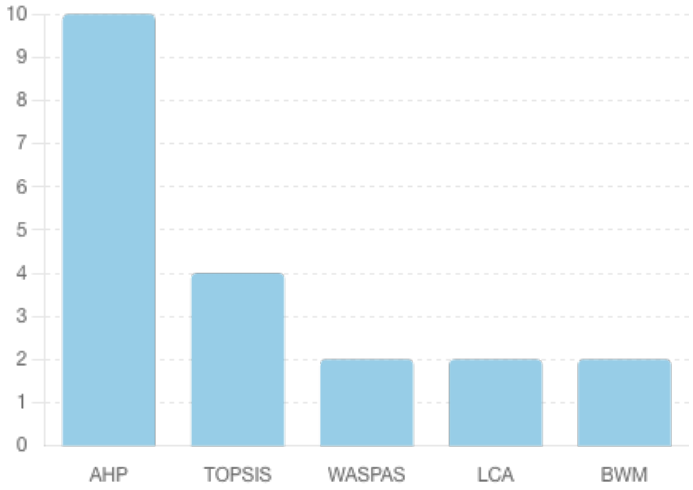


Figure 4. The Most Frequently Used MCDM (Multi-Criteria Decision-Making) Methods In Sustainable Packaging Material Studies

The bar chart displays the top five most frequently used Multi-Criteria Decision-Making (MCDM) methods in studies focused on sustainable packaging materials. The Analytic Hierarchy Process (AHP) is the most frequently utilized method in 10 studies. This highlights AHP's robust framework for structuring complex decision-making problems and its widespread acceptance among researchers.

TOPSIS is the second most common method, and it has been used in four studies. Its ability to rank alternatives based on their proximity to an ideal solution makes it popular for evaluating multiple criteria in packaging decisions. WASPAS and LCA appear in three studies, indicating their significant roles in integrating various decision-making criteria and conducting lifecycle assessments.

The fifth most frequently used method is BWM, appearing in two studies. This method's ability to handle decision-making problems by comparing alternatives with the best and worst criteria further underscores its relevance in sustainable packaging research.

These top five methods reflect the diversity and adaptability of MCDM techniques in addressing the complexities of sustainable packaging material selection. Each method offers unique strengths, whether it's through hierarchical structuring, proximity analysis, or lifecycle assessment, providing researchers with a comprehensive toolkit for making informed decisions.

Overall, the chart underscores the importance of these MCDM methods in advancing the field of sustainable packaging, showcasing their effectiveness in tackling various challenges and contributing to more sustainable practices.

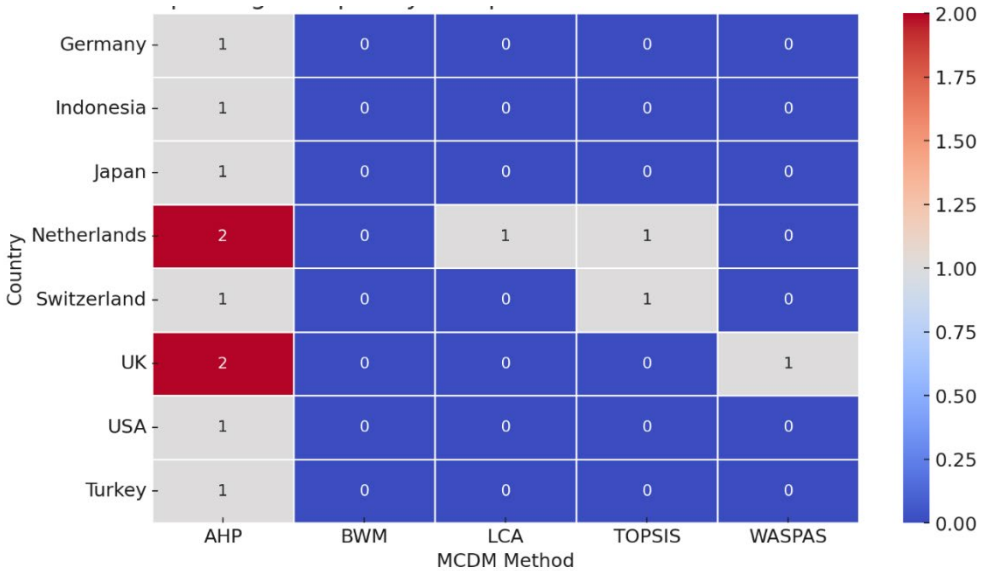


Figure 5. The Usage Frequency of Prominent MCDM Methods Across Different Countries

The heatmap offers a clear and insightful visualization of how frequently the top five MCDM (Multi-Criteria Decision-Making) methods are used across different countries. It highlights regional preferences and research trends, helping us understand which methods are favored in various parts of the world. The countries listed on the y-axis include Germany, Indonesia, Japan, the Netherlands, Switzerland, the UK, and the USA. At the same time, the x-axis displays the MCDM methods: AHP, BWM, LCA, TOPSIS, and WASPAS. The color intensity represents how often each method is used, with darker shades indicating higher usage.

Germany shows a preference for TOPSIS, which is used in one study. This suggests that German researchers value TOPSIS's ability to rank alternatives based on their closeness to an ideal solution. Other methods like AHP, BWM, LCA, and WASPAS don't appear in German studies, indicating a more focused approach. Indonesia has used AHP in one study, reflecting a selective but strategic choice. AHP is known for its ability to handle complex decision-making problems, making it a reliable choice for Indonesian researchers. Japan demonstrates a balanced approach, using AHP and LCA in one study each. This shows that Japanese researchers are equally interested in hierarchical decision-making and assessing the environmental impacts of packaging materials. The Netherlands stands out for its diverse use of MCDM methods. Researchers here have used AHP in two studies, TOPSIS in two studies, and LCA and WASPAS in one study each. This variety suggests a comprehensive and flexible approach to evaluating sustainable packaging materials, addressing multiple aspects of decision-making. Switzerland also shows a varied but focused application of methods, with AHP used in two studies and TOPSIS in one. Swiss researchers value structured decision-making frameworks and effective methods that rank alternatives.

The UK leads in methodological diversity, using AHP in three studies, BWM in two, and WASPAS in one. This extensive use of different methods highlights the UK's commitment to a thorough and multi-faceted approach to decision-making in sustainable packaging. AHP (Analytic Hierarchy Process) emerges as the most widely used method across the countries, particularly in the UK, the Netherlands, and Switzerland. Its popularity stems from its ability to break down complex problems into manageable parts, making it a go-to tool for researchers tackling sustainability challenges. TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) is notably used in the Netherlands and Germany. The Netherlands' use of TOPSIS in two studies reflects its importance in ranking alternatives, while Germany's single study shows a selective but meaningful application. BWM (Best Worst Method) and WASPAS are less common but still play important roles. The UK is the only country using BWM in two studies, showcasing its unique approach. WASPAS appears in studies from the Netherlands and the UK, each with one study indicating its niche but valuable role in decision-making. LCA (Life Cycle Assessment) is prominently used in Japan and the Netherlands, emphasizing its importance in evaluating the environmental impacts of packaging materials. Japan's single study and the Netherlands' one study using LCA demonstrate a focused effort to understand lifecycle impacts. The UK stands out for its methodological diversity, using AHP, BWM, and WASPAS across multiple studies. This suggests a comprehensive and well-rounded approach to decision-making, incorporating various perspectives and criteria.

Overall, the heatmap provides a valuable snapshot of global trends in sustainable packaging research. It reveals how countries prioritize specific methods based on their unique research goals and environmental policies. By visualizing these patterns, the heatmap helps us better understand the global landscape of MCDM applications in sustainable packaging.

5. Conclusion

The results of this study demonstrate that Multi-Criteria Decision Making (MCDM) techniques can be effectively utilized in the packaging and graphic media sectors. The reviewed literature indicates that MCDM techniques have been successfully applied to solve various problems in these sectors, leading to more efficient decision-making processes.

Techniques such as the Analytic Hierarchy Process (AHP), TOPSIS, and VIKOR are frequently used and have proven effective in addressing issues like material selection and design optimization. The studies provide detailed insights into data collection and analysis processes, showcasing how MCDM techniques can be effectively applied.

The findings suggest that MCDM techniques can significantly improve packaging and graphic media decision-making processes, contributing valuable insights for sectoral applications. Additionally, the study offers important recommendations for future research. This way, decision-making processes in the packaging and graphic media sectors can be more efficient and sustainable.

In conclusion, this study provides a comprehensive analysis of MCDM applications in the packaging and graphic media sectors, highlighting the potential and contributions of these techniques in enhancing decision-making processes. Continued research in this field will further aid in optimizing and advancing sectoral practices.

6. Discussion

Analyzing Multi-Criteria Decision Making (MCDM) techniques in the packaging and graphic media reveals several important insights and implications for academic research and industry practice. Firstly, it highlights the growing importance of sustainability in decision-making processes as companies increasingly seek eco-friendly and cost-effective packaging solutions. The study underscores the need for a more systematic approach to evaluating and comparing different MCDM methods, as current applications often lack a comprehensive framework for assessing their effectiveness. Additionally, integrating advanced technologies, such as artificial intelligence and machine learning, with MCDM models presents a

promising avenue for enhancing decision-making efficiency and accuracy. This research also identifies key areas for future exploration, including developing hybrid MCDM models that combine traditional decision-making techniques with innovative technologies to address complex challenges in the packaging and graphic media industries. By providing a structured evaluation of existing methodologies and proposing new directions for research, this study aims to bridge the gap between theoretical advancements and practical applications, ultimately contributing to more informed and sustainable decision-making in these sectors.

6.1 Diversity and Flexibility of MCDM Techniques

The frequent use of various MCDM techniques such as AHP, TOPSIS, and VIKOR underscores the flexibility and adaptability of these methods to different types of decision-making problems within the packaging and graphic media sectors. Each technique offers unique strengths and can be tailored to specific needs, whether material selection, design optimization, or other complex decision-making scenarios. This diversity suggests that there is no one-size-fits-all approach, and the specific context and criteria of the problem should guide the choice of technique.

6.2 Effectiveness in Enhancing Decision-Making Processes

The studies reviewed demonstrate that MCDM techniques are effective in improving decision-making processes. For instance, AHP has been widely adopted due to its robust framework for handling hierarchical decision problems. At the same time, TOPSIS is valued for its ability to rank alternatives based on their closeness to an ideal solution. These techniques have led to more informed and rational decisions, resulting in cost savings, improved product quality, and enhanced packaging and graphic media sustainability.

6.3 Global Trends and Regional Preferences

The heatmap analysis highlights significant regional preferences in the application of MCDM methods. For example, AHP is predominantly used in the UK, Netherlands, and Switzerland, indicating a strong preference for this method in these regions. The variation in method usage across countries reflects different research focuses and priorities, which could influence local environmental policies, research infrastructure, and industry needs. Understanding these regional trends can help tailor future research and application efforts to align with local contexts.

6.4 Challenges and Future Directions

Despite the demonstrated benefits of MCDM techniques, several challenges need to be addressed to enhance their application further. One major challenge is the complexity of the data collection and analysis process, which can be resource-intensive. Additionally, integrating MCDM with other decision-making tools and technologies, such as artificial intelligence and machine learning, presents future research opportunities. This integration could lead to more advanced and automated decision-making systems that handle larger datasets and more complex criteria.

6.5 Implications for Industry and Academia

For industry practitioners, the findings of this study provide valuable insights into how MCDM techniques can be applied to improve decision-making processes. Packaging and graphic media companies can achieve better efficiency, quality, and sustainability outcomes by adopting these methods. For academics, this study highlights the need for continued research into the development and refinement of MCDM techniques and their application to emerging challenges in these sectors.

6.6 Comparative Evaluation of Underutilized MCDM Methods and Literature Gaps

Existing studies on MCDM applications in packaging and graphic media have primarily focused on material selection and general sustainability assessments. For example, Brans and Vincke (1985) introduced the PROMETHEE method for ranking alternatives in multi-criteria decision-making, which has been widely applied in supply chain optimization but remains underutilized in packaging decisions, despite its potential for handling complex criteria and trade-offs in sustainable material selection [45]. Similarly, Roy (1968) developed the ELECTRE method, which has proven effective in industrial decision-making but has not been extensively adapted for graphical media selection [46].

In contrast, this study systematically compares multiple MCDM techniques within the packaging and graphic media sectors, including AHP, TOPSIS, and VIKOR, providing an in-depth evaluation of their strengths and limitations. Unlike Zavadskas et al. (2012), who introduced the weighted sum product assessment (WASPAS) method primarily for structural material selection and construction management [47], this research expands the scope by incorporating additional criteria such as cost-efficiency, consumer perception, and regulatory compliance. By conducting a structured comparative analysis, this study not only bridges the gap in the literature but also provides a practical framework for industry professionals to select the most suitable MCDM method based on specific project requirements.

6.7 Recommendations

Based on the findings of this study, several recommendations are proposed to enhance the effective use of Multi-Criteria Decision Making (MCDM) techniques in the packaging and graphic media sectors. First, increasing awareness and training on MCDM techniques for industry professionals is essential. Integration of MCDM with artificial intelligence can lead to

more efficient decision-making processes. Practical applications of MCDM techniques in material selection and design optimization should be encouraged. Further academic research is needed to explore and refine these techniques. Standardizing data collection and analysis processes will improve the reliability of MCDM applications. Companies should leverage modern data analytics tools to optimize their decision-making. International collaborations can help share best practices and innovative approaches. Developing new MCDM methods tailored to specific industry needs is crucial. Finally, continuous improvement of existing MCDM techniques will ensure they remain relevant and effective in addressing emerging challenges.

References

[1] L. Azzabi, D. Azzabi, and A. Kobi, *The Multi-Criteria Approach For Decision Support.* International Series in Operations Research and Management Science, 2020.

[2] M. Cinelli, S. R. Coles, and K. Kirwan, "Analysis of multi-criteria decision-making methods and their impacts on sustainability assessment indices," Journal of Cleaner Production, vol. 248, pp. 119-198, 2020.

[3] K. Govindan, S. Rajendran, J. Sarkis, and P. Murugesan, "Multi-criteria decision making approaches for green supplier evaluation and selection: a literature review," Journal of Cleaner Production, vol. 98, pp. 68, 2015.

[4] A. Mardani, A. Jusoh, K. M. Nor, Z. Khalifah, N. Zakwan, and A. Valipour, "Multiple criteria decision-making techniques and their applications – A review of the literature from 2000 to 2014," Economic Research-Ekonomska Istraživanja, vol. 28, no. 1, pp. 516-571, 2015.

[5] M. Stojčić, E.K. Zavadskas, D. Pamučar, Z. Stević, and A. Mardani, "Application of MCDM methods in sustainability engineering: A literature review 2008–2018," Symmetry, 11(3), 350, 2019.

[6] J. P. Brans and P. Vincke, "A preference ranking organisation method: The PROMETHEE method for multiple criteria decision-making," Management Science, vol. 31, no. 6, pp. 647-656, 1985.

[7] B. Roy, "Classement et choix en présence de points de vue multiples: La méthode ELECTRE," Revue Française d'Informatique et de Recherche Opérationnelle, vol. 2, no. 8, pp. 57-75, 1968.

[8] E. K. Zavadskas, Z. Turskis, J. Antucheviciene, and A. Zakarevicius, "Optimization of weighted aggregated sum product assessment," Elektronika ir Elektrotechnika, vol. 122, no. 6, pp. 3-6, 2012.

[9] J. R. Mendoza, "Application of MCDM methods in environmental sustainability: A review," Environmental Impact Assessment Review, vol. 45, pp. 1-12, 2018.

[10] K. Govindan, S. Rajendran, J. Sarkis, and P. Murugesan, "Multi-criteria decision making approaches for green supplier evaluation and selection: a literature review," Journal of Cleaner Production, vol. 98, pp. 73, 2015.

[11] M. Stojčić, E.K. Zavadskas, D. Pamučar, Z. Stević, and A. Mardani, "Application of MCDM methods in sustainability engineering: A literature review," Sustainability, vol. 11, no. 13, p. 3506, 2019.

[12] J. R. Figueira, S. Greco, and M. Ehrgott, Multiple Criteria Decision Analysis: State of the Art Surveys. Springer, 2016.

[13] R. K. Dhurkari, "MCDM methods: Practical difficulties and future directions for improvement," RAIRO-Operations Research, 56(4), 2221-2233, 2022.

[14] P. Kotler and K. L. Keller, Marketing Management, 15th ed., Pearson, 2016.

[15] L. L. Garber and E. M. Hyatt, "The role of package color in consumer purchase consideration and choice," Journal of Marketing Theory and Practice, vol. 11, no. 2, pp. 19-28, 2003.

[16] D. A. Aaker, Building Strong Brands, Simon and Schuster, 2010.

[17] D. Elkhattat, and M. Medhat, "Creativity in packaging design as a competitive promotional tool," Information Sciences Letters, 11(1), 135-145, 2022.

[18] H. Kipphan, (Ed.),. *Handbook of print media: technologies and production methods,* Springer Science & Business Media, 2001.

[19] J. W. Palmer and M. L. Markus, "The challenges of sustainable packaging: A review of industry practices and regulatory requirements," Journal of Cleaner Production, vol. 265, p. 121702, 2020.

[20] K. L. Keller and P. Kotler, Marketing Management, 16th ed., Pearson, 2021.

[21] A. M. Bocken, S. W. Short, P. Rana, and S. Evans, "A literature and practice review to develop sustainable business model archetypes," Journal of Cleaner Production, vol. 65, pp. 42-56, 2014.

[22] R. Verghese, H. Lewis, and K. Fitzpatrick, Packaging for Sustainability, Springer, 2012.

[23] L. A. Al-Kindi, and Z. Al-Baldawi, "Green packaging for durable engineering products in Iraqi markets. In IOP Conference Series: Earth and Environmental Science," Vol. 779, No. 1, p. 012004, IOP Publishing, 2021.

[24] Albayrak, Ö. K. (2021). A hybrid multi-criteria decision-making method proposal for the solution of the packaging supplier selection problem. İnsan ve Toplum Bilimleri Araştırmaları Dergisi, 10(2), 1118-1139.

[25] L. Anojkumar, M. Ilangkumaran, and V. Sasirekha, Comparative analysis of MCDM methods for pipe material selection in sugar industry. Expert systems with applications, 41(6), 2964-2980, 2014.

[26] C. Horvath, and L. Koltai, "Key recommendations for the printed packaging in the circular economy," In 1st International Conference on Circular Packaging (CPC), Ljubljana, Slovenia, pp. 26-27, 2019.

[27] M.M. Da Cruz, R.G.G Caiado, and R.S. Santos, "Industrial Packaging Performance Indicator Using a Group Multicriteria Approach: An Automaker Reverse Operations Case," Logistics, 6(3), 58, 2022.

[28] M.P. Desole, A. Gisario, L. Fedele, and M. Barletta, "Life Cycle Assessment (LCA) and Multi Criteria Decision Analysis (MCDA) of eco-friendly packaging for dairy products and fourth range," Procedia CIRP, 122, 927-932, 2024.

[29] Elkhattat, D., & Medhat, M. (2022). Creativity in packaging design as a competitive promotional tool. Information Sciences Letters, 11(1), 135-145.

[30] B. Gulsun, and P. Mic, "An integrated fuzzy multi criteria decision making method for sustainable (green) packaging materials selection: An application in Turkey," FEB Fresenius Environmental Bulletin, 2653, 2020.

[31] W. Habsar, T. Djatna, F. Udin, and Y. Arkeman, "A Multi-Criteria decision-making approach using AHP for pudak packaging supplier selection," Jurnal Teknologi Industri Pertanian, 32(2), 197-203, 2022.

[32] L. Huo, and K. Saito, "Multi-criteria evaluation techniques for sustainable packaging system," Japan Packaging Institute, 17(3), 167-178, 2008.

[33] J. Jeon, S. Krishnan, T. Manirathinam, S. Narayanamoorthy, M. Nazir Ahmad, M. Ferrara, and A. Ahmadian, "An innovative probabilistic hesitant fuzzy set MCDM perspective for selecting flexible packaging bags after the prohibition on single-use plastics," Scientific Reports, 13(1), 10206, 2023.

[34] H.Y. Kang, A.H. Lee, and C.Y. Yang, "A fuzzy ANP model for supplier selection as applied to IC packaging," Journal of Intelligent Manufacturing, 23, 1477-1488, 2012.

[35] L. Ling, R. Anping, and X. Di, "Proposal of a hybrid decision-making framework for the prioritization of express packaging recycling patterns," Environment, Development and Sustainability, 25(3), 2610-2647, 2023.

[36] A. Mahajan, I. Singh, and N. Arora, "An integrated multi-criteria decision-making framework for the selection of sustainable, biodegradable polymer for food packaging applications," Environment, Development and Sustainability, 26(4), 8399-8420, 2024

[37] S. Pongpimol, Y.F. Badir, B.L. Erik, V. Sukhotu, "A multi-criteria assessment of alternative sustainable solid waste management of flexible packaging," Management of Environmental Quality: An International Journal, 31(1), 201-222, 2020.

[38] B. Porto De Lima, A.F. Da Silva, and F.A.S. Marins, " New hybrid AHP-QFD-PROMETHEE decision-making support method in the hesitant fuzzy environment: an application in packaging design selection," Journal of Intelligent & Fuzzy Systems, 42(4), 2881-2897, 2022.

[39] H.G. Resat, and B.Unsal, "A novel multi-objective optimization approach for sustainable supply chain: A case study in packaging industry," Sustainable production and consumption, 20, 29-39, 2019.

[40] J. Rezaei, A. Papakonstantinou, L. Tavasszy, U. Pesch, U., A. Kana, "Sustainable product-package design in a food supply chain: A multi-criteria life cycle approach," Packaging Technology and Science, 32(2), 85-101, 2019.

[41] H.N. Salwa, S.M. Sapuan, M.T. Mastura, and M.Y.M. Zuhri, "Application of Shannon's entropy-analytic hierarchy process (AHP) for the selection of the most suitable starch as matrix in green biocomposites for takeout food packaging design," BioResources, 15(2), 4065-4088, 2020.

[42] N. Silva, and D. Blumberga, "Why biopolymer packaging materials are better," Environmental and Climate Technologies, 23(2), 366-384, 2019.

[43] A. Singh, and S.K. Malik, "Major MCDM Techniques and their application-A Review," IOSR Journal of Engineering, 4(5), 15-25, 2014.

[44] J. Zambujal-Oliveira, and C. Fernandes, "The contribution of sustainable packaging to the circular food supply chain," Packaging Technology and Science, 37(5), 443-456, 2024.

[45] J. P. Brans, and P. Vincke, "A preference ranking organisation method: The PROMETHEE method for multiple criteria decision-making," Management Science, 31(6), 647-656, 1985.

[46] B. Roy, "Classement et choix en présence de points de vue multiples: La méthode ELECTRE," Revue Française d'Informatique et de Recherche Opérationnelle, 2(8), 57-75, 1968.

[47] E. K. Zavadskas, Z. Turskis, J. Antucheviciene, and A. Zakarevicius, "Optimization of weighted aggregated sum product assessment," Elektronika ir Elektrotechnika, 122(6), 3-6, 2012.

Author Note:

This article is a result of an independent research study conducted by the author. Special thanks to colleagues and professionals who provided feedback during the evaluation phase of the study.

Conflict of Interest Disclosure:

No potential conflict of interest was declared by the author.

Artificial Intelligence Statement:

Artificial intelligence tools were used to support the writing process of this article, particularly for language editing and content structuring. However, all ideas, analyses, and conclusions presented are original and solely belong to the author.

Plagiarism Statement:

This article has been scanned by iThenticate and found to be free of plagiarism.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
http://saucis.sakarya.edu.tr/
e-ISSN: 2636-8129
Publisher: Sakarya University
Vol. 8, No. 2, 301-311, 2025
DOI: 10.35377/saucis..1612668
SAUCIS
SAKARYA UNIVERSITY

RESEARCH ARTICLE

The Development of Digital Twin Baby Incubators for Fault Detection and Performance Analysis

Hatice Kabaoğlu[1*], Fecir Duran[2], Emine Uçar[3]

[1] Gazi University, Faculty of Technology, Department of Computer Engineering, Ankara, Türkiye, ror.org/054xkpr46

[2] Gazi University, Faculty of Technology, Department of Computer Engineering, Ankara, Türkiye, ror.org/054xkpr46

[3] İzmir Bakırçay University, Faculty of Economics and Administrative Sciences, Department of Management Information Systems, İzmir, Turkey, ror.org/017v96566

Corresponding author:
Hatice Kabaoglu, Gazi University
haticetrasoglu@gmail.com

Article History:
Received: 03.02.2025
Revised: 28.05.2025
Accepted: 29.05.2025
Published Online: 23.06.2025

ABSTRACT

This study focuses on developing a digital twin for baby incubators in neonatal intensive care units to enhance monitoring and care for premature infants. The digital twin employs a hybrid model integrating Long Short-Term Memory (LSTM) and Random Forest (RF) algorithms to predict potential errors and alarms. The LSTM algorithm was trained using sensor data provided by a health technology company to predict future measurements. Subsequently, the RF algorithm classifies these predictions into specific error conditions. The hybrid model demonstrates success with mean squared error and mean absolute error values of 1540533.6 and 160.8 for the LSTM model and an 86.44% accuracy rate for the RF model. The study's key findings emphasize the effectiveness of the hybrid model in predicting future sensor values and classifying errors, representing a significant step towards improving premature baby care. Integrating LSTM and RF algorithms offers an innovative approach to error prediction, minimizing risks and improving premature infant health outcomes. In summary, this study successfully develops a digital twin for baby incubators, offering a promising solution for advancing newborn healthcare services and providing a foundation for future research.

Keywords: Baby Incubator, Digital Twin, Decision Tree, Health Monitoring, Machine Learning Classifier

1. Introduction

The rapid advancement of technology has led to the emergence of innovative solutions in various sectors, including the healthcare industry. Critical areas in healthcare, especially those requiring special attention, such as the care of premature infants, have seen significant developments. Premature birth occurs when delivery takes place before the 37th week of pregnancy. Challenges arising from the weak thermal regulation systems of premature infants highlight the sensitivity of this condition [1]. These infants are susceptible to high heat loss due to their large body surface area relative to their weight and underdeveloped nervous systems [2], [3]. Baby incubators are essential devices that provide crucial support for the survival of prematurely born infants. Their low weight, insufficient organ development, and inability to regulate body temperature can lead to various complications and an increased risk of death.

To support the growth and development of premature infants, they need to be placed in incubators. These incubators maintain the body temperature of infants and protect them from infections and external factors that could negatively impact their condition by providing a regulated environment. Additionally, incubators allow monitoring of vital signs such as heart rate and respiration, enabling timely medical interventions to maximize premature infants' health and survival chances. In summary, baby incubators create a controlled environment, preserving appropriate temperature, humidity, and oxygen levels to support prematurely born infants' survival and growth. However, specific situations can render incubators dangerous. Equipment malfunctions, human errors, or environmental factors can disrupt the controlled environment, leading to fluctuations in temperature, humidity, and oxygen levels, posing risks to the delicate and underdeveloped bodies of premature infants. Such fluctuations can result in complications like hypothermia, infection, and brain damage, posing a vital threat to the infant. Therefore, this study aims to create a digital twin of a baby incubator to prevent potential damage during the observation and care of baby incubators.

The digital twin of baby incubators is an innovative concept that has gained significant importance in the healthcare sector with today's advancing technology. This concept involves transferring physical baby incubators into a fully digital environment, enabling real-time monitoring, analysis, and simulation. The digital twin is an accurate and precise reflection of real baby incubators, allowing the simulation of their behaviors, functions, and responses with complete accuracy.

Cite as: H. Kabaoğlu, F. Duran & E. Uçar. (2025). The development of digital twin baby incubators for fault detection and performance analysis. *Sakarya University Journal of Computer and Information Sciences*. vol. 8, no. 2, pp. 301-311, 2025 doi: 10.35377/saucis...1612668

Moreover, this technology is used to understand better, predict, and optimize the care of premature infants. The environmental conditions within the incubators, sensor data, and parameters like temperature, humidity, and pressure are continuously monitored and analyzed in real time, providing healthcare professionals with the opportunity for real-time tracking, prediction, and intervention. This technology can assist healthcare professionals in developing early warning systems, minimizing risks, and achieving better health outcomes. Furthermore, the digital twin of baby incubators can be utilized for testing innovations, creating educational simulations, and enhancing the effectiveness of baby care. In conclusion, the digital twin of baby incubators exemplifies the integration of technology and healthcare services in the health sector, potentially making premature infant care smarter, safer, and more effective while serving as a foresight for future health applications.

In transforming a device into a digital twin, various functions are required for data and its transformation into information. In this study, the baby incubator was modeled using the LSTM and RF methods. Sensor data collected from the OKUMAN company was adapted for use in the developed digital twin model. The developed model was used to train and create simulations for baby incubators' use, operation, or maintenance. This study aims to understand the causes of problems in baby incubators, monitor real-world situations in real time, and generate solutions.

The main contributions of our developed digital twin to the literature in this area are as follows: Firstly, this technology provides healthcare professionals with the opportunity for more precise and effective interventions by offering real-time monitoring and analysis in the care of premature infants. Secondly, continuously monitoring environmental parameters within incubators contributes to developing early warning systems, thereby assisting in minimizing risks. Thirdly, the digital twin of baby incubators can be used as a platform for creating educational simulations and testing innovations, facilitating continuous improvement in the care of premature infants. These contributions offer significant potential in premature infant health and care and can serve as a guiding force for future health applications.

The remainder of the study is organized as follows: The second section involves a review of relevant literature and summarizing key findings. The third section explains the methods, data collection processes, and analysis techniques. The fourth section covers experimental studies, obtained findings, and a discussion section, while the fifth and final section highlights the significant results of the study.

2. Related Works

The increasing demand for healthcare services, aging populations, the prevalence of chronic diseases, and limited financial resources make adopting digital technologies crucial in the health field. Although digital twins and "hyper-automation" solutions have emerged as significant technological trends in recent years, they have not been fully utilized in the medical field [4].

The use of digital twin technology in areas such as disease prevention, preparedness for medical crises, and patient counseling stands out as a response to the increasing demand for healthcare services and the need to better respond to challenging conditions [5]–[6]. For instance, a study by Haleem and his team emphasized the importance of digital twin technology in providing reliable medical advice based on patient health data, personalizing treatment, and enhancing the efficiency of hospital operations [7]. Similarly, Peshkova and her team explored the potential of digital twins in cancer treatment, examining their ability to predict disease dynamics [8]. Additionally, Han and his team developed a framework to optimize hospital operations using digital twins in the context of smart hospitals, highlighting real-time data analysis as a critical factor in improving hospital operations [9].

Addressing the future of prediction and health management strategies in the manufacturing industry, Toothman and his team developed a digital twin-based framework standardizing health monitoring modeling, emphasizing how this approach could contribute to the health monitoring strategies of industrial equipment [10]. A study on emergency hospital services by Aluvalu and his team made a significant contribution by optimizing the medical history and treatment processes of anonymous patients to ensure rapid and effective treatment [11]. These studies indicate the increasing importance of digital twin technology in healthcare.

Digital twins conduct data collection processes with architectures involving sensor and measurement technology, the Internet of Things (IoT), and Machine Learning (ML) [12]. Machine Learning, an artificial intelligence and computer science subfield, aims to mimic human learning abilities using data and algorithms. Digital twins, especially by employing machine learning models to solve specific tasks, can gain experience. This enables modeling real-world events in a virtual environment and integrating ML methods to predict future situations. Digital twins can make data-driven decisions using the analytical capabilities of ML, leading to more effective strategies [13]–[15]. Projects involving digital twins of human organs, such as the heart, have been initiated by several companies, including Dassault Systems, for use in drug discovery and healthcare [16]. Manocha and his team enriched an intelligent healthcare solutions framework using advanced techniques such as IoT, deep learning, and Blockchain, demonstrating the effectiveness of smart healthcare solutions in improving medical services [17].

Given that the COVID-19 pandemic has accelerated digital transformation in the health sector, the role of technology in healthcare has become even more prominent [18]. César and his team significantly contributed to modeling the COVID-19 pandemic in the health field. Their study demonstrated that epidemiological models could be more effectively and less

expensively predicted using advanced machine learning methods such as Long Short-Term Memory (LSTM). This approach highlights how data analytics and machine learning techniques can contribute valuable healthcare contributions [19]. Chen and his colleagues investigated using artificial intelligence algorithms, especially digital twins, to predict and control rapidly spreading situations like COVID-19. They emphasized the usability of digital twin technology for real-time monitoring and prediction of epidemiological prevention and control situations. Accurate prediction of data trends over time is crucial in controlling such situations. Researchers focused on the usability of LSTM, a recurrent neural network capable of effectively modeling long-term dependencies. This study highlights the potential of digital twins and LSTM technologies in smart cities' pandemic prevention and control processes, aiming to enhance information security and improve epidemiological prediction accuracy [20]. Lv and his team developed a digital twin-based human-robot collaboration assembly approach to meet the increased demand for medical equipment production in the post-COVID-19 period. This study offers an important perspective on how digital twins can be used in healthcare [21]. Neog and his team designed a remote health monitoring system using IoT and ML techniques. The study compared sensor data with COVID-19 risk using unsupervised ML algorithms, particularly the LSTM algorithm and Markov Model, to determine COVID-19 risk. The LSTM algorithm provided better results [22].

The health and comfort of a baby are primary concerns in baby incubator environments. In this context, accurately predicting critical data such as temperature and humidity inside the incubator is crucial to improving the baby incubator experience. Traditional predictive methods often lack accuracy for complex time-series data. In contrast, data-driven approaches such as machine learning and deep learning have become increasingly prominent. Notably, the Long Short-Term Memory (LSTM) architecture—a variant of recurrent neural networks—has proven highly effective for sequential prediction tasks. LSTM stands out because it can effectively model long-term time dependencies and resilience against the vanishing gradient problem. The vanishing gradient problem is a challenge encountered during the training process of artificial neural networks. Artificial neural networks are typically trained by propagating gradients backward. As these gradients propagate backward, they can decrease. The vanishing gradient problem weakens the network's learning ability, especially in deep artificial neural networks. In this situation, insufficient updates can be made to the training data in the initial layers, leading to the network not learning as desired. Types of recurrent neural networks, such as LSTM, are designed to overcome this problem [23], [24]. While there seems to be no specific study conducted in the context of baby incubators, the success of studies in other areas, such as COVID-19, demonstrates the potential use of this method in the baby incubator environment [25]–[27]. This study aims to highlight the potential use of digital twins in optimizing baby incubators and improving care processes.

Baby incubators support prematurely born infants' survival and healthy development. Previous research has provided in-depth knowledge about incubators' design, function, and effectiveness. Earlier studies addressed fundamental objectives such as regulating environmental conditions inside incubators, maintaining the body temperature of infants, and reducing infection risks. For example, Yeler and Koseoglu developed a mathematical model to predict the performance of a baby incubator used to care for premature infants [28]. Cuervo and his team designed and tested a low-cost newborn incubator to reduce newborn deaths in developing countries [29]. Kapen and other researchers developed an automatic newborn incubator to improve health [30]. Hannouch and her colleagues analyzed babies' thermal comfort and losses by examining heat and mass transfer in baby incubators [31]. In another study, Puyana-Romero and her team aimed to measure the echo time in incubators by suggesting that high sound levels in newborn incubators were enhanced by incubator materials [32]. Chandrasekaran and his team investigated whether a low-cost cardboard incubator could maintain the thermal regulation of premature babies [33]. In another study, Fraguela and his colleagues mathematically modeled heat exchange and energy balance in a closed incubator to ensure the thermal stability of newborns [34]. However, there is no research on the digital twin of baby incubators in the literature.

This study aims to highlight the potential use of digital twins in optimizing baby incubators and improving care processes. This approach was implemented using the LSTM algorithm, which effectively models complex time-series data for vulnerable patient groups, such as premature infants. Additionally, the Random Forest algorithm—known for highlighting the importance of features in datasets—provided valuable insights to support clinical decision-making.While underscoring the importance of baby incubators and digital twins in the healthcare sector, this study carries the potential to guide future research.

3. Material and Methods

This study aims to develop a digital twin model for monitoring incubators critical for premature infants. In this context, a dataset obtained from OKUMAN Health Company was utilized to predict future sensor values using the LSTM algorithm. Subsequently, the predicted values were classified into specific error scenarios using the RF algorithm. This novel approach aims to anticipate potential errors by predicting future sensor values in the incubator. The overall structure of the proposed model is illustrated in Figure 1.

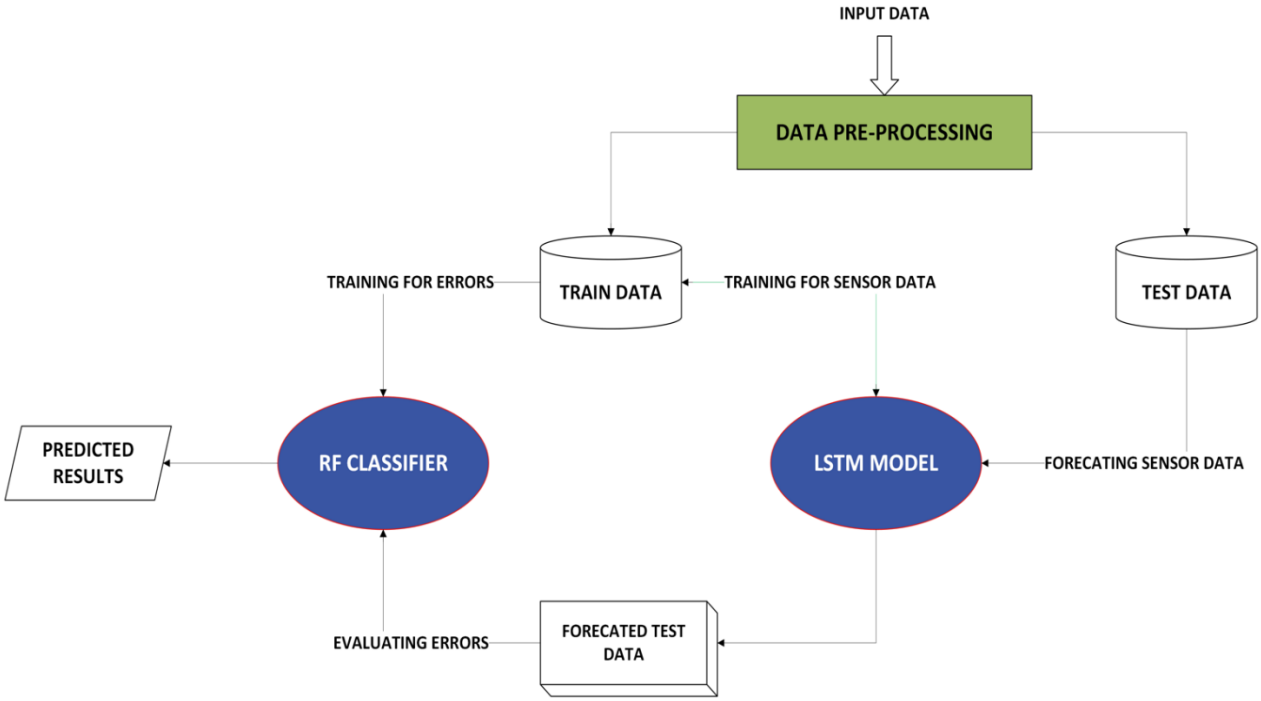


Figure 1. Architecture of the Proposed Forecasted Model

3.1 Datasets

The data used in the study were obtained from the OKUMAN company. The dataset comprises real sensor readings obtained from neonatal incubators developed and maintained by the company. Although the dataset is proprietary, it can be made available for academic use upon reasonable request to the authors or the company. This research, 13 variables were used as input parameters for the prediction algorithm, excluding date and alarm output values. As seen in Table 1, The dataset used contains 279,671 data points. The data were acquired from a series of sensors in the incubator, as well as information related to the system state.

Table 1. Incubator Sensor Data Properties

Serial No	Property Name	Description
1	Set Temperature	Indicates the set temperature of the incubator for the baby. The temperature range is determined to maintain the baby's comfort and health.
2	Air Temperature 1	Data was obtained from a sensor measuring the air temperature inside the incubator.
3	Air Temperature 2	Data from a second sensor measuring the air temperature inside the incubator.
4	Air 1 Temperature Data (Raw)	Raw data from the first air temperature sensor.
5	Air 2 Temperature Data (Raw)	Raw data from the second air temperature sensor.
6	Skin Temperature 1	First, skin temperature is measured from the baby's skin.
7	Skin Temperature 2	Second, skin temperature is measured on the baby's skin.
8	Heater Temperature	Indicates the temperature of the heating element inside the incubator.
9	Heater Power Percentage	Percentage value at which the heater power is set.
10	Heater Current	Electric current passes through the heater to generate heat.
11	Oxygen Percentage	Data was obtained from a sensor measuring the oxygen level inside the incubator.
12	Fan Current	Electric current passes through the fan motor inside the incubator.
13	Fan Speed	Indicates the rotational speed of the fan inside the incubator.
14	Date-Time	Date and time information when the data were recorded.
15	System Alarms	Alarm information indicating possible errors or deviations related to the incubator system.

3.2 Data Preparation

The dataset was prepared sequentially, undergoing data cleaning, splitting, and saving the data to make it suitable for analysis. In the data cleaning stage, unnecessary or irrelevant data that does not contribute to the prediction process was initially removed. The dataset was then divided into input and output data using specific input and output step parameters, ensuring the data was appropriately prepared for the model's training. Additionally, each month's dataset segments were merged to create a single dataset, mitigating the impact of gaps and missing data. The prepared datasets were saved for future use, allowing for the data's reuse and enhancing the results' reproducibility.

To minimize the impact of time and prevent negative effects on predictions, the dataset was arranged chronologically and grouped on a time basis. While 80% of the dataset was used for training, the remaining 20% was reserved for testing.

3.3 Long Short-Term Memory (LSTM) Algorithm

LSTM (Long Short-Term Memory) is one of the fundamental deep learning algorithms used in this study to predict the future health status of premature babies. LSTM is known for its resistance to the vanishing and exploding gradient problems that arise in traditional Recurrent Neural Network (RNN) models. This feature makes it ideal for modeling long-term dependencies, making it suitable for analyzing complex relationships over time, such as time series. LSTM cells comprise four main components: the forget gate, input gate, memory cell, and output gate. These components are essential for the network to forget past information, incorporate new information, and generate output. Below are the key equations describing the behavior of these gates:

Forget Gate (ft):

$$\boldsymbol{ft = \sigma(Wf \cdot [ht - 1, xt] + bf)} \tag{1}$$

Input Gate (it):

$$\boldsymbol{it = \sigma(Wi \cdot [ht - 1, xt] + bi)} \tag{2}$$

Updated Memory (C_t):

$$\boldsymbol{Ct = ft \cdot Ct - 1 + it \cdot \tanh(WC \cdot [ht - 1, xt] + bC)} \tag{3}$$

Output Gate (o_t):

$$\boldsymbol{ot = \sigma(Wo \cdot [ht - 1, xt] + bo)} \tag{4}$$

Cell Output (h_t):

$$\boldsymbol{ht = ot \cdot \tanh(Ct)} \tag{5}$$

Here, x_t represents the input data , h_t is the cell's output, C_t is the memory cell, W_f , W_i , W_C ,W_o and , b_f , b_i , b_C , b_0 represent the learned weights and biases. σ denotes the sigmoid function, and tanh represents the hyperbolic tangent function. As seen in Equation 1, the forget gate determines which information is discarded from the cell state. The input gate's function, as seen in Equation 2, involves determining which values from the input should be updated to the cell state. Equations 3 and 4 describe how the memory cell is updated and how the output is calculated.

The LSTM algorithm has been customized to analyze the dataset's time series and sensor data. These analyses rely on data obtained from various sensors in the baby incubator and information related to the system's status. The model has been trained on an 80% data slice and tested on the remaining 20%. The predictive capabilities of LSTM are aimed at accurately forecasting future alarm situations based on sensor data within a specific time interval. This algorithm provides a valuable tool for monitoring the health of premature babies and intervening when necessary.

3.4 The Random Forest (RF) Algorithm

The Random Forest (RF) algorithm is another significant machine-learning technique employed in this study to predict the health status of premature babies.

Random Forest is a widely used learning algorithm in machine learning, and it does not have a specific general formula; its fundamental structure is based on decision trees. RF creates multiple decision trees by randomly sampling from the dataset. Through this sampling method, each tree is trained on a different subset. A decision tree is created for each random sample. These trees assign data points to specific classes or values using the features in the dataset, learning complex relationships in the data. For classification problems, the class is determined by a voting process of predictions from all generated decision trees. For regression problems, the final prediction is made by averaging the predictions from the trees [59]. Although the performance of the random forest classifier surpasses individual decision trees, it heavily depends on the structure of the dataset [60], [61]. Nevertheless, the classifier requires minimal configuration and can make reasonable predictions across a broad range of data.

This study uses the RF algorithm to analyze sensor data and time series. It focuses on predicting future alarm situations within a specific time frame by processing information from different sensors in the dataset.

3.5 Performance Metrics

To evaluate the performance success of the LSTM model in the study, the following metrics have been employed: MSE (Mean Squared Error), MAE (Mean Absolute Error), and MAPE (Mean Absolute Percentage Error).

Mean Squared Error (MSE):

$$\boldsymbol{MSE} = \frac{\mathbf{1}}{\boldsymbol{n}}\sum_{\boldsymbol{n=1}}^{\boldsymbol{n}}(\boldsymbol{y^i} - \boldsymbol{\hat{y}^i})^2 \tag{6}$$

MSE assesses predictions made in a single step. It is calculated by subtracting the observed value from the predicted value, squaring this difference, summing all squared values, and then dividing by the number of observations. As seen in Equation 6, this metric evaluates the average of the squares of the errors, indicating how close a regression line is to a set of points.

Mean Absolute Error (MAE):

$$\boldsymbol{MAE} = \frac{\mathbf{1}}{\boldsymbol{n}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}\left|\boldsymbol{y^i} - \boldsymbol{\hat{y}^i}\right| \tag{7}$$

MAE measures the average magnitude of the errors in a set of predictions without considering their direction. As seen in Equation 7, it averages the absolute differences between observed and predicted values.

MAPE (Mean Absolute Percentage Error):

$$\boldsymbol{MAPE} = \frac{\mathbf{1}}{\boldsymbol{n}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}\left(\frac{\left|\boldsymbol{y^i} - \boldsymbol{\hat{y}^i}\right|}{\left|\boldsymbol{y^i}\right|}\right)\boldsymbol{X100} \tag{8}$$

MAPE expresses accuracy as a percentage, which is calculated by taking the average of the absolute percentage errors by the number of observations. As seen in Equation 8, this metric is useful for understanding the accuracy of the prediction in percentage.

The accuracy value was used to evaluate the performance of the Random Forest algorithm. Accuracy is a metric that measures the accuracy rate of a classification model. Generally, it expresses the ratio of correctly predicted instances to the total instances, as seen in Equation 9. "Correct Predictions" represents the number of instances correctly classified by the model, and "Total Number of Samples" indicates the total number of evaluated instances. Thus, accuracy shows the ratio of correctly predicted instances to the total number of instances and is usually expressed as a percentage (%). For example, 80% accuracy indicates that the model correctly classified 80% of the instances. Accuracy is a good performance measure for balanced classes or equal class distributions. This metric is used to assess the overall performance of the model.

$$\textbf{Accuracy} = \frac{\textbf{Correct Predictions}}{\textbf{Total number of samples}} \tag{9}$$

4. Experiments and Results

All models used in this study were compiled on the Google Colab platform with GPU support. The open-source deep learning library Keras, a high-level Python language API, was used for all codes. Data processing was conducted using the Pandas and NumPy libraries.

4.1 Data Preparation for the Model

Initially, the dataset is grouped according to a specific column in the study. The original dataset is grouped monthly, and monthly sub-data frames are obtained from these groups. This is a common practice when working with time series data, aiming to analyze and model the data based on a specific period. Training data are obtained from a comprehensive dataset consisting of 13 different sensor parameters. These sensors include measurements taken from the incubator, as detailed in Section 3.1. The dataset consists of time series representing periods of 20 seconds each.

4.2 Training the LSTM Model and Error Prediction with the RF Algorithm

The input sequence determined for the LSTM model, a sequential series where each period is 20 seconds, has predicted states at the 21st and 22nd seconds. In other words, the shape of the input sequence is set as (20, 2). Sequential data processing models like LSTM can analyze such sequences and be used in tasks such as predicting future values or predicting the next period in a sequential series.

The labels for the Random Forest (RF) classifier were obtained from the "System Alarms" field within the dataset, which records predefined error conditions generated by the incubator system. A total of 11 distinct error classes were identified and utilized as output labels, including conditions such as high temperature, low temperature, sensor malfunction, heater failure, and oxygen imbalance. Each alarm type was encoded as a categorical variable to facilitate multi-class classification. Due to the imbalanced nature of the label distribution, appropriate measures were considered during the performance evaluation of the classification model.

During the initial prediction data acquisition, the LSTM layer structure contains 300 neurons, and the 'relu' activation function is used. The LSTM model is compiled with the 'Adam' optimizer and mean squared error (MSE) loss. Figure 2 provides a convergence chart showing the mean squared error (MSE) values on the training and validation datasets during the training of this LSTM model. The chart monitors the model's performance on the training and validation datasets during training. If the validation error rises while the training error drops, it may indicate overfitting, suggesting a decrease in the model's generalization ability. A reasonable balance between training and validation errors is sought. Therefore, the early stopping technique is applied in model training in this study, and training is stopped if the validation error does not decrease throughout 10 epochs.

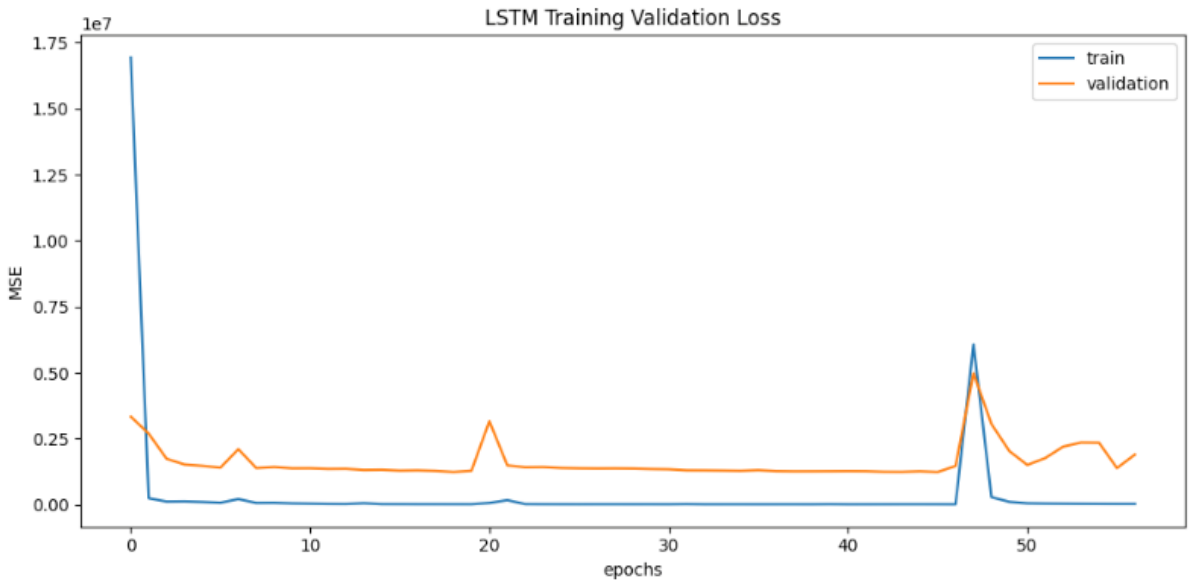


Figure 2. LSTM Training Graph

Later, experiments were repeated by changing the hyperparameters of the LSTM model to create the best-performing training conditions. Table 2 shows the hyperparameters of the LSTM model and their respective value ranges.

Table 2. LSTM Model Hyperparameters and Search Ranges

Model Hyperparameter Name	Search Range for Optimal Hyperparameter
Number of Epoch	[50, 75, 100]
Activation Function	[relu, sigmoid, tanh, softmax, linear]
Optimization Algorithm	[adam, RMSProp, AdaDelta]
Loss Function	[mse, binary_crossentropy]
Batch Size	[256]
Number of LSTM Units	[200, 250, 300, 400]
Input Number for Prediction	[10, 20, 40]

Then, the classification process with the RF algorithm was performed using the data predicted by the LSTM model. The classifier's predictions were compared with the errors of the values that the LSTM model should provide, and the model's success was determined. The parameter values used in the experimental studies and the performance results of the models are shown in Table 3. For example, in Table 3, it can be observed that an LSTM model trained with 256 units, 'relu' activation function, 100 epochs, and 'adam' optimizer has an MAE value of 163.2, MAPE value of 6.10 x 10^15, and MSE value of 11640.2. At the same time, the RF algorithm achieves 76.81% accuracy. The table includes important parameters such as the number of epochs, optimizer type, input sequence step, output step, and metrics used (MAE, MAPE, MSE).

Table 3. Comparison of LSTM and RF Model Performance with Various Hyperparameter Configurations

									LSTM			RF
Test	**Batch size**	**units**	**activation**	**epochs**	**optimizer**	**loss**	**In**	**Out**	**MAE**	**MAPE**	**MSE**	**Accuracy**
1	256	200	relu	100	adam	mse	20	2	163.2	6.10 x 10^15	11640.2	%76,81
2	256	200	relu	50	adam	mse	20	2	241.3	9.16 x 10^15	15830.5	%85,16
3	256	200	relu	75	adam	mse	20	2	114.0	7.03 x 10^15	94030.4	%80,79
4	256	200	relu	120	adam	mse	20	2	140.82	1.40 x 10^15	39590.8	%78.14
5	256	200	relu	50	RMSProp	mse	20	2	210.0	6.20 x 10^15	11910.9	%46.12
6	256	200	relu	50	AdaDelta	mse	20	2	5004.8	1.16 x 10^15	68040.0	%38.61
7	128	200	relu	50	AdaDelta	mse	20	2	4076.2	4.49 x 10^15	44840.0	%37.95
8	128	200	relu	50	adam	mse	20	2	153.8	5.99 x 10^15	13210.4	%70.13
9	256	200	sigmoid	50	adam	mse	20	2	6167.7	1.06 x 10^15	35059.0	%50.0
10	256	200	tanh	50	adam	mse	20	2	6206.9	1.07 x 10^15	34764.0	%11.16
11	256	200	Softmax	50	adam	mse	20	2	5206.9	1.06 x 10^15	24764.0	%20.16
12	256	200	linear	50	adam	mse	20	2	516.7	4.45 x 10^15	1858793.5	%21.84
13	256	200	relu	50	adam	b_cr	20	2	4182.2	9.47 x 10^15	24360.0	%11.16
14	256	300	relu	50	adam	mse	20	2	**160.8**	**9.01 x 10^15**	**1540533.6**	**%86.44**
15	256	400	relu	50	adam	mse	20	2	147.4	6.18 x 10^15	846971.94	%72.56
16	256	350	relu	50	adam	mse	20	2	225.5	4.52 x 10^15	1157109.9	%56.29
17	256	250	relu	50	adam	mse	20	2	181.9	7.44 x 10^15	1421834.6	%56.93
18	256	300	relu	50	adam	mse	40	2	-4.61	8.95 x 10^15	3214528.8	%69.36
19	256	300	relu	50	adam	mse	20	4	230.6	2.25 x 10^15	1362678.9	%63.5
20	256	300	relu	50	adam	mse	10	2	121.8	5.11 x 10^15	1228995.9	%51.3

By comparing the performance of models trained with different hyperparameters, the results of experiments shed light on determining the most effective hyperparameter combinations to optimize the prediction of the incubator conditions. As seen from Table 3, the RF model achieved the best performance with 86.44% accuracy when trained with an LSTM model with 256 units, 'relu' activation function, 50 epochs, 'adam' optimizer, and 'mse' loss, yielding MAE of 160.8, MAPE of 9.01 x 10^15, and MSE of 1540533.6.'

The LSTM model attempts to understand patterns in time series data using learned features and relationships. After processing and learning from the data, this model can predict the results at each step. Following the predictions of the LSTM model, an

RF classifier model comes into play. This model takes the outputs of the LSTM and classifies the states at each time step into specific alarm conditions. In other words, based on the outputs of the LSTM, it classifies the state at each time and predicts alarm conditions.

Finally, a confusion matrix is used to evaluate the performance of the RF model. The confusion matrix is a matrix that contains the numbers of correct and incorrect classifications by the model. As seen in Figure 3, the matrix displays the true and predicted values for alarm conditions within the test dataset. This matrix is visualized with a heatmap, allowing a visual understanding of which alarm conditions the model predicts better or worse. The heatmap is a colored matrix representation, providing insights to understand and improve the model's performance.

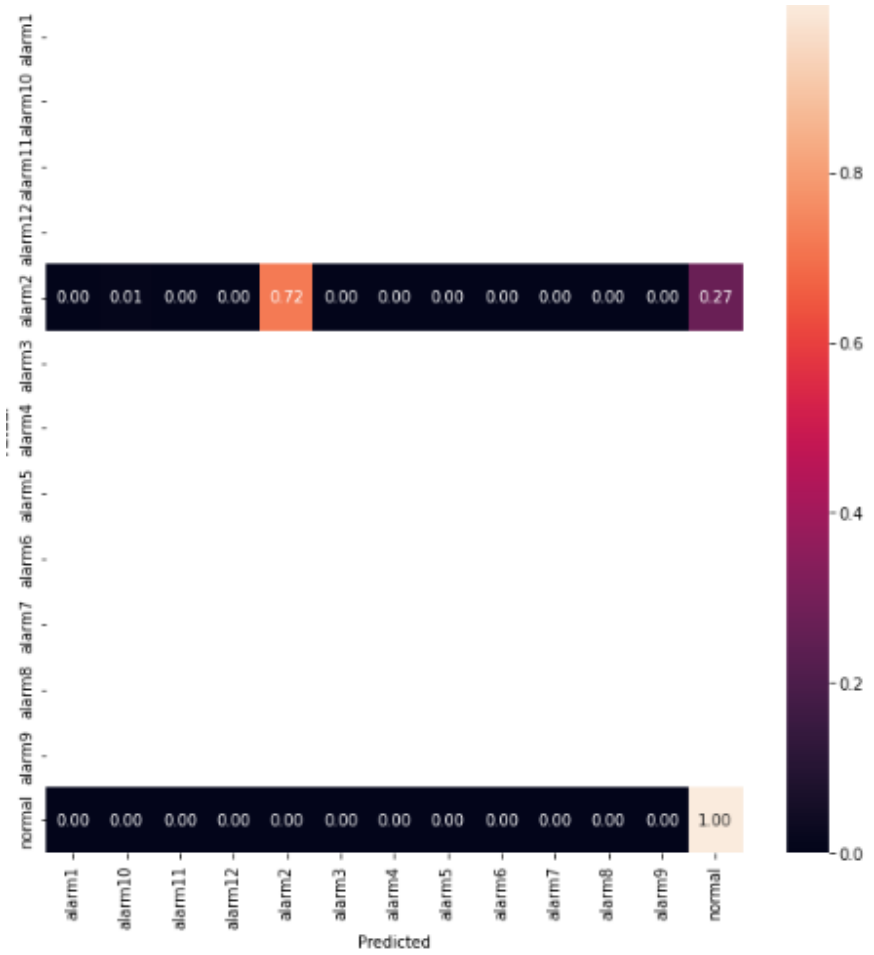


Figure 3. Confusion Matrix

5. Results and Recommendations

This study highlights the significant potential of artificial intelligence and deep learning techniques in neonatal

care. The results demonstrate that the LSTM-RF hybrid model is an effective tool for assessing the health status of premature infants. The algorithm successfully identified important patterns that could be associated with the health conditions of infants using sensor data.

The successful performance metrics of the RF algorithm, such as accuracy, sensitivity, and specificity, suggest that this model could be reliably used in neonatal care. However, it's essential to note that these successes were typically achieved on limited datasets, and further research is needed to explore the generalizability potential in different hospital environments.

Among the study's limitations are the limited scope of the dataset and the lack of information on infant sensor data. This limitation might restrict the model's application to a broader context. However, future studies incorporating larger datasets and additional sensor data could enhance the model's reliability and generalizability.

In conclusion, this study has shown that the accurate configuration of the model significantly influences the effectiveness of artificial intelligence models in neonatal care. Future research conducted on larger datasets and in different hospital environments will help us better understand how effective these models are in clinical applications.

References

[1] Goldenberg, R. L., & Rouse, D. J. (1998). Prevention of premature birth. *New England Journal of Medicine*, *339*(5), 313-320.

[2] Kumar, V., Shearer, J. C., Kumar, A., & Darmstadt, G. L. (2009). Neonatal hypothermia in low resource settings: a review. *Journal of perinatology*, *29*(6), 401-412

[3] Lunze, K., & Hamer, D. H. (2012). Thermal protection of the newborn in resource-limited environments. *Journal of Perinatology*, *32*(5), 317-324.

[4] H. Elayan, M. Aloqaily, and M. Guizani, "Digital Twin for Intelligent Context-Aware IoT Healthcare Systems," *IEEE Internet of Things Journal*, vol. 8, no. 23, pp. 16749–16757, Dec. 2021, doi: 10.1109/JIOT.2021.3051158.

[5] Y. Liu *et al*., "A Novel Cloud-Based Framework for the Elderly Healthcare Services Using Digital Twin," in *IEEE Access*, vol. 7, pp. 49088-49101, 2019, doi: 10.1109/ACCESS.2019.2909828

[6] Erol, T., Mendi, A. F., & Doğan, D. (2020, October). The digital twin revolution in healthcare. In *2020 4th international symposium on multidisciplinary studies and innovative technologies (ISMSIT)* (pp. 1-7). IEEE.

[7] A. Haleem, M. Javaid, R. Pratap Singh, and R. Suman, "Exploring the revolution in healthcare systems through the applications of digital twin technology," *Biomedical Technology*, vol. 4, pp. 28–38, Dec. 2023, doi: 10.1016/j.bmt.2023.02.001.

[8] M. Peshkova, V. Yumasheva, E. Rudenko, N. Kretova, P. Timashev, and T. Demura, "Digital twin concept: Healthcare, education, research," *Journal of Pathology Informatics*, vol. 14, p. 100313, Jan. 2023, doi: 10.1016/j.jpi.2023.100313.

[9] Y. Han, Y. Li, Y. Li, B. Yang, and L. Cao, "Digital twinning for smart hospital operations: Framework and proof of concept," *Technology in Society*, vol. 74, p. 102317, Aug. 2023, doi: 10.1016/j.techsoc.2023.102317.

[10] "A digital twin framework for prognostics and health management," *Computers in Industry*, vol. 150, p. 103948, Sep. 2023, doi: 10.1016/j.compind.2023.103948.

[11] R. Aluvalu, S. Mudrakola, U. M. V, A. C. Kaladevi, M. V. S. Sandhya, and C. R. Bhat, "The novel emergency hospital services for patients using digital twins," *Microprocessors and Microsystems*, vol. 98, p. 104794, Apr. 2023, doi: 10.1016/j.micpro.2023.104794.

[12] M. J. Kaur, V. P. Mishra, and P. Maheshwari, "The convergence of digital twin, IoT, and machine learning: transforming data into action," *Digital twin technologies and smart cities*, pp. 3–17, 2020.

[13] J. Kumari, R. Karim, K. Karim, and M. Arenbro, "MetaAnalyser - A Concept and Toolkit for Enablement of Digital Twin," *IFAC-PapersOnLine*, vol. 55, no. 2, pp. 199–204, Jan. 2022, doi: 10.1016/j.ifacol.2022.04.193.

[14] I. Kononenko and M. Kukar, *Machine learning and data mining*. Horwood Publishing, 2007.

[15] Attaran, M., & Celik, B. G. (2023). Digital twin: Benefits, use cases, challenges, and opportunities. *Decision Analytics Journal, 6*, 100165. https://doi.org/10.1016/j.dajour.2023.100165

[16] Dassault Systèmes. (n.d.). *The Living Heart Project*. https://www.3ds.com/products-services/simulia/solutions/life-sciences-healthcare/the-living-heart-project/

[17] A. Manocha, Y. Afaq, and M. Bhatia, "Digital Twin-assisted Blockchain-inspired irregular event analysis for eldercare," *Knowledge-Based Systems*, vol. 260, p. 110138, Jan. 2023, doi: 10.1016/j.knosys.2022.110138.

[18] Wahab, S. M. A. A., & Saad, M. (2022). Digital transformation acceleration in health sector during COVID-19: Drivers and consequences. *Journal of Business and Management Sciences*, *10*(4), 164-179.

[19] C. Quilodrán-Casas, V. L. S. Silva, R. Arcucci, C. E. Heaney, Y. Guo, and C. C. Pain, "Digital twins based on bidirectional LSTM and GAN for modelling the COVID-19 pandemic," *Neurocomputing*, vol. 470, pp. 11–28, Jan. 2022, doi: 10.1016/j.neucom.2021.10.043.

[20] D. Chen, N. A. AlNajem, and M. Shorfuzzaman, "Digital twins to fight against COVID-19 pandemic," *Internet of Things and Cyber-Physical Systems*, vol. 2, pp. 70–81, Jan. 2022, doi: 10.1016/j.iotcps.2022.05.003.

[21] Q. Lv, R. Zhang, X. Sun, Y. Lu, and J. Bao, "A digital twin-driven human-robot collaborative assembly approach in the wake of COVID-19," *Journal of Manufacturing Systems*, vol. 60, pp. 837–851, Jul. 2021, doi: 10.1016/j.jmsy.2021.02.011.

[22] H. Neog, P. E. Dutta, and N. Medhi, "Health condition prediction and covid risk detection using healthcare 4.0 techniques," *Smart Health*, vol. 26, p. 100322, Dec. 2022, doi: 10.1016/j.smhl.2022.100322.

[23] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," *Advances in neural information processing systems*, vol. 28, 2015.

[24] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[25] C. Quilodrán-Casas, R. Arcucci, C. Pain, and Y. Guo, "Adversarially trained LSTMs on reduced order models of urban air pollution simulations," *arXiv preprint arXiv:2101.01568*, 2021.

[26] A. Elsheikh, S. Yacout, and M.-S. Ouali, "Bidirectional handshaking LSTM for remaining useful life prediction," *Neurocomputing*, vol. 323, pp. 148–156, Jan. 2019, doi: 10.1016/j.neucom.2018.09.076.

[27] G. Liu and J. Guo, "Bidirectional LSTM with attention mechanism and convolutional layer for text classification," *Neurocomputing*, vol. 337, pp. 325–338, Apr. 2019, doi: 10.1016/j.neucom.2019.01.078.

[28] O. Yeler and M. F. Koseoglu, "Performance prediction modeling of a premature baby incubator having modular thermoelectric heat pump system," *Applied Thermal Engineering*, vol. 182, p. 116036, 2021.

[29] R. Cuervo, M. A. Rodríguez-Lázaro, R. Farré, D. Gozal, G. Solana, and J. Otero, "Low-cost and open-source neonatal incubator operated by an Arduino microcontroller," *HardwareX*, vol. 15, p. e00457, Sep. 2023, doi: 10.1016/j.ohx.2023.e00457.

[30] P. T. Kapen, Y. Mohamadou, F. Momo, D. K. Jauspin, N. Kanmagne, and D. D. Jordan, "Development of a neonatal incubator with phototherapy, biometric fingerprint reader, remote monitoring, and heart rate control adapted for developing countries hospitals," *Journal of Neonatal Nursing*, vol. 25, no. 6, pp. 298–303, Dec. 2019, doi: 10.1016/j.jnn.2019.07.011.

[31] A. Hannouch, T. Lemenand, K. Khoury, and C. Habchi, "Heat and Mass Transfer of Preterm Neonates Nursed inside Incubators-A Review," *Thermal Science and Engineering Progress*, p. 100553, 2020.

[32] V. Puyana-Romero *et al.*, "Reverberation time measurements of a neonatal incubator," *Applied Acoustics*, vol. 167, p. 107374, 2020.

[33] A. Chandrasekaran *et al.*, "Disposable low-cost cardboard incubator for thermoregulation of stable preterm infant – a randomized controlled non-inferiority trial," *EClinicalMedicine*, vol. 31, p. 100664, Jan. 2021, doi: 10.1016/j.eclinm.2020.100664.

[34] A. Fraguela, F. D. Matlalcuatzi, and Á. M. Ramos, "Mathematical modelling of thermoregulation processes for premature infants in closed convectively heated incubators," *Computers in biology and medicine*, vol. 57, pp. 159–172, 2015.

Author(s) Contributions

This article was collaboratively written by three authors who contributed to the research and manuscript preparation as follows:

Hatice Kabaoğlu, Conceptualization, Writing – Original Draft, Software Development, Data Analysis. Fecir Duran, Supervision, Methodology, Conceptual Guidance, Technical Evaluation. Emine Uçar, Software Verification, Data Support, Writing – Review & Editing. All authors have read and approved the final version of the manuscript.

Conflict of Interest Notice

There is no conflict of interest regarding the publication of this paper.

Support/Supporting Organizations

The dataset used in this study was provided by OKUMAN Health Company (https://okuman.com.tr/). No additional institutional or financial support was received for this research.

Ethical Approval and Informed Consent

In this section, the author(s) declaration regarding research and publication ethics will be included. This title must be included for all articles. The name, date and number of the Ethics Committee can be given in this section.

It is declared that during the preparation process of this study, scientific and ethical principles were followed, and all the studies benefited from are stated in the bibliography.

Artificial Intelligence Statement

No AI tools were used to generate original research content, data analysis, or interpretation.

Availability of data and material

Not applicable / or link

Plagiarism Statement

This article has been scanned by iThenticate ™

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
http://saucis.sakarya.edu.tr/
e-ISSN: 2636-8129
Publisher: Sakarya University
Vol. 8, No. 2, 312-321, 2025
DOI: 10.35377/saucis..1582098
SAUCIS
SAKARYA UNIVERSITY

RESEARCH ARTICLE

Diagnosis of Lichen Sclerosus, Morphea, and Vasculitis Using Deep Learning Techniques on Histopathological Skin Images

Recep Güler[1], Zehra Karapınar Şentürk[2*], Mehmet Gamsızkan[3], Yunus Özcan[4]

[1] Düzce Vocational School, Düzce University, Düzce, Türkiye, ror.org/04175wc52

[2] Department of Computer Engineering, Düzce University, Düzce, Türkiye, ror.org/04175wc52

[3] Department of Pathology, Faculty of Medicine, Düzce University, Düzce, Türkiye, ror.org/04175wc52

[4] Department of Dermatology, Faculty of Medicine, Düzce University, Düzce, Türkiye, ror.org/04175wc52

Corresponding author:
Zehra Karapınar Şentürk,
Department of Computer Engineering,
Düzce University, Düzce, Türkiye
zehrakarapinar@duzce.edu.tr

Article History:
Received: 10.11.2024
Revised: 15.01.2025
Accepted: 12.04.2025
Published Online: 20.06.2025

ABSTRACT
Skin diseases are very common all over the world. The examination can be done by photographing the relevant area or taking a tissue sample to diagnose skin diseases. Examining tissue samples allows examination at the cellular level. This study discussed three skin diseases: lichen sclerosus, morphea, and cutaneous small vessel vasculitis (vasculitis). For this problem, which does not have an open-access dataset in the literature, a dataset consisting of histopathological images belonging to each class was created. Convolutional neural network models were created for this three-class classification problem, and their results were evaluated. In addition, in this problem where it is difficult to obtain sample images, the efficiency of transfer learning methods was evaluated with a limited number of examples. For this purpose, tests were performed with VGG16, ResNet50, InceptionV3, and EfficientNetB4 models, and the results were given. Among all the results, the accuracy value of the VGG16 model was 0.9755 and gave the best result. However, although the accuracy value was quite good, precision, recall, and f1-score metrics values were around 0.65. This shows deficiencies in how often the model correctly predicts the positive class and how well it predicts all positive examples in the dataset.

Keywords: Convolutional neural networks, Data augmentation, Transfer learning, Histopathology

1. Introduction

Skin diseases are the fourth most common non-fatal disease in 188 countries worldwide [1]. The frequency of skin diseases is increasing with the extension of human life and the effects of modern living conditions. In addition, many skin diseases are chronic and require long treatment periods and multiple physician check-ups. Considering the scarcity of trained specialist physicians, it is inevitable that there will be inadequacies and disruptions in the provision of health services for skin diseases from time to time.

Although the same skin disease can manifest itself in many different ways, most skin diseases can be diagnosed by a specialist at the first examination. However, when specialists have difficulty diagnosing, blood tests that will help are limited. At this point, the methods that help in making a diagnosis are histopathological examinations. Histopathology is the examination of tissue samples by an expert pathologist [2].

In these examinations, a small specimen taken from the skin tissue where the disease is located is examined under a microscope after a series of processes. The tissue architecture formed by the cells and their association with the changes that develop in these due to the diseases are determined. Experts have analyzed hematoxylin and eosin (H&E) stained sections of tissue samples, and it has emerged as the most effective method for examining histopathological images in the last century [3]. At this stage, a second bottleneck emerges in the health system, and service disruptions occur occasionally. The field of work of the pathology physicians who perform these examinations is wide, while the number of experienced physicians who focus on skin diseases is small. The decision support systems to be developed at this stage will reduce these disruptions.

Cite as: R. Güler, Z. Karapınar Şentürk, M. Gamsızkan & Y. Özcan. (2025) Diagnosis of Lichen Sclerosus, Morphea, and Vasculitis Using Deep Learning Techniques on Histopathological Skin Images. *Sakarya University Journal of Computer and Information Sciences*. vol. 8, no. 2, pp. 312-321, 2025 doi: 10.35377/saucis...1582098

Because of the similar appearance of skin diseases, it is not easy to distinguish them by the human eye, and it takes a long time to train experts [4]. For this reason, image processing, segmentation and machine learning studies have been carried out on dermatological images as auxiliary systems. The studies were generally conducted using color photographs of the diseased area, and machine learning and deep learning methods were preferred in processes such as classification and segmentation. A classification study was conducted with EfficientNetB4 using 13603 images containing 14 different skin diseases obtained from a hospital in China [4]. In a different study where photographic images were used for 59 diseases, including vasculitis and morphea, deep learning (ResNet18) based classification was performed [5]. A dataset consisting of skin images of five different skin diseases was tested on popular CNN networks such as ResNet50, DenseNet, MobileNet, Xception and comparative results were shared [6]. Lesion detection and classification were performed in the study, and color images were used to classify dermatological diseases [7]. Some of these studies used existing online datasets, while others created them. Table 1 compares some of these studies, summarizing the dataset's characteristics, the deep learning method used for classification, and the results obtained with the corresponding performance metric.

In the literature, there are also those working on web-based or mobile applications to reduce the workload of physicians and ensure image processing and results are delivered directly to the user. With the developed smartphone application, a DenseNet161-based network was designed to classify 40 skin diseases, such as eczema, melanoma, and lichen sclerosus [1]. A different study presented a web application where users of 5 skin diseases could learn the predicted disease result by uploading skin photos [6]. However, the long processing time of the mentioned web application and the fact that it only gives results for a limited number of diseases can be stated as the points that need to be completed in the study [6].

Another method used in addition to dermatological photographs to examine skin diseases in more detail is histopathological examination of tissue samples. Experts can examine Histopathological images directly under a microscope, or images can be transferred to a computer using digital microscopes. Another way to transfer histopathological images to a computer is to use whole slide imaging (WSI) devices. Thus, images can be recorded at high resolution (100000x100000 pixels, etc.). After the positive effects of deep learning methods on objectivity and efficiency in cancer diagnosis with full slide imaging emerged, many studies have been conducted on diagnosing or segmenting diseases such as skin cancer, prostate cancer, lung cancer, etc. [3]. Due to the large size of the images obtained from full slide imaging, applying them directly as input to CNN networks is impossible. When the size of the images is reduced to a suitable resolution for CNN networks, since there will be losses in image features, extracting smaller patches from large images and using them as input increases efficiency [8], [9].

In convolutional neural networks (CNN) and deep learning models, the size of the dataset, its correct labeling and distribution are important for solving image classification problems with high accuracy. Especially since the datasets consisting of medical images are small and labeling the images is costly, the number of images in the dataset can be increased using data augmentation methods. It is known that data augmentation methods used in computer vision studies contribute positively to the overall performance and that the combined use of several data augmentation methods generally increases the performance [10], [13]. Data augmentation methods have been examined in different CNN networks for melanoma classification. It has been shown that the best results are obtained by using geometric and color transformation methods together [11]. Among the data augmentation methods, synthetic data creation also uses the features of the existing data. Experimental studies have been conducted on melanoma, histopathological images, and MRI images related to the study. A new method is presented to obtain new data by style transfer using the texture and color features of the images in the existing data set [12]. It has also been shown that overfitting can be prevented in CNN networks by using data augmentation methods [13], [14]. Data augmentation is usually performed on the training dataset, and as a result, accuracy, precision, sensitivity and F1-score values are expected to increase. In addition to expanding the dataset with data augmentation, class imbalances can also be eliminated.

Table 1. Selected deep learning-based skin image classification studies

Ref.	Dataset	Classifier	Performance measure
1	15418 images for 40 skin disease	DenseNet161	Overall accuracy 0.769
4	13603 images for 14 skin disease	Custom CNN based on EfficientNet-B4	Overall accuracy 0.948
5	70196 images for 59 skin disease	ResNet18	Overall accuracy 0.579
6	18692 images for 5 skin disease	ResNet50, Inception-V3, Inception-ResNet, DenseNet, MobileNet, Xception	MobileNet accuracy 0.960 Xception accuracy 0.979
7	505 images for 5 skin disease	DenseNet121	Overall accuracy 0.952

A generalized model and many images are needed to create a deep-learning model with high accuracy. The transfer learning method eliminates the need for computational resources and creates models that can produce faster results [6]. When machine learning-based studies in cancer diagnosis are examined, attempts are made to increase the prediction accuracy by using data augmentation and transfer learning methods. In classifying benign and malignant skin lesions with CNN networks, data preprocessing and data augmentation methods are used to remove image artifacts and correct imbalances in the data set [15]. ResNet, Inception, VGG, and DenseNet networks are frequently used in the transfer learning approach, and their results are compared. With the different architecture, filter size, and parameter number values of these deep learning networks, the

accuracy values obtained in medical image classification vary depending on the feature of the image [16], [17]. High accuracy values can be obtained with transfer learning methods to detect and classify skin and breast cancer [17], [18]. Learning processes can be accelerated with transfer learning methods in the segmentation of cancer tissue [19]. In addition, the transfer learning method is not used directly for classification but for feature extraction, and the obtained features can be used in a different CNN network [19], [20], [21].

This study uses histopathological images to present a computer-aided diagnosis of lichen sclerosus, morphea and vasculitis diseases. Different CNN network structures were trained and tested using medical image data augmentation and transfer learning. Unlike studies in the literature using skin image datasets, our study is one of the first studies in the field using histopathological images to classify related diseases using machine learning. The main contributions of this study are as follows:

- A new histopathological dataset was created to diagnose lichen, morphea and vasculitis diseases and deep learning was used for classification.
- The success of data augmentation and transfer learning methods was demonstrated on the dataset.

The rest of the paper is structured as follows: section 2 presents the dataset's characteristics, data preprocessing stages and the methods mentioned in the CNN networks used. In section 3, under the results and discussion, the results of the training and testing processes are presented in detail (accuracy, precision, sensitivity and F1-score and confusion matrix) in a table. Finally, in section 4, the results are presented in the conclusion.

2. Methods

The proposed method for classifying histopathological images consists of three main parts (Fig. 1). The first part is transferring the tissue sample to the digital environment and creating the data set. The second part is the resizing process to bring the images to the appropriate input size for the CNN network to be used with data preprocessing. Following this process, data sets consisting of different samples were created with data augmentation to see the effect of the data set size on success. In the last part, training and testing processes were performed with a custom CNN and deep learning models (ResNet, VGG, Inception, EfficientNet) found in the literature.

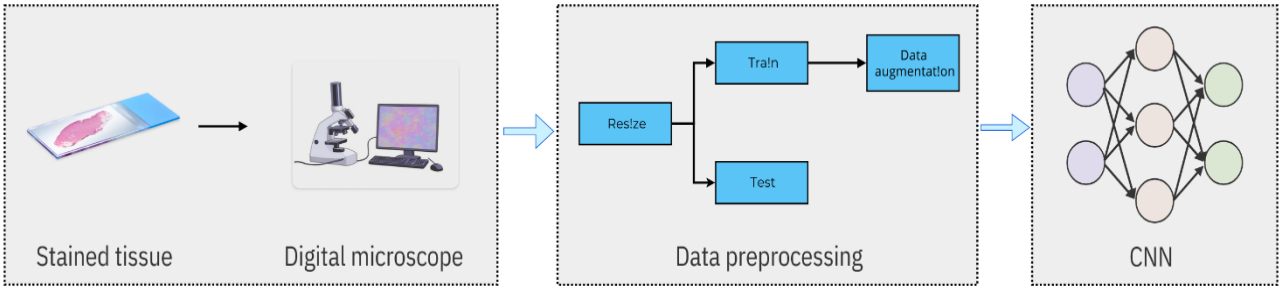


Figure 1. The general structure of the study

2.1 Dataset

The specimen obtained for histopathological examinations is first reduced to appropriate dimensions and embedded in paraffin. A 3-5 µm thick section is taken from this prepared specimen. This section, fixed on a glass called a slide, is stained using different dyes and is made ready for viewing under a microscope after a series of processes.

Usually, the first sample is prepared with hematoxylin-eosin stain, and subsequent stain selections are determined by the tissue being processed and the disease being investigated. When an examination with a different stain, a new section is taken from the tissue previously stored in paraffin and stained. Considering the size of the sections, the newly taken section usually looks similar to the previous one when viewed with the naked eye, but it is never the same.

The dyes used are usually determined by the region intended to be examined. Among the diseases we study, lichen sclerosus and morphea are diseases in which the structure of proteins called collagen is disrupted. The Masson Trichrome dye we use allows examination by staining areas where collagen is present more visibly. In patients with cutaneous vasculitis, since collagen is not affected, it is expected that no abnormal pattern will occur.

This study consists of 147 microscopic skin images of three diseases (lichen sclerosus, morphea, vasculitis) created by Düzce University Faculty of Medicine, Department of Pathology specialist doctors. After the tissue samples were stained with Masson Trichrome dye by the specialists, they were saved as 1920x1080 JPG files using a digital microscope. The distribution of diseases constituting the data set is given in Table 2. The images that comprise the data set do not contain personal information.

Table 2. Dataset properties

Disease	Number of Images
Lichen sclerosus	51
Morphea	42
Vasculitis	54

Convolutional neural networks are deep learning structures that consist of a multi-layered structure and require a large amount of input data to make the right decision. Although it is possible to access many different data sets from different fields of study today, it has been observed that medical data sets, in particular, are not sufficient in number or not open access. In addition, labeling images by experts in creating a medical data set is disadvantageous in terms of time and cost.

2.2 Data Preprocessing

The images that make up the dataset are large enough to be used in an artificial neural network model. Although it is possible to work with large-sized images and produce better results by preventing data loss, they are not preferred since they require a lot of processing power and have time-cost disadvantages. Smaller images, such as 224x224, are generally used in the literature. Thus, less processing power is required for matrix calculations. In addition, the squareness of the input image allows convolution, padding, and resizing operations to be performed more efficiently.

In image classification problems, the model's ability to achieve better results is directly related to the size of the data set. Using data augmentation methods, it aims to increase the numerical increase of the image in the data set and its diversity. In this way, in addition to better learning, the overfitting problem can be reduced [10]. It has been shown in studies that the use of data augmentation increases the performance of deep learning models. Data augmentation methods include geometric transformation, color space transformations, Kernel filters, image deletion, image fusion, style transfer, and GAN-based image generation [14].

In this study, geometric transformation-based data augmentation methods were used. Flipping is obtaining a mirror image of the image horizontally or vertically (Figure 2b). Rotating is rotating the image around itself by a certain degree (Figure 2c). Scaling can be applied by zooming in or out of the image (Figure 2d).

Since the microscopic histopathological images have large dimensions (1920x1080), they were resized during preprocessing. Resizing the images at this stage can also cause data loss. A total of four data sets were created: two data sets (A and B) to see the effect of image resizing on the performance of the CNN model used and two data sets (C and D) to see and compare the performance of data augmentation processes (Table 2). First, the data set was divided into 85% for training and 15% for testing. Then, the data allocated for training was divided again into 85% for training and 15% for validation during the training of the CNN model.

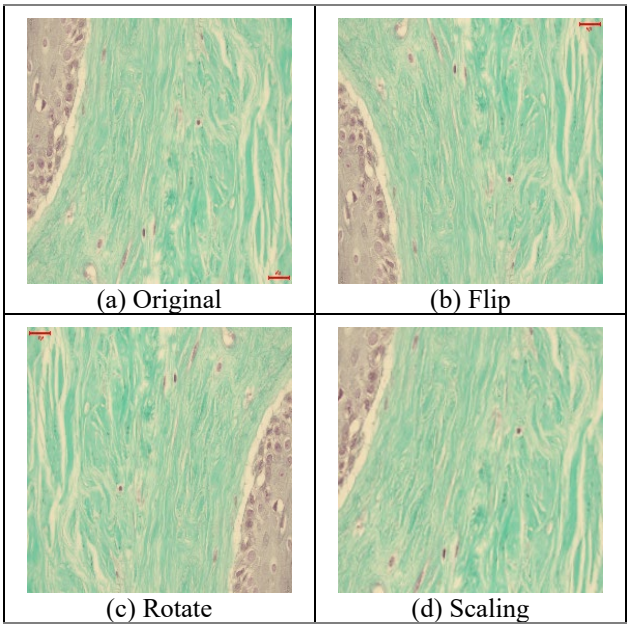


Figure 2. Geometric transformations

After separating the dataset into training and testing, the dataset clusters whose features are given in the Table 3 were created. Dataset A was created by resizing the original images to 256x256 pixels, and dataset B was created by resizing the original

images to 512x512 pixels. Then, the training data was increased approximately 5 and 10 times with the previously mentioned geometric data augmentation methods, and dataset clusters C and D were created, respectively.

Table 3. Created dataset clusters (LS=Lichen Sclerosus, M=Morphea, V=Vasculitis)

Dataset	Image Size	Training		Test		Total Number of Images
A	256x256	122	LS:42 M:35 V:45	25	LS:9 M:7 V:9	147
B	512x512	122	LS:42 M:35 V:45	25	LS:9 M:7 V:9	147
C	256x256	675	LS:225 M:225 V:225	25	LS:9 M:7 V:9	700
D	256x256	1350	LS:450 M:450 V:450	25	LS:9 M:7 V:9	1375

2.3. CNN Models

The number and success of deep learning studies are increasing day by day. Convolutional neural networks are a sub-element of deep learning, which is frequently used, especially in classification and image processing studies [8], [12]. Convolutional neural networks consist of convolution, pooling and fully connected layer sections. CNN models can be designed with different sizes and features to perform different tasks.

Deep learning models require large data sets and the ability to include quite complex operations to extract the most important and useful features from the data set for classification. One disadvantage of working with complex models and large data sets is that training a deep-learning model takes a long time. In medical image processing, there is usually a problem with small data sets and images not being labeled correctly. Transfer learning is used to overcome these problems in deep learning. Transfer learning uses knowledge from models trained on large datasets for different tasks and datasets [22], [23]. With transfer learning, models trained using large datasets are used especially to perform tasks with a small number of images [24]. AlexNet is one of the first large-scale convolutional neural network models developed. AlexNet won the ImageNet large-scale image recognition challenge in 2012, drawing attention to the success of CNN models in image processing [25]. In 2014, VGG and GoogLeNet (Inception) achieved better results on the ImageNet dataset with deeper CNN models [26], [27]. Considering that deeper networks are more difficult to train and that accuracy does not always increase even with increasing network depth, the ResNet model, which was developed, facilitated the training of deep networks and won the ImageNet challenge in 2015 [28]. These models were studied on the ImageNet dataset, which consists of 1,281,167 training images, 50,000 validation images, and 100,000 test images of 1000 object classes [29].

Table 4. Features of the created CNN models

CNN Model	Number of Convolution Layer	Activation Function	Total Parameter
Model 1	2	ReLU	134239171
Model 2	2	ReLU	134241603
Model 3	3	ReLU	33612803
Model 4	3	ReLU	31548419

This study created four CNN models to detect diseases from microscopic skin images. The features of these CNN models are given in Table 4. ReLU was used as each model's activation function, and the MaxPooling (2x2) operation was performed after the convolution process. In addition, the dropout functions used were used to prevent memorization during training. Finally, the classification result for three diseases was obtained with the SoftMax output layer.

Table 5. Features of the used pre-trained models

CNN Model	Number of Total Layer	Activation Function	Total Parameter
VGG16	16	ReLU	138.4M
ResNet50	50	ReLU	25.6M
InceptionV3	189	ReLU	23.9M
EfficientNetB4	258	Softmax	19.5M

Using the transfer learning method, VGG16, ResNet50, InceptionV3, and EfficentNetB4 pre-trained models were also tested on the same datasets. The features of the pre-trained networks used are given in Table 4.

3. Results and Discussion

The previously mentioned datasets were tested on a total of eight networks. The studies were performed with Keras and TensorFlow on Google Colab. The program was run for the first time for each network, and learning was performed for 50 epochs. While evaluating the networks, Accuracy, Precision, Recall (Sensitivity), and F1-score metrics were calculated.

Table 6. Test Results

Data Set	CNN Model	Validation Accuracy	Precision	Recall	F1 score
A	Model 1	0.5714	**0.70**	**0.64**	**0.64**
	Model 2	0.3333	0.13	0.36	0.19
	Model 3	0.4762	0.49	0.44	0.45
	Model 4	0.4286	0.13	0.36	0.19
	VGG16	0.7619	0.64	0.60	0.60
	ResNet50	**0.8095**	0.57	0.60	0.57
	InceptionV3	0.5238	0.39	0.28	0.28
	EfficientNetB4	0.7143	0.48	0.44	0.45
B	Model 1	0.5238	0.66	0.60	0.61
	Model 2	0.3333	0.13	0.36	0.19
	Model 3	0.5714	**0.78**	**0.72**	**0.70**
	Model 4	0.3810	0.13	0.36	0.19
	VGG16	**0.8095**	0.59	0.56	0.52
	ResNet50	**0.8095**	0.67	0.64	0.65
	InceptionV3	0.4762	0.32	0.40	0.34
	EfficientNetB4	0.6190	0.53	0.52	0.52
C	Model 1	0.5392	0.58	0.52	0.51
	Model 2	0.7647	0.32	0.44	0.37
	Model 3	0.6569	0.67	0.56	0.57
	Model 4	0.5392	0.50	0.48	0.48
	VGG16	0.9314	0.62	0.60	0.59
	ResNet50	**0.9608**	**0.81**	**0.80**	**0.79**
	InceptionV3	0.5784	0.41	0.40	0.35
	EfficientNetB4	0.8922	0.56	0.48	0.45
D	Model 1	0.6520	0.20	0.28	0.23
	Model 2	0.7892	0.56	0.52	0.46
	Model 3	0.4755	**0.72**	0.60	0.61
	Model 4	0.7598	0.48	0.48	0.46
	VGG16	**0.9755**	0.69	0.68	0.68
	ResNet50	0.9608	**0.72**	**0.72**	**0.71**
	InceptionV3	0.7157	0.52	0.52	0.51
	EfficientNetB4	0.9118	0.55	0.48	0.49

Precision, Recall, and F1-score values were calculated for the performance measurement of the test results. Accuracy gives the correct result rate among all measured results (Equation 1). Precision gives the proportion of correctly predicted among all correctly labeled examples (Equation 2). Sensitivity gives the correct prediction rate among all examples (Equation 3). The F1-score value is the harmonic mean of precision and sensitivity values (Equation 4).

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

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

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

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

The performance of each model on the created datasets is presented in Table 6. When the evaluation is made according to the validation process results, it is seen that for dataset A, the best accuracy value among the created CNN models belongs to Model 1 with 0.5714 and ResNet50 from the pre-trained networks with 0.8095. For dataset B, the best accuracy value among the created CNN models belongs to Model 3 with 0.5714 and ResNet50 from the pre-trained networks with 0.8095. Considering that no data augmentation was performed on datasets A and B, it is seen that the results obtained for the dataset containing a small number of images are not sufficient for the created CNN models. However, the results of the pre-trained networks were better than expected.

Table 5 shows that especially pre-trained networks have much better accuracy values for the C and D data sets created by applying data augmentation. For dataset C, the best accuracy value among the created CNN models belongs to Model 2 with 0.7642 and ResNet50 from the pre-trained networks with 0.9608. For dataset D, the best accuracy value among the created CNN models belongs to Model 2 with 0.7892 and VGG16 from the pre-trained networks with 0.9755.

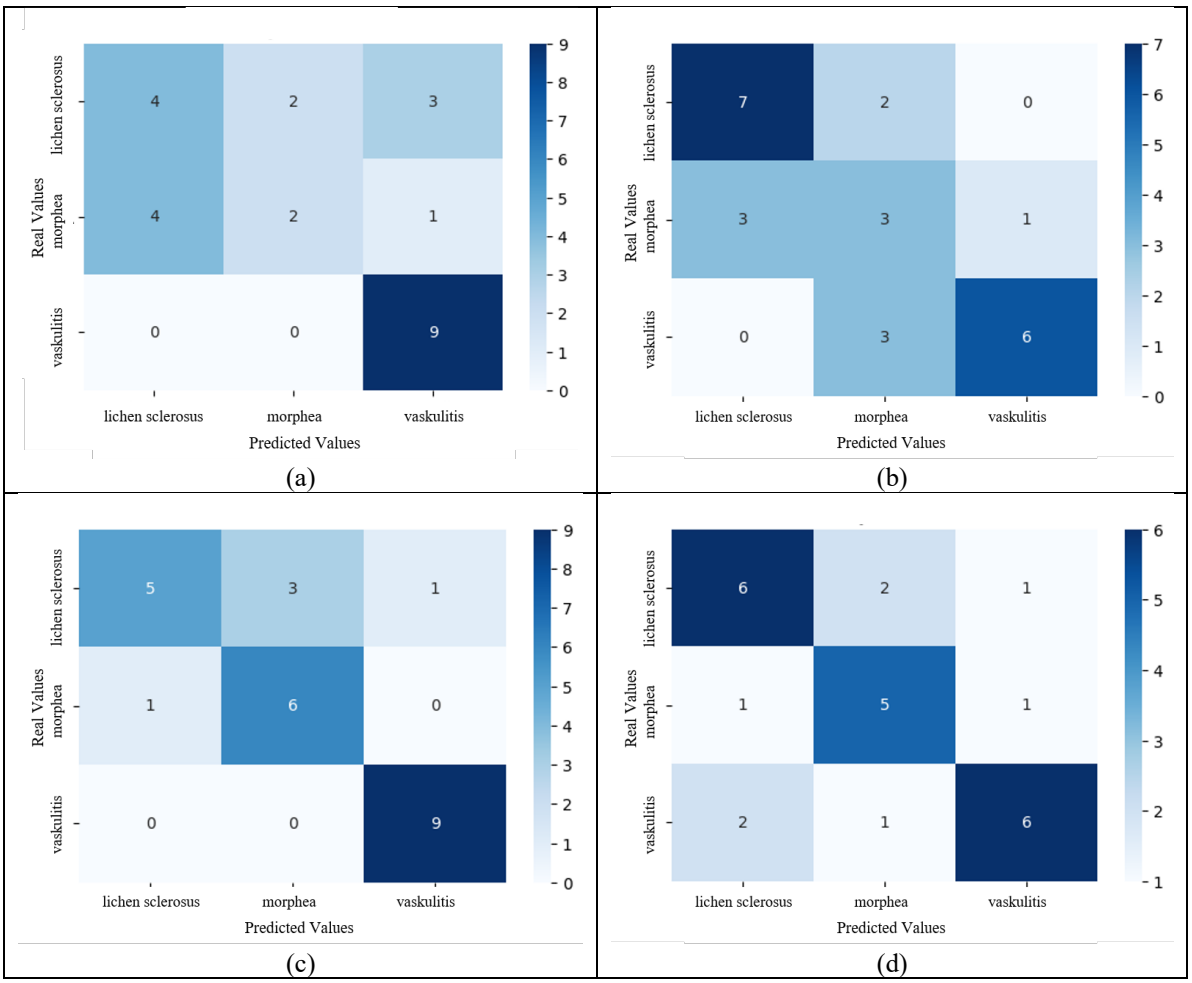


Figure 3. Confusion matrix for the best models concerning datasets A, B, C, and D

The model's performance can also be visualized by creating a Confusion Matrix based on the test results. Confusion matrices for the best models according to the datasets are given in Figure 3. The confusion matrices are given in Figure 3a for the ResNet50 model for dataset A, Figure 3b for the ResNet50 model for dataset B, Figure 3c for the ResNet50 model for dataset C, and Figure 3d for the VGG16 model for dataset D, respectively.

Although the accuracy values obtained due to the training and validation processes of the models are quite good, the precision, sensitivity, and F1-score values obtained as a result of the test are lower than the accuracy value. This situation shows that the model is inadequate in correctly predicting the positive class. Data augmentation was performed for the images in the data set for training and validation, thus increasing the diversity and preventing the overfitting problem. However, to see the model's performance on real data and obtain reliable results, it is important not to perform data augmentation on the test data

set. Since the number of images in the created data set is limited, the small amount of data allocated for testing causes the precision, sensitivity, and F1-score values more sensitive.

Additionally, according to the experimental results, although we obtained high accuracy values in certain models, other parameters are relatively low. As we have seen in the literature, the main reason for the low results is a class imbalance in the data set or the nature of the data set. There are no class imbalances in the data sets we used, both for the original and the data augmented data set. However, when the complexity matrices are examined in more detail, it is seen that lichen sclerosus and morphea diseases are predicted more incorrectly than vasculitis instead of each other. While there is a deterioration in the collagen structure in lichen sclerosus and morphea diseases, vasculitis disease does not affect the collagen structure. The number of convolution layers can be increased for a deep learning model that produces better predictions, or a different model can be developed to distinguish lichen sclerosus and morphea diseases better.

4. Conclusion

This study applied classification with machine learning methods for three different skin diseases: Lichen Sclerosus, Morphea, and Vasculitis. Testing operations were performed on eight different models using transfer learning with CNN models and different deep learning models created within the scope of the study. Unlike studies on the classification of skin diseases with machine learning, histopathological images were used in this study for the first time. Since cells and tissues are detailed in histopathological images, detecting related diseases through features extracted from these images provides more reliable and accurate results than diagnostic methods performed with non-invasive images. With the proposed approach, even if the effect of the disease on the skin is not yet seen, the disease status can be revealed with the changes observed at the cellular level.

According to the test results, better results were obtained in both CNN networks and transfer learning methods on data sets with data augmentation. Among all the results, although the accuracy value of the VGG16 model was obtained as the highest with 0.9755, it is seen that the overall performance of the ResNet50 model is good on all data sets. It has been shown with experimental results that the classification performance can be increased by using transfer learning in small-scale data sets.

Our future work will continue to evaluate the reliability of the classification performed with real and artificially augmented images by increasing the number of images in the dataset and obtaining better results by fine-tuning the parameters in transfer learning.

References

[1] R. Pangti *et al.*, "A machine learning-based, decision support, mobile phone application for diagnosis of common dermatological diseases," *Journal of the European Academy of Dermatology and Venereology*, vol. 35, no. 2, pp. 536–545, Feb. 2021, doi: 10.1111/JDV.16967.

[2] O. Jimenez-Del-Toro *et al.*, "Analysis of Histopathology Images: From Traditional Machine Learning to Deep Learning," *Biomedical Texture Analysis: Fundamentals, Tools and Challenges*, pp. 281–314, Jan. 2017, doi: 10.1016/B978-0-12-812133-7.00010-7.

[3] G. Litjens *et al.*, "Deep learning as a tool for increased accuracy and efficiency of histopathological diagnosis," *Sci Rep*, vol. 6, no. 1, p. 26286, May 2016, doi: 10.1038/srep26286.

[4] C. Y. Zhu *et al.*, "A Deep Learning Based Framework for Diagnosing Multiple Skin Diseases in a Clinical Environment," *Front Med (Lausanne)*, vol. 8, p. 626369, Apr. 2021, doi: 10.3389/FMED.2021.626369/BIBTEX.

[5] M. Tanaka *et al.*, "Classification of a large-scale image database of various skin diseases using deep learning," *Int J Comput Assist Radiol Surg*, vol. 16, no. 11, pp. 1875–1887, Nov. 2021, doi: 10.1007/S11548-021-02440-Y/FIGURES/8.

[6] R. Sadik, A. Majumder, A. A. Biswas, B. Ahammad, and M. M. Rahman, "An in-depth analysis of Convolutional Neural Network architectures with transfer learning for skin disease diagnosis," *Healthcare Analytics*, vol. 3, p. 100143, Nov. 2023, doi: 10.1016/J.HEALTH.2023.100143.

[7] E. Goceri, "Deep learning based classification of facial dermatological disorders," *Comput Biol Med*, vol. 128, p. 104118, Jan. 2021, doi: 10.1016/J.COMPBIOMED.2020.104118.

[8] L. Duran-Lopez, J. P. Dominguez-Morales, A. F. Conde-Martin, S. Vicente-Diaz, and A. Linares-Barranco, "PROMETEO: A CNN-Based Computer-Aided Diagnosis System for WSI Prostate Cancer Detection," *IEEE Access*, vol. 8, pp. 128613–128628, 2020, doi: 10.1109/ACCESS.2020.3008868.

[9] Y. Wu *et al.*, "Recent Advances of Deep Learning for Computational Histopathology: Principles and Applications," *Cancers (Basel)*, vol. 14, no. 5, p. 1199, Feb. 2022, doi: 10.3390/cancers14051199.

[10] A. Mumuni and F. Mumuni, "Data augmentation: A comprehensive survey of modern approaches," *Array*, vol. 16, p. 100258, Dec. 2022, doi: 10.1016/J.ARRAY.2022.100258.

[11] F. Perez, C. Vasconcelos, S. Avila, and E. Valle, "Data augmentation for skin lesion analysis," *Lecture Notes in*

Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 11041 LNCS, pp. 303–311, 2018, doi: 10.1007/978-3-030-01201-4_33/FIGURES/3.

[12] A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," *2018 International Interdisciplinary PhD Workshop, IIPhDW 2018*, pp. 117–122, Jun. 2018, doi: 10.1109/IIPHDW.2018.8388338.

[13] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 91–99, Jun. 2022, doi: 10.1016/J.GLTP.2022.04.020.

[14] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019, doi: 10.1186/S40537-019-0197-0/FIGURES/33.

[15] M. S. Ali, M. S. Miah, J. Haque, M. M. Rahman, and M. K. Islam, "An enhanced technique of skin cancer classification using deep convolutional neural network with transfer learning models," *Machine Learning with Applications*, vol. 5, p. 100036, Sep. 2021, doi: 10.1016/J.MLWA.2021.100036.

[16] A. Saber, M. Sakr, O. M. Abo-Seida, A. Keshk, and H. Chen, "A Novel Deep-Learning Model for Automatic Detection and Classification of Breast Cancer Using the Transfer-Learning Technique," *IEEE Access*, vol. 9, pp. 71194–71209, 2021, doi: 10.1109/ACCESS.2021.3079204.

[17] M. Fraiwan and E. Faouri, "On the Automatic Detection and Classification of Skin Cancer Using Deep Transfer Learning," *Sensors 2022, Vol. 22, Page 4963*, vol. 22, no. 13, p. 4963, Jun. 2022, doi: 10.3390/S22134963.

[18] H. Aljuaid, N. Alturki, N. Alsubaie, L. Cavallaro, and A. Liotta, "Computer-aided diagnosis for breast cancer classification using deep neural networks and transfer learning," *Comput Methods Programs Biomed*, vol. 223, p. 106951, Aug. 2022, doi: 10.1016/J.CMPB.2022.106951.

[19] S. Hosseinzadeh Kassani, P. Hosseinzadeh Kassani, M. J. Wesolowski, K. A. Schneider, and R. Deters, "Deep transfer learning based model for colorectal cancer histopathology segmentation: A comparative study of deep pre-trained models," *Int J Med Inform*, vol. 159, p. 104669, Mar. 2022, doi: 10.1016/J.IJMEDINF.2021.104669.

[20] N. Ahmad, S. Asghar, and S. A. Gillani, "Transfer learning-assisted multi-resolution breast cancer histopathological images classification," *Visual Computer*, vol. 38, no. 8, pp. 2751–2770, Aug. 2022, doi: 10.1007/S00371-021-02153-Y/FIGURES/21.

[21] A. Ben Hamida *et al.*, "Deep learning for colon cancer histopathological images analysis," *Comput Biol Med*, vol. 136, p. 104730, Sep. 2021, doi: 10.1016/J.COMPBIOMED.2021.104730.

[22] P. Kora *et al.*, "Transfer learning techniques for medical image analysis: A review," *Biocybern Biomed Eng*, vol. 42, no. 1, pp. 79–107, Jan. 2022, doi: 10.1016/J.BBE.2021.11.004.

[23] M. Iman, H. R. Arabnia, and K. Rasheed, "A Review of Deep Transfer Learning and Recent Advancements," *Technologies 2023, Vol. 11, Page 40*, vol. 11, no. 2, p. 40, Mar. 2023, doi: 10.3390/TECHNOLOGIES11020040.

[24] H. E. Kim, A. Cosa-Linan, N. Santhanam, M. Jannesari, M. E. Maros, and T. Ganslandt, "Transfer learning for medical image classification: a literature review," *BMC Medical Imaging 2022 22:1*, vol. 22, no. 1, pp. 1–13, Apr. 2022, doi: 10.1186/S12880-022-00793-7.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. Burges, L. Bottou, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2012.

[26] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, Sep. 2014, Accessed: Sep. 06, 2024.

[27] C. Szegedy *et al.*, "Going Deeper with Convolutions," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June-2015, pp. 1–9, Sep. 2014, doi: 10.1109/CVPR.2015.7298594.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 770–778, Dec. 2015, doi: 10.1109/CVPR.2016.90.

[29] "ImageNet," https://www.image-net.org/.

Author(s) Contributions

Recep Güler contributed to design, data collection, data analysis, writing, material support and literature review. Zehra Karapınar Şentürk contributed to design, data analysis, writing, material support and literature review. Mehmet Gamsızkan contributed to data collection, data analysis, material support. Yunus Özcan contributed to data collection, data analysis, writing, material support.

Acknowledgments

The authors would like to thank Research Assistant Sümeyye Güneş Takır for creating the dataset used in this study.

Conflict of Interest Notice

The authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical Approval and Informed Consent

It is declared that during the preparation process of this study, scientific and ethical principles were followed, and all the studies benefited from are stated in the bibliography.

Availability of data and material

Not applicable.

Artificial Intelligence Statement

No artificial intelligence tools were used while writing this article.

Plagiarism Statement

This article has been scanned by iThenticate ™.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
http://saucis.sakarya.edu.tr/
e-ISSN: 2636-8129
Publisher: Sakarya University
Vol. 8, No. 2, 322-345, 2025
DOI: 10.35377/saucis..1626057
SAUCIS
SAKARYA UNIVERSITY
RESEARCH ARTICLE

Flood Area Prediction using a Stacked Ensemble of Tree-Based Algorithms

Olusogo Julius Adetunji

Department of Computer Engineering, Olabisi Onabanjo University, Ago Iwoye, Ogun State, Nigeria, ror.org/05jt4c572

Olusogo Julius Adetunji,
Department of Computer Engineering, Olabisi Onabanjo University, Ago Iwoye , Ogun State, Nigeria.
adetunji.olusogo@oouagoiwoye.edu.ng

Article History:
Received: 25.01.2025
Revised: 13.06.2025
Accepted: 13.06.2025
Published Online: 30.06.2025

ABSTRACT
Floods cause significant loss of life, property damage, and long-term socioeconomic disruptions, with over 100 annual deaths globally. This research addresses the drawbacks of the existing models, such as overfitting effects, inadequate dataset and limited study areas through the adoption of a stacked ensemble-based model. The model contained five different tree - based models namely hoeffding tree, decision tree, functional tree, reduced error pruning (REP) tree and decision stump algorithms. The model was implemented as a system using MATLAB Simulink, version 2020a on laptop with 4GB Memory. Experimental results indicate that REP Tree performed better than other four individual tree algorithms with accuracy of 98.74%, 97.81% and 97.43% for Dataset A, Dataset B and Dataset C respectively. For Dataset A, stacked ensemble model performed better than single algorithms with accuracy, precision, specificity, f1score and recall of 99.62%, 99.51%, 99.51%, 99.63% and 99.73% respectively. For Dataset B, stacked ensemble model also performed better than single algorithms with accuracy, precision, specificity, f1score and recall of 98.45%, 99.11%, 98.12%, 97.37% and 99.06% respectively. For Dataset C, stacked ensemble model performed better than single algorithms with accuracy, precision, specificity, f1score and recall of 98.75%, 99.25%, 99.64%, 99.90% and 99.24% respectively. Our model's 99.62% accuracy on Dataset A demonstrates potential for integration with real-time sensor networks, enabling scalable flood early-warning systems in vulnerable regions like Lagos and Kuala Lumpur.
Keywords: Ensemble, Flood, Stacked, Tree Based

1. Introduction

Flooding is the leading contributor of natural disasters that occurs in a situation, in which the soil are supplied with water, more than its retentive capacity [1–4]. Different factors contribute to the occurrence of flood and these factors are categorized into natural and human factors [3]. Floods are caused naturally as a result of houses built near river, insufficient or no vegetation and nature of soil [5–8] while human causes include dam overflow or dam break down and poor town planning [1], [9]. Considering and examining different types of flood, flash flood is prevalent and most fatal form due to its sudden mode of actions [10, 11]. Other types include coastal, riverine and urban flooding [12]; coastal flood is caused by storm sudden rise [13]; urban flooding occurs in cities where there is no or less proper drainage to accommodate the passage of water; riverine flooding occurs when water fills the river or stream to the extent of spreading on its bank [1]. Globally, flood occurrence results to greater than 100 deaths on average of 10 times a year [14]. Several measures can be taken in controlling and preventing different categories of floods, such measures include effective town planning system, construction of proper drainage structures, effective flood waste management system; conservation of soil along drainage areas which helps in reducing soil erosion caused by flood and also, improved ecosystem protection and planting of trees by relevant authorities [22,23]. Machine learning algorithms, artificial intelligence and decision support systems have been widely applied for the prediction of different categories of diseases and natural disasters [4, 17, 24, 25, 46, 47]. Examples of such machine learning models for the prediction of floods are Artificial Neural Networks [11, 17, 26, 27]; Support Vector Machine [11, 28, 29, 37]; decision tree [2, 30–32]. Artificial Neural Network was developed for the prediction of floods[17, 26, 27] and compared with the performance of logistic regression [26]. Different conditioning factors (CgFs) were considered in the data employed for flood prediction using Artificial Neural Network (ANN) and those conditioning factors were rainfall, aspect ratio, curvature, distance to rail, distance to water, nature of soil, roughness, slope, stream power index, topographic wetness index, temperature, elevation, land use, curve number and road [17, 26]. Authors in [26] compared the performance of artificial neural network (ANN) model with logistic regression. Artificial Neural Network performed better than logistic regression (LR) with accuracy and performance success of 76.4% and 96.4% respectively. The model developed by [17] employed parameter tuning approach to enhance its predictive performance. The novel model gives more enhanced predictive performance with testing accuracy of 96.54% and training accuracy of 98.91%. Decision tree model was developed with synthetic minority oversampling technique (SMOTE) and compared with or without dataset imbalance by [33 - 38]. The dataset has eight variables with distinct variables from the dataset employed by [17]. Those variables include wind, temperature, humidity, water level, date daily rainfall, monthly rainfall and class for the prediction of floods.

Cite as: O. J. Adetunji. (2025). Flood area prediction using a stacked ensemble of tree-based algorithms. *Sakarya University Journal of Computer and Information Sciences*. vol. 8, no. 2, pp. 322-345, 2025 doi: 10.35377/saucis...1626057

Ensemble methods are known for improving prediction accuracy and robustness by combining prediction of multiple single models [21]. Some of the works where ensemble techniques have been applied are summarized in the Table 1. Authors in [15] Applied ensemble algorithms for the prediction of flood areas. Those ensemble algorithms are extreme gradient boosting ensemble model, adaptive boosting, boosted generalized linear model and deep boosting model and carried out on Talar Watershed, Mazandaran Province, Iran study area.. The experimental results showed that all the applied models are efficient for the flood hazards prediction with area under curve of 0.91, 0.88, 0.89 and 0.87 for deep boosting model, boosted generated linear model, adaboost and extreme gradient boosting ensemble model respectively. Considering other evaluation metrics, deep boosting model outperformed the performances of other ensemble models with sensitivity, specificity, positive predictive value (PPV) and NPV of 0.88, 0.86, 0.88, 0.86 and 0.86 respectively. Stacked ensemble of decision tree classifier, K-nearest neighbor, binary logistic regression and support vector classifier were applied by [18] flood areas prediction. Ensemble model outperformed other individual classifiers with accuracy and standard deviation of 93.3% and 0.098 respectively [18]. Random Forest with Bagged CART, XG Boost, Stochastic Gradient Boosting were applied for the prediction of floods by [19] and with AdaBoost, Gradient Boosting, Random Forest and Random Forest – Gradient Boosting by [20]. Random Forest performed better than other ensemble models in both works [19, 20] with accuracy of 91% for the work of [19] and 83% for the work of [20]. Authors in [4] compared and applied four different ensemble models. The experimental results showed that the performance of Adaptive Neuro-fuzzy inference system (ANFIS) ensemble with genetic algorithm exceeded the performances of other three models with highest success rate area under curve of 0.922, prediction rate AUC of 0.924 and the accuracy of training and validation with 0.886 and 0.883 respectively. Authors in [11] developed a novel model that combined Bayesian belief network model with extreme learning machine and back propagation (BP) structure optimized by a genetic algorithm (GA) named GA-BN-NN model. The experimental results indicated that the novel model (GA-BN-NN) model has better goodness-of-fit with prediction accuracy of 0.966. Some of the drawbacks of these existing single and ensemble models are overfitting effects, inadequate datasets and limited study areas, hence this work addresses these drawbacks by developing stacked ensemble models for improving predictive capacity, presentation of three different Datasets in three different study areas. Other sections are section 2, section 3 and section 4 which depict materials and method, results and discussion and conclusion respectively.

Table 1. Related Ensemble Models and Current Works

Author	Focus	ML algorithm used	Source of dataset	Result
[15]	Flood hazard areas prediction using different boosting ensemble models	Adaptive Boosting, Boosted Generalized Linear Model, Extreme Gradient Boosting and Deep Boost (DB)	Talar Watershed, Mazandaran Province, Iran	DB has most efficient Area Under Curve (AUC) with 91%, compared with other boosting ensemble models.
[16]	Adoption of ensemble machine learning model for flood prediction	Bagging, Random Subspace, Random Forest, Support Vector Machine and Artificial Neural Network (ANN)	Teesta sub catchment, Northern region of Bangladesh	Bagging Model has the maximum performance with Area Under Curve (AUC) of 87.3%
[18]	Compare performance of single classifiers and stacked ensemble model for flood prediction	Stacked ensemble of K-Nearest Neighbors (KNN), Support Vector Classifier (SVC), Decision Tree (DT), Binary Logistic Regression,	Kerala dataset of Southern region of India.	Stacked ensemble model has accuracy of 93.3%
[19]	Adoption of ensemble model for the prediction of floods	Bagged CART, Random Forest, XG Boost, Stochastic Gradient Boosting	36 States of Nigeria and Federal Capital Territory	Random Forest and XG Boost performed better than other models with the same accuracy of 91%.
[20]	Flood prediction using different ensemble models.	Random Forest, AdaBoost, Random Forest – Gradient Boosting and Gradient Boosting	Oum Er Rbia watershed, located in the Khenifra Province	Random Forest performed better than other ensemble models with accuracy of 83%
[48]	Ensemble of ANN for Urban flood prediction	Ensemble ANN model	Chinese City of Macao	The model has Root Mean Square (RMS) and coefficient of determination of 0.20 and 0.96 respectively
[49]	Flood prediction using google earth engine and remote sensing	Gradient boosting ensemble, adaboost and gradient boostung	Oum Er Rbia watershed, Morocco	Gradient boosting ensemble has accuracy of 0.96
Current work	Stacked ensemble of tree-based algorithms for the prediction of flood areas.	Stacked Ensemble of Decision Stump, Hoeffding Tree, Functional Tree, Decision Tree and Reduced Error Pruning Tree	Dataset A: NiMeT/NIHSA: Nigeria, Dataset B: USGS/DEM/NASA and Dataset C: Jabatan Meteorologi Malaysia Dataset, Kuala Lumpur Area, Malaysia	Stacked ensemble of trees performed better than single classifiers with accuracy of 99.62%, 98.45% and 98.75% for Datasets A, B and C respectively.

2. Materials and Methods

A stacked ensemble model was developed with MATLAB (R2020a version) platform. MATLAB R2020a version was installed on laptop computer hardware with two Intel Celeron (N3060) processors each having 1.60 GHz speed and 4GB Memory. Wrapper feature selection techniques integrated with particle swarm optimization algorithm (PSO) were employed for the selection of features on the Datasets. Tree-based classifiers namely Functional Tree (FT), Hoeffding Tree (HT), Decision tree (DT), Decision Stump (DS) and REP tree as depicted in the Figure 1 were selected as base classifiers. Tree based algorithms were selected based on their performances in the preliminary evaluation and have been known for better accuracy, stability and ease of interpretation. Also, fine-tuned particle swarm optimization algorithm was employed as a meta learner for stacked ensemble model.

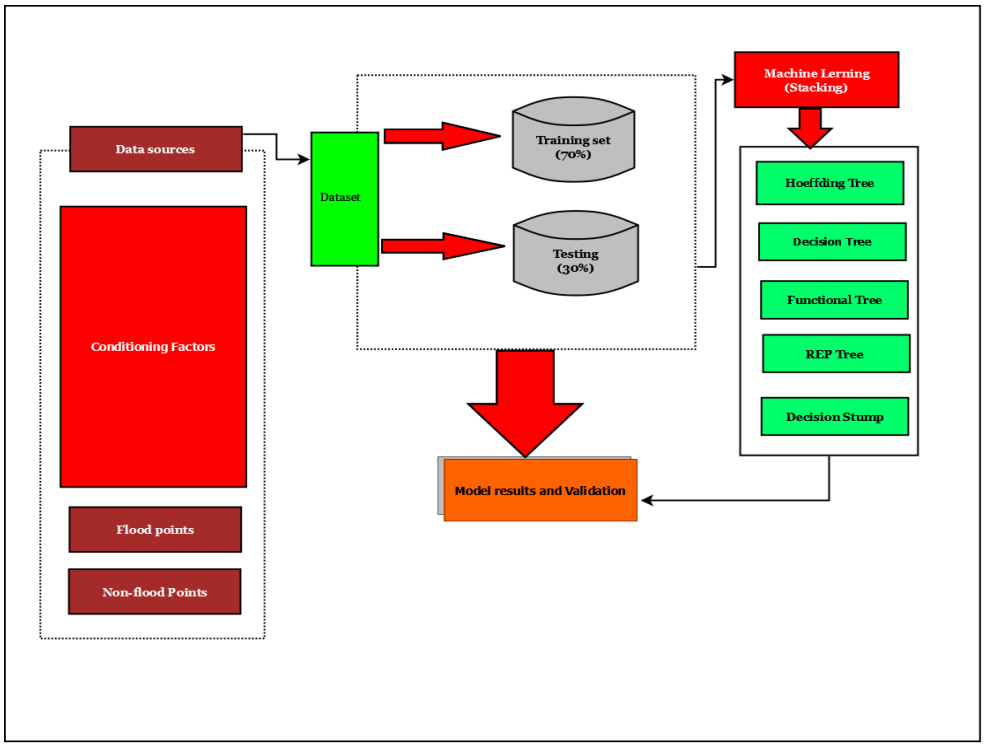


Figure 1. Flowchart of model developed for flood prediction

2.1 Data Acquisition and Data Sources

Three distinct sets of dataset were employed for this research; the first dataset (NiMeT/ NIHSA Dataset) was obtained from two different organizations in Nigeria, namely Nigerian Meteorological Agency (NiMeT) and Nigerian Hydrological Service Agency (NIHSA); the second dataset (USGS/DEM/NASA dataset) is also Nigeria dataset and was collected from different sources such as United State Geological Survey Earth Explorer, digital elevation model (DEM), Nigerian Aeronautics and Space Administration (NASA) and lastly, the third dataset (JMM dataset) was obtained from Jabatan Meteorologi Malaysia. The first, second and third Datasets are labelled Dataset A, Dataset B and Dataset C respectively.

2.1.1 NiMeT/NIHSA Flood Dataset

NiMeT/NiHSA Flood dataset was sourced from Eti - Osa area of Lagos State, Nigeria for five (5) years from January 2017 to December 2021 consisting of 1826 instances with 6 independent variables or input and 1 dependent variable or output. The attributes of Dataset A were examined with different characteristics as depicted in Table 2. The input features are date, water level, daily rainfall, temperature, wind and humidity while flood class which can be flood (1) or No flood (0). This Dataset A is related to Dataset employed for the work of [34]. The Dataset generally follows the principle depicted in the Equation 1.

$$\text{Dataset (A)} = (p_1, k_1), (p_2, k_2), (p_3, k_3) \ldots\ldots\ldots\ldots (p_n, k_n) \quad (1)$$

Where $p_1 \in P$, is the i^{th} independent variable or input and $q_1 \in K$, the corresponding dependent variable or output. $P = \mathbb{R}^d$, where $p_1 = (p_{i1}, p_{i2}, p_{i3} \ldots\ldots p_{id})$ is a d-dimensional vector or instance. Figure 2 depicts Pearson's correlation plot of the input and target variables for Dataset A. There is no strong correlation between the attributes. The major strong correlation exists between the daily rainfall (RF Daily) attribute and the flood (class) with 0.82. Therefore, there is high degree of association between the attribute, daily rainfall and the output, flood (class). Considering the attributes of the dataset in the Etiosa on the map, Figure 3 (A), 3(B), 3 (C), 3(D), 3(E) illustrate contexture view of humidity, daily rainfall, temperature, water level and wind speed respectively for the study area (Etiosa).

Table 2: Statistics of attributes of the Dataset A

Attribute	Min	Max.	Mean	Standard Deviation	25th Percentile	50th Percentile	75th Percentile	90th Percentile
Water Level (cm)	40	360	222	59.22	190	230	270	290
Rainfall Daily (mm)	0	136	4.07	12.20	0	0	0.28	11.90
Temperature (^{0}C)	11.85	31.35	26.95	1.87	25.75	26.75	28.35	29.40
Humidity (%)	19	99	80.22	7.96	77	81	85	88.5
Wind (m/s)	0.5	11.0	3.90	1.40	3	3.72	4.63	5.75

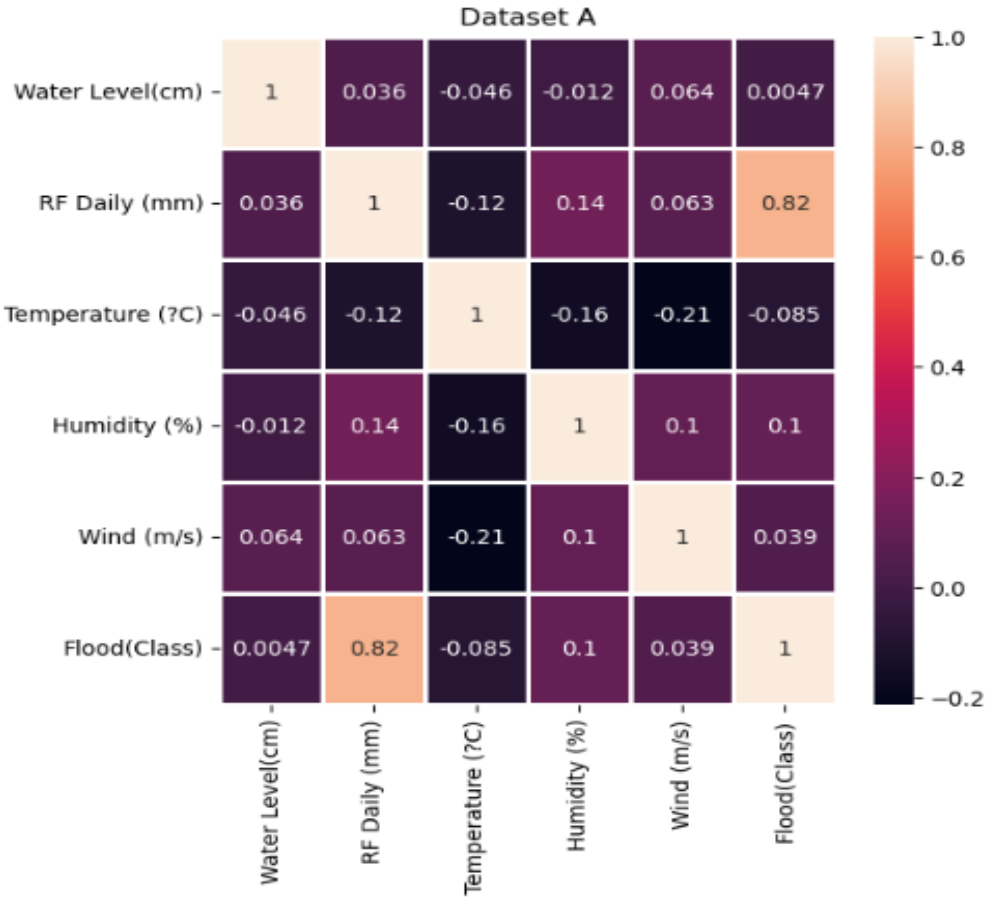


Figure 2. Pearson correlation's plot of the input and target variables for Dataset A

2.1.2 USGS/DEM/NASA Dataset

USGS/DEM/NASA Dataset was sourced majorly from United State Geological Survey Earth Explorer, digital elevation model (DEM), Nigerian Aeronautics and Space Administration (NASA). The dataset has 1530 samples with sixteen conditioning factors (CgFs). Those conditioning factors are rainfall, temperature, land cover, soil type, slope, aspect, elevation, road distance, river distance, roughness, curve number, stream power index, curvature, Topographic Wetness Index (TWI) and distance to trail. This Dataset B generally follows the form as illustrated in the Equation 1. The summary and some of characteristics of this Dataset B are elucidated in the Table 3.

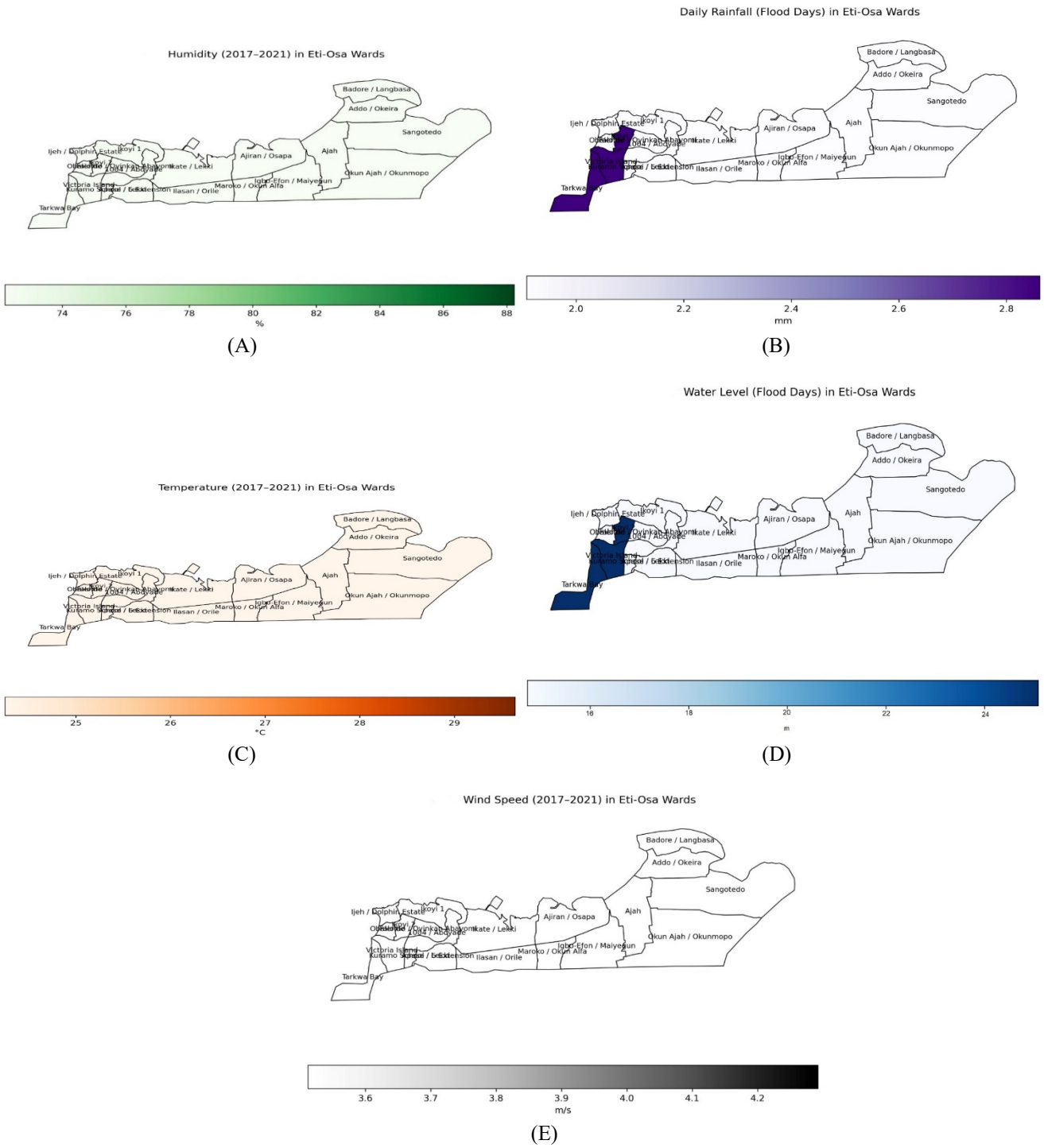


Figure 3. Contexture view of the attributes of Dataset A in the specified study area (Eti Osa): (A) Humidity; (B) Daily Rainfall; (C) Temperature; (D) Water Level; (E) Wind Speed

Figure 4 shows the Pearson's correlation plot between the input and output variables. There is no strong correlation between the input and output variables. The highest correlation is between the elevation and rainfall variables with a Pearson's correlation value of 0.6. Considering the attributes of the dataset in the Niger-Benue axis on the Nigeria map, Figure 5 (F), 5(G), 5 (H), 5(I), 5(J), 5(K), 5(L), 5(M) and 5(N) illustrate contexture view of Niger-Benue axis, states within the Niger-Benue axis, elevation, water level, soil type, slope, roughness, rainfall and land cover respectively.

Table 3: Statistics of attributes of the Dataset B

Attribute	Min.	Max.	Mean	Standard Deviation	25th Percentile	50th Percentile	75th Percentile	90th Percentile
Slope	0	89.99	89.44	5.62	89.84	89.93	89.96	89.98
Soil Type	1	117	27.20	33.66	1.00	1.00	54.00	73.00
Elevation	-3	1595	309.60	218.39	154	297	433	564.30
Land Cover	10	90	25.42	19.76	10	20	30	50
Roughness	0	553	34.38	51.17	11	22	36	68
Rainfall	587	2647	1278	440	901	1162	1544	1891
Water	0	1.89	0.38	0.29	0.15	0.31	0.56	0.80
Road	0	0.64	0.04	0.05	0	0.02	0.05	0.09
Rail	0	3.07	0.63	0.59	0.18	0.48	0.90	1.49
Curvature	-4080	3360	-6.15	285.22	-82.83	0	85.87	219.19
Aspect	1.39	360	181.50	104.34	90	180	270	330.37
Temperature	0	33.41	29.05	4.48	28.00	29.71	30.92	32
TWI	-22.65	-1.41	-15.93	2.09	-16.06	-16.06	-16.06	-14.54
SPI	-47.33	5.09	-0.24	1.44	-0.12	-0.05	-0.03	0
Curve Number	0	100	78.73	10.87	71	81	83	83

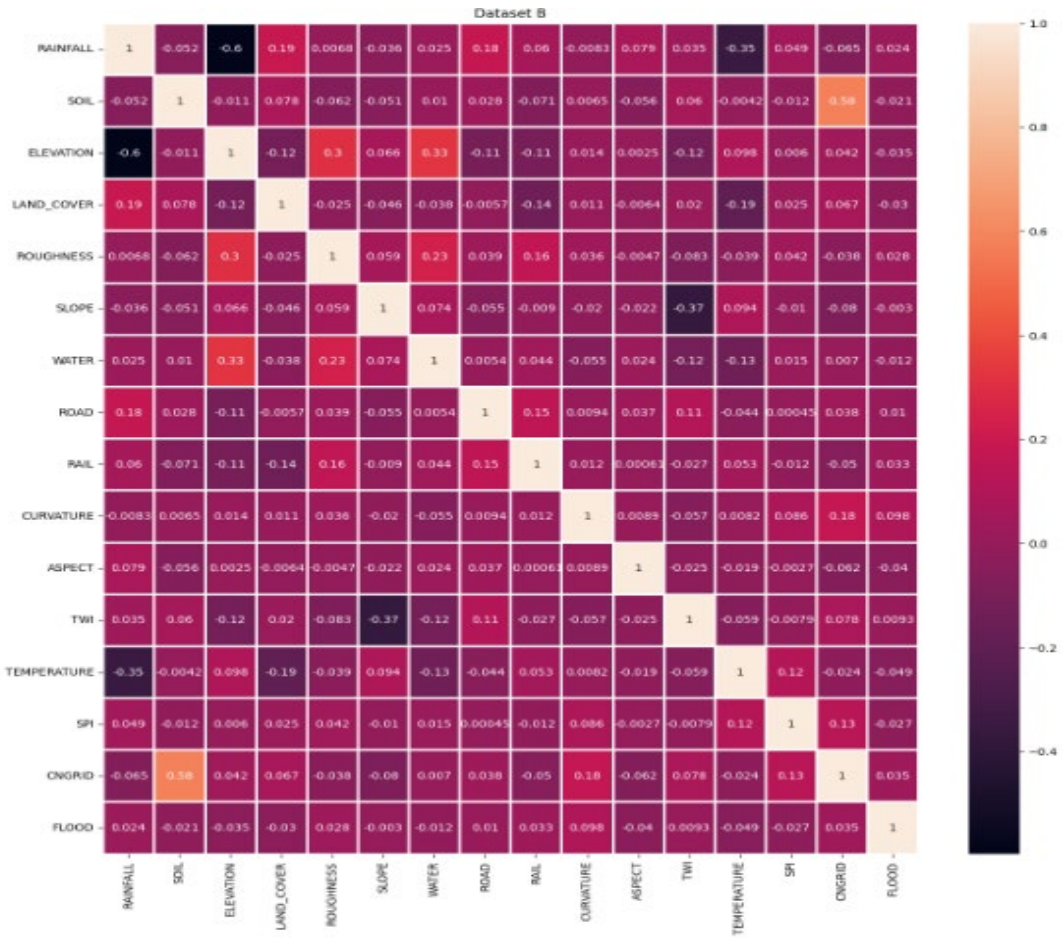
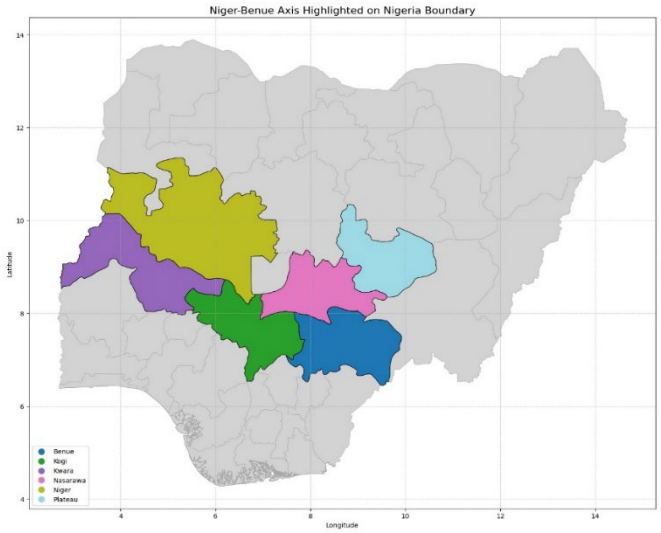
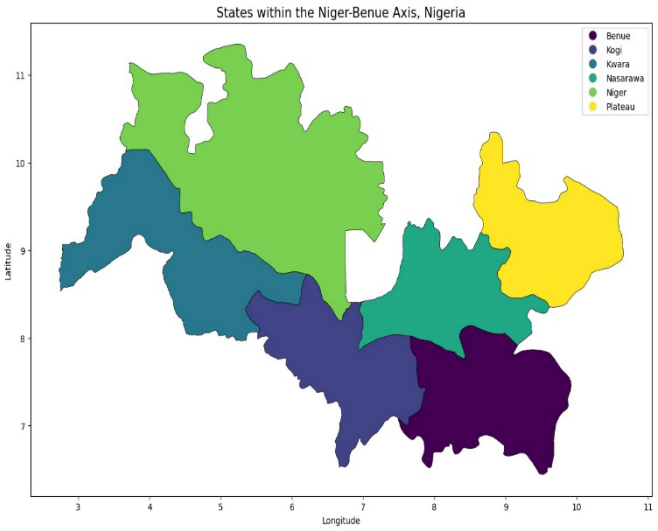


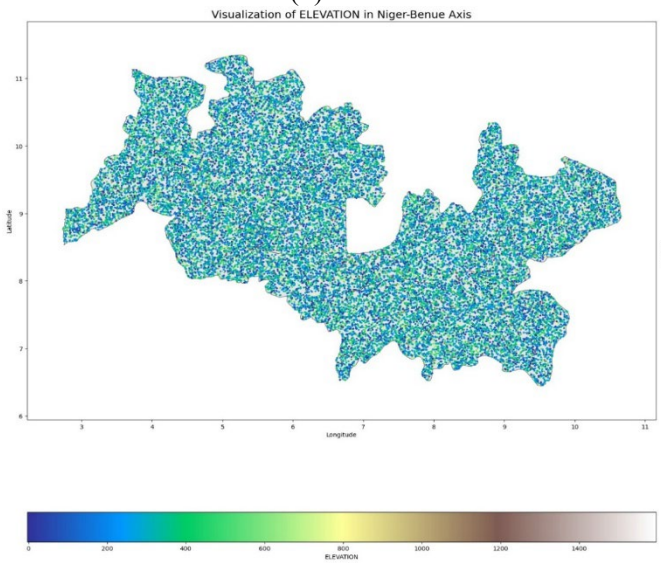
Figure 4: Pearson correlation's plot of the input and target variables for Dataset B


Niger-Benue Axis Highlighted on Nigeria Boundary
Benue
Kogi
Kwara
Nasarawa
Niger
Plateau
Latitude
Longitude

(F)

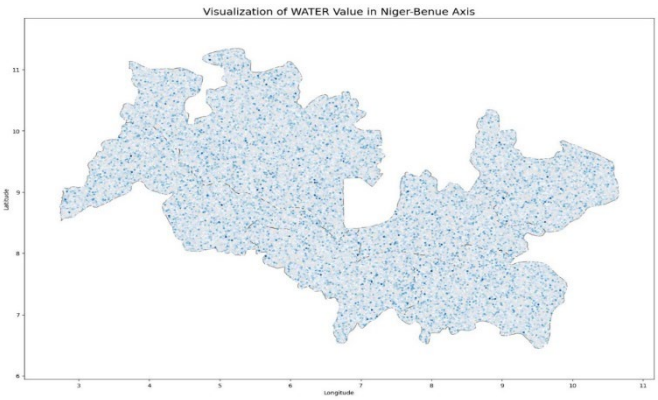

States within the Niger-Benue Axis, Nigeria
Benue
Kogi
Kwara
Nasarawa
Niger
Plateau
Latitude
Longitude

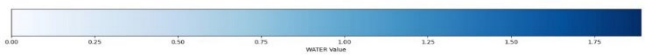
(G)


Visualization of ELEVATION in Niger-Benue Axis
Latitude
Longitude

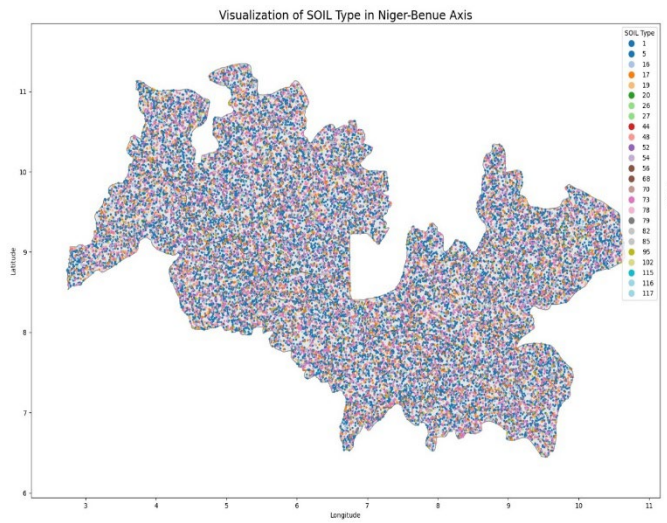
0
200
400
600
800
1000
1200
1400
ELEVATION

(H)

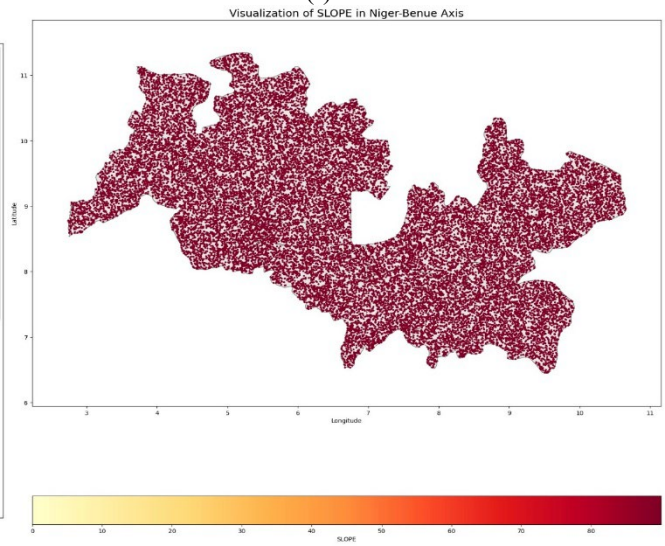

Visualization of WATER Value in Niger-Benue Axis
Latitude
Longitude


0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
WATER Value

(I)


Visualization of SOIL Type in Niger-Benue Axis
SOIL Type
Latitude
Longitude

(J)


Visualization of SLOPE in Niger-Benue Axis
Latitude
Longitude
SLOPE

(K)

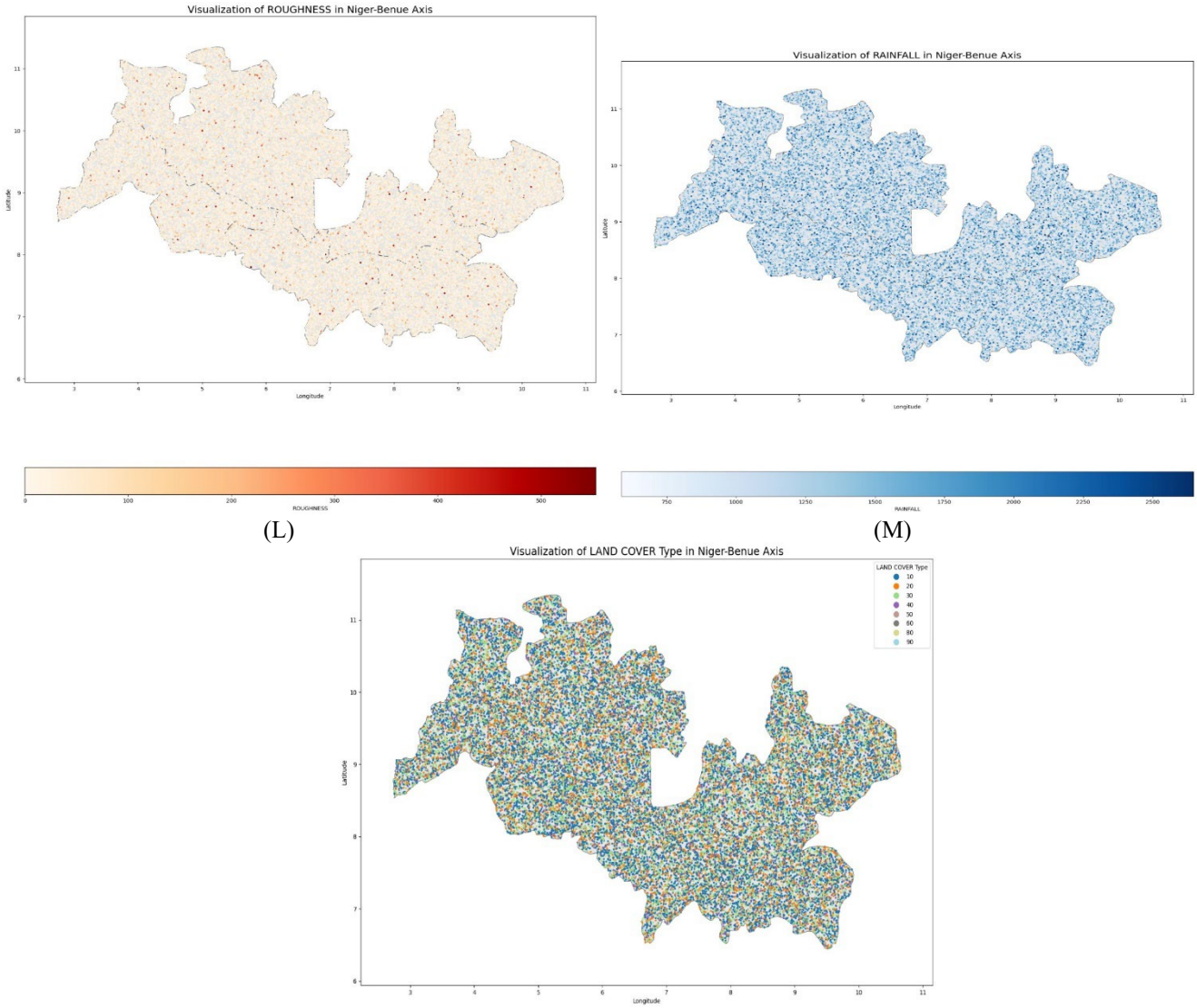


Figure 5. Contexture view of the attributes of Dataset B in the specified study area (Niger – Benue Axis, Nigeria): (E) Niger-Benue Axis on Nigeria boundary; (G) States within the Niger -Benue Axis, Nigeria; (H) Elevation; (I) Water Level; (J) Soil Type; (K) Slope; (L) Roughness; (M) Rainfall; (N) Landcover.

2.1.3 Jabatan Meteorologi Malaysia Dataset

Jabatan Meteorologi Malaysia Dataset is the dataset sourced for five years between 2014 – 2019 in Kuala Lumpur area of Malaysia and contained 1823 instances with date, temperature, rainfall (daily), humidity, temperature, rainfall (monthly), wind and class (either flood or no flood) attributes. This Dataset C generally follows the form as illustrated in the Equation 1. The summary and some characteristics of this Dataset C are elucidated in the Table 3. Figure 6 illustrates the Pearson's correlation plot of the input and output variables. There is no strong correlation between the input and output variables. The highest correlation is between the daily rainfall (RF Daily) attribute and class variable with a Pearson's correlation value of 0.47. Considering the attributes of the dataset in the Kuala Lumpur, Malaysia, Figure 7 (O), 7(P), 7 (Q), 7(R), 7(S) and 7(T) illustrate contexture view of wind, temperature, monthly rainfall, daily rainfall, water level and humidity of the specified area respectively.

Table 3: Statistics of attributes of the Dataset C

Attribute	Min.	Max.	Mean	Standard Deviation	25th Percentile	50th Percentile	75th Percentile	90th Percentile
Water Level (cm)	1703	2543	1898	194	1796	1863	1930	2084
Rainfall Monthly (mm)	1053	1275	1134	69	1071	1121	1173	1275
Rainfall Daily(mm)	0	102	13.12	22.03	0	6	12	45
Temperature (^{0}C)	23.2	26.7	25.04	0.85	24.3	24.9	26	26.2
Humidity (%)	80.5	95.9	87.71	4.08	84.1	84.1	91.2	92.8
Wind (m/s)	0.4	1.4	0.85	0.19	0.7	0.87	1	1

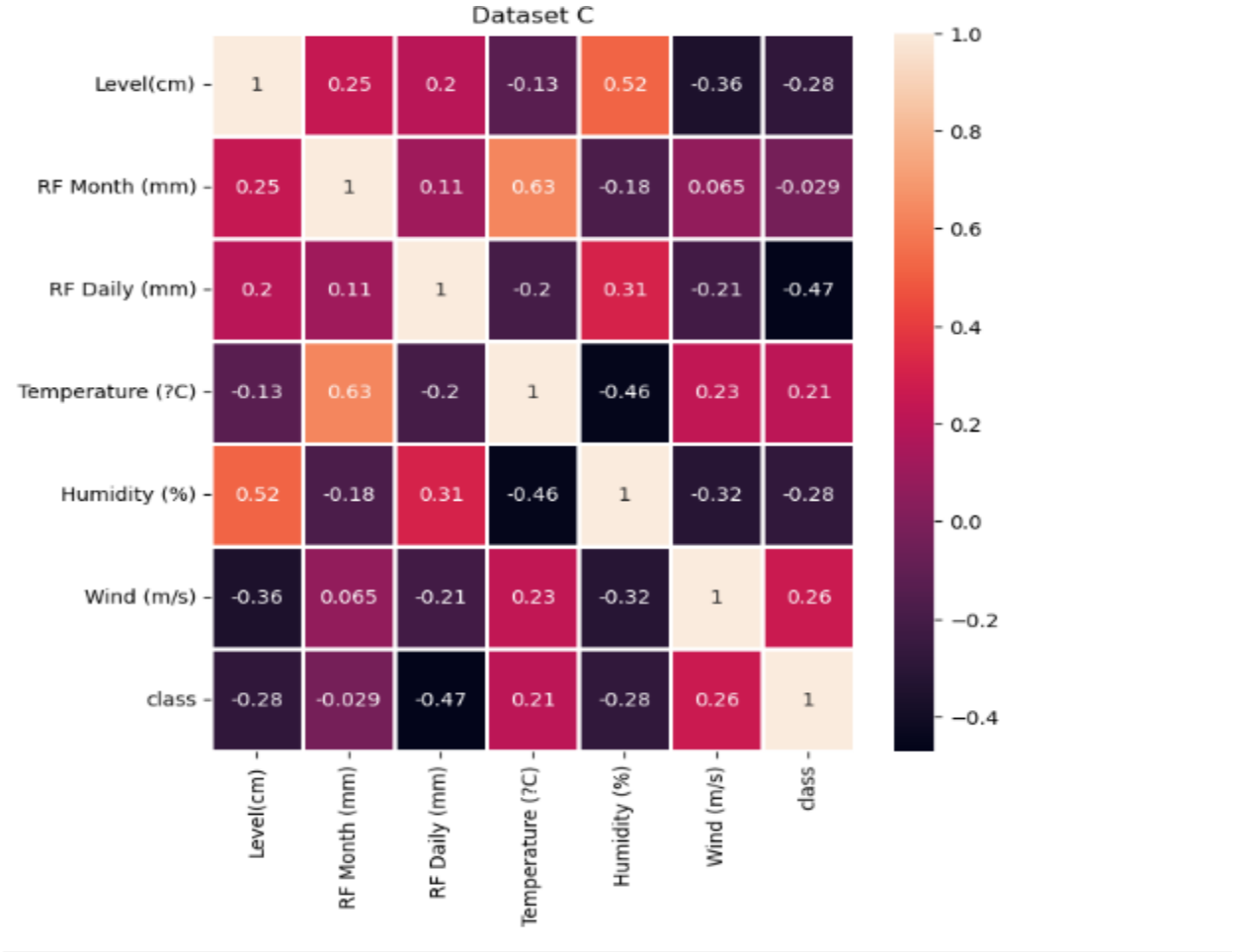
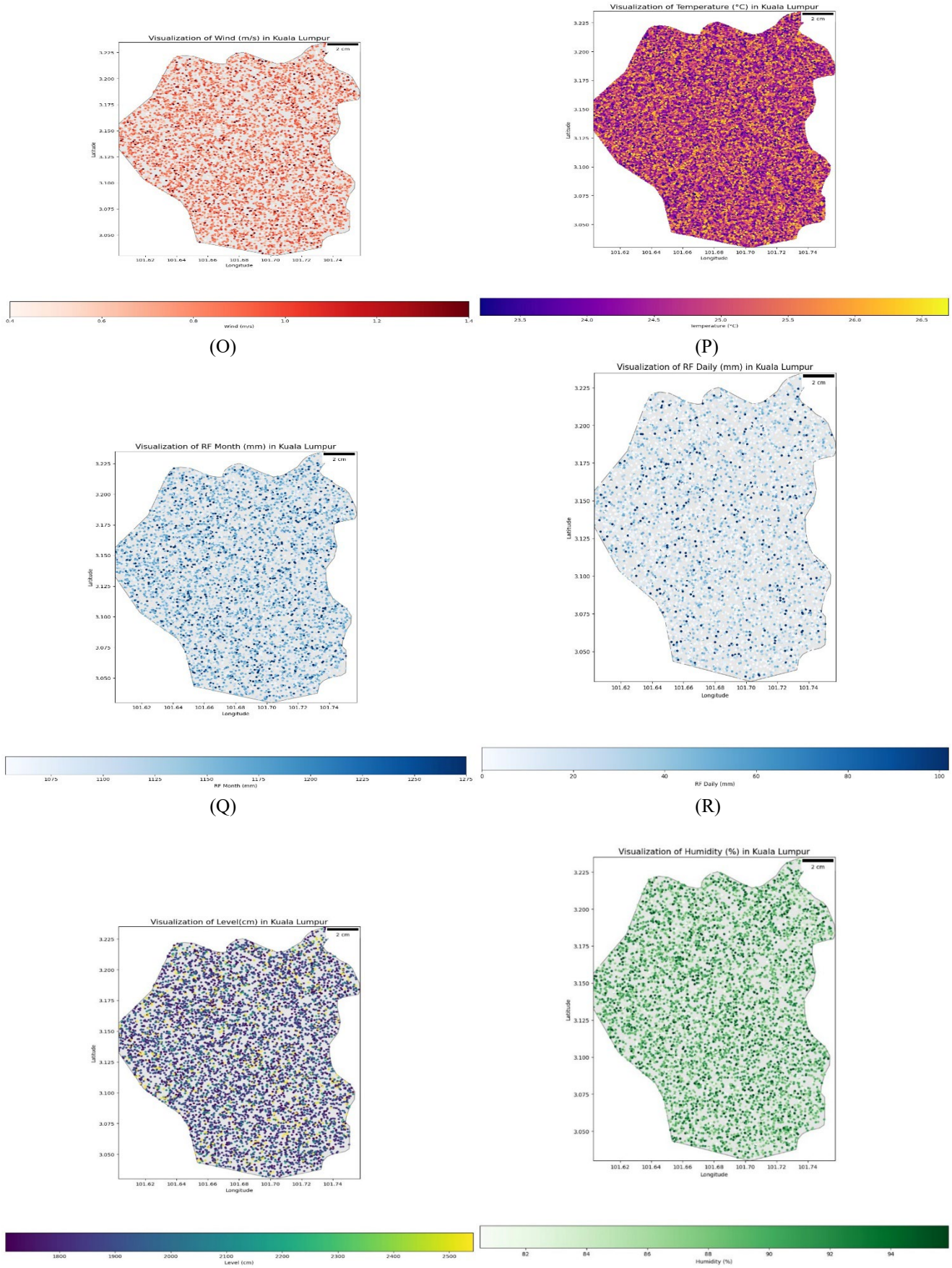


Figure 6: Pearson correlation's plot of the input and target variables for Dataset C



(O) (P)

(Q) (R)

(S) (T)

Figure 7. Contexture view of the attributes of Dataset C in the specified study area (Kuala Lumpur): (O) Wind; (P) Temperature; (Q) Monthly Rainfall; (R) Daily Rainfall; (S) Water Level; (T) Humidity

2.2 Wrapper Feature Selection

Wrapper method of feature selection was employed for the selection of the features in three datasets using particle swarm optimization algorithm to improve the predictive performance of the model. In this case, features are selected iteratively based on particle swarm optimization machine learning algorithm. Particle swarm optimization (PSO) selects features based on this solution vector which represents feature subset of as depicted in the Equations 2, 3 and 4.

$$x = [x_1, x_2, x_{3,\ldots\ldots\ldots\ldots} x_d]; \; x_i \in [0,1] \quad (2)$$

Where d represents the number of features in a given dataset. Considering using the threshold of 0.5 to ascertain tendency of selecting feature (s) and this is given as:

$$x_i = \begin{cases} 1, \\ 0, \end{cases} \text{ if } x_i > 0.5, Otherwise \quad (3)$$

Optimization of the following function occurs,

$$f(x) = \propto \times (1 - P) \times (1 - \propto) \times \frac{N_{selected}}{N_{features}} \quad (4)$$

Where $\propto$ determines trade-off between classifier performance and selected features with respect to the total of all features.

2.3 Tree Based Machine Learning Algorithms

The developed model (stacked ensemble model) consists of five different tree-based algorithms (base classifiers). The algorithms were Hoeffding Tree, Decision Tree, Functional Tree, REP Tree and Decision Stump. Generally, Tree based algorithms are predictive algorithms with high accuracy, stability, ease of interpretation and also map non-linear relationship well. Tree based model trees also require less effort for data preparation during pre – processing, normalization and does not require scaling of data as well.

2.3.1 Hoeffding Tree Algorithm

A very fast decision tree algorithm for streaming data instead of the reuse of instances was proposed by [39]. The main problem of decision tree is the need to reuse instances to compute the best splitting features. The estimation confidence interval of the entropy at a node on a basis of bond is

$$\epsilon = \sqrt{\frac{R^2 \, In 1/\delta}{2n}} \quad (5)$$

Where $R = \text{range of randomvariable}$

$\delta =$ is the probability of estimate not being within ϵ of its expected value, δ is the desired probability of the estimate *not* being within ϵ of its expected value, and n is the number of examples collected at the node.

Algorithm 1: Hoeffding Tree Listing Algorithm [39]

1	Hoeffding Tree Algorithm (Stream,δ)
2	Input: a stream of labelled examples, confidence parameter δ
3	Let HT be a leaf with a single leaf (root)
4	init counts n_{ijk} at root
5	**for** each example (x, y) in Stream
6	**do** HTGROW $(x, y), HT, \delta$
7	HTGROW $((x, y), HT, \delta)$
8	sort (x, y) to $leaf\ l\ using\ HT$
9	update counts $n_{ijk}\ at\ leaf\ l$
10	**if** examples seen so far at l are not all of the same class
11	**then**
12	Compute G for each attribute
13	**if** G (attribute) - G (Second best)> $\sqrt{\frac{R^2 in\ 1/\delta}{2n}}$
14	**then**
15	split leaf on best attribute
16	**for** each branch
17	**do** start new leaf and initialize counts

2.3.2 Functional Tree Algorithm

Functional trees can be categorized as the generalization of multivariate trees. Multivariate decision nodes are built when growing the tree, while functional trees are developed when pruning the trees [40]. Functional tree has the merit of using logistic regression function to isolate the internal nodes and prediction at the leaves [41]. Functional tree has regression model (RM) which is used in internal nodes and leaves [42].

Algorithm 2: Functional Tree Listing Algorithm [41]

1	Functional Tree Algorithm (Dataset, Constructor)
2	If Stop Criterion (Dataset)
3	Return a Leaf Node with a constant value
4	Construct a model $\emptyset$ using constructor
5	For each example $\overrightarrow{x} \in$ DataSet
6	Compute $\hat{y} = \emptyset(\vec{x})$
7	Extend $\vec{x}$ with new attributes $\hat{y}$
8	Select the attributes of original as well as of newly constructed
9	Attributes that maximize some merit-function
10	For each partition i of the Dataset using the selected attribute
11	$\text{Tree}_\text{i} = \text{GrowTree}\ (\text{Dataset}_\text{i}, \text{Constructor})$
12	Return a Tree, as functional node based on selected attribute
13	Containing the $\emptyset\ model$, and descendant $Tree_i$
14	End Function
	.

2.3.3 Decision Stump Algorithm

Decision Stump Algorithm makes use of only one attribute for splitting and discrete attributes, simply consist of single interior node (root has only leaves as successor nodes). Tree becomes more complex for numeric attributes [43]

Algorithm 3: Decision Stump Listing Algorithm [43]

1	A decision stump is defined by
2	$f(X\|j,t) \coloneqq \begin{cases} +1 \\ -1 \end{cases} x^{(\text{j})>t}_{\text{otherwise}}$
3	
4	$where\ j\ \in \{1, \ldots\ldots, \ldots d\}$ indexes an axis in $\mathscr{R}^{\text{d}}$.
5	Weighted data
6	Training data $(\widetilde{X_1}, \widetilde{Y_1})$……., $(\widetilde{X_n}, \widetilde{Y_n})$.
7	With each data point $\widetilde{X_\iota}$, we associate a weight $w_i \geq 0$
8	Training on weighted data
9	Minimize the weighted misclassification error:
10	$(j^*,t^*) \coloneqq arg^{min}_{j,t} \frac{\sum_{i=1}^{n} w_i \text{I}[\{\widetilde{y_1} \neq f(\widetilde{X_\iota}\|j,\ t)\}}{\sum_{i=1}^{n} w_i}$

2.3.4 Reduced Error Pruning Tree Algorithm (REP Tree Algorithm)

Reduced Error Pruning (REP) tree adopts regression tree logic and creates diverse trees in different iterations. The end point of a regression tree is predicted function value rather than predicted classification [44]. In pruning tree, mean square error is measured on the predictions made by the trees. The sum of mean square errors is given and shown in the Equations 6, 7 and 8.

$$S = \sum_{E \in leaves(RT)} \sum_{i \in E} (Y_i - N_T)^2 \quad (6)$$

Where N_T is expressed as,

$$N_T = \frac{1}{P_C} \sum_{i \in T} Y_i; \quad (7)$$

Hence,

$$S = \sum_{E \in leaves(RT)} P_c V_c; \quad (8)$$

Where, $N_T = \text{Predictions for leaf N};\ \text{V}_\text{c} = \text{leaf within variance}$ and P_c is the class prediction.

2.3.5 Decision Tree Algorithm

Decision tree is one of the classification techniques in data mining method that is employed for decision support systems and machine learning processes [41, 43]. The basic structure of decision tree is shown in the Figure 8

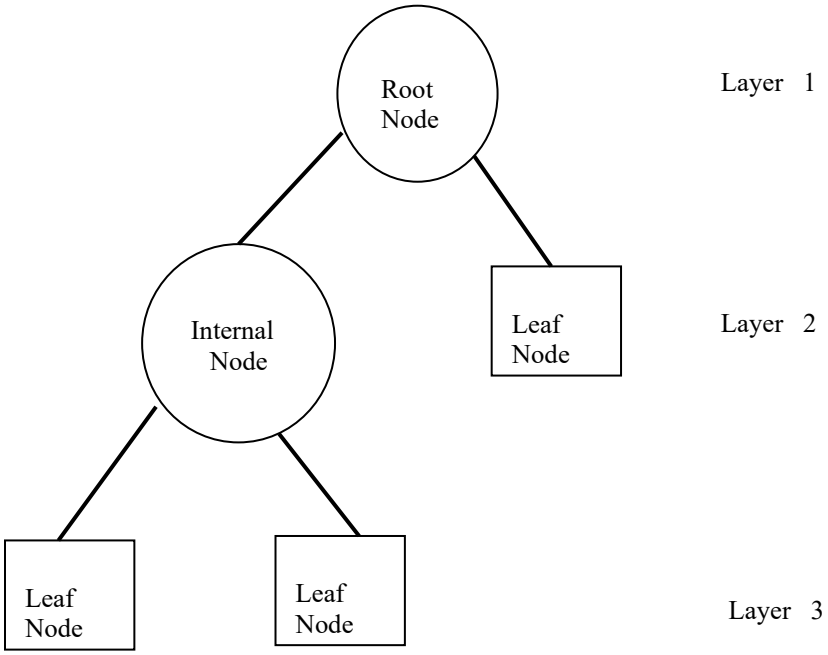


Figure 8. Basic Structure of Decision Tree [45]

Algorithm 3: Decision Tree Listing Algorithm [43]

```
1   Decision Tree Learner (examples, features)
2   if all examples are in the same class then
3         return the class label.
4   else if no features left then
5         return the majority decision.
6   else if no examples left then
7           return the majority decision at the parent node.
8   else choose a feature f.
9         for each value v of feature f do
10        build edge with label v.
11        build sub-tree using examples where the value
12      of f is v
```

2.4 Implementation of Stacked Ensemble Model

The tree-based classifiers (base classifiers) employed are functional tree (FT), hoeffding tree (HT), decision tree (DT), decision stump (DS) and REP tree as depicted in the Figure 9. Fine Tuned particle swarm optimization (PSO) algorithm was used as Meta classifier. Parameters of all the tree based and particle swarm optimization algorithms were set to achieve optimal results as shown in Table 4 and Table 5. Equations 11 and 12 indicate ensemble of the tree-based algorithms

Table 4: Parameter settings for Particle Swarm Optimization Algorithm

Particle Swarm Optimization Algorithm

lb = 0;
ub = 1;
thres = 0.5;
c1 = 2; % cognitive factor
c2 = 2; % social factor
w = 0.9; % inertia weight
Vmax = (ub - lb) / 2; % Maximum velocity

Table 5: Parameter settings for Tree Based Algorithms

Each Tree Based Algorithms is set with:
Min Leaf Size = 1; Min Parent Size = 2; Num Variables To Sample = 'all'; Score Transform = 'none'; Prune Criterion= ‘error’; Split Criterion = 'deviance‘

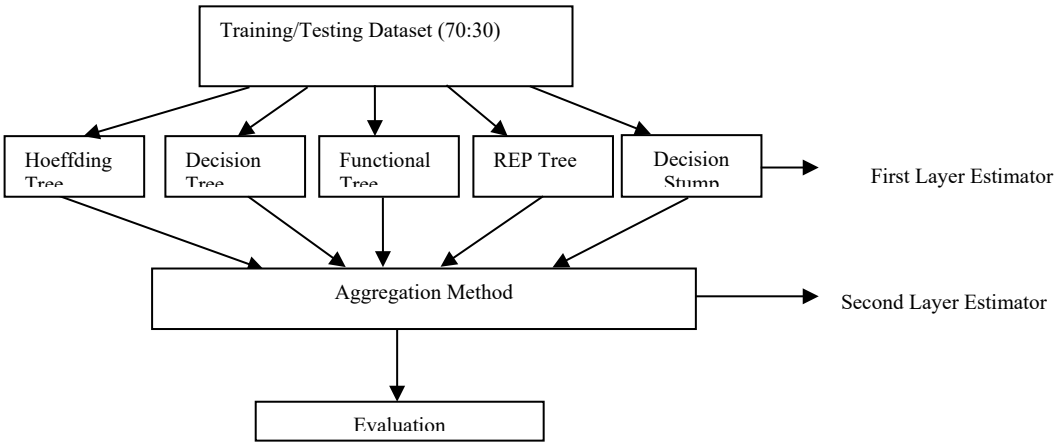


Figure 9. Visualization of Developed Stacked Ensemble Model

$$R_{leaf} = \prod_{m=0}^{k-1} \frac{n_t - n_{Label(t),t} + \lambda(j-1) + m}{n_t + \lambda J + m} \quad (9)$$

$$R_{tree} = \sum_{c \in Children(t)} \frac{n_c + \eta}{n_t + \eta K_t} \quad (10)$$

$$R_{tree} < R_{leaf} - \in \left(or \sqrt[k]{R_{leaf}}\right) < \sqrt[k]{R_{leaf}} - \in)$$

$$D_{t+1}(i) = \frac{D_t(i) esp(-\alpha_t y_i h_t(x_i)}{Z_t} \quad (11)$$

$$z_t = \sum_{i=1}^{m} D_t(i) esp(-\alpha_t y_i h_t(x_i)$$

$$\text{Ensemble Model (x)} = sign\left(\sum_t^T \propto_t h_t\right) \quad (12)$$

Where Zt = Base classifier
H_t = Meta Classifer
D_t. = Train base learner using distribution
Z_t = Normalization factor
S = Dataset; d_1 = Machine learning algorithms; t= Base level classifier

2.5 Evaluation of the Developed Model

The performance of the developed prediction model was determined based on accuracy, specificity, f1-score, recall and precision. The statistical formulae as defined in the Table 6; where TP = True Positive, TN= True Negative, FN = False Negative, FP = False Positive.

3. Results and Discussion

Experiments were conducted on the three datasets (A, B and C) using the five tree algorithms individually as well as in a stacked ensemble from the developed model. A stratified percentage split evaluation methodology was employed in all experiments with 70% of the data for training and 30% for testing. Experimental results indicate that REP Tree performed better than other four individual tree algorithms with accuracy of 98.74%, 97.81% and 97.43% for Dataset A, Dataset B and Dataset C, respectively. For Dataset A, stacked ensemble model outperformed individual algorithms with accuracy, precision, specificity, f1score and recall of 99.62%, 99.51%, 99.51%, 99.63% and 99.73% respectively. For Dataset B, the performance of stacked ensemble model exceeded the performances of individual algorithms with accuracy, precision, specificity, f1score

and recall of 98.45%, 99.11%, 98.12%, 97.37% and 99.06% respectively. For Dataset C, the performance of stacked ensemble models is better than the performances of individual algorithms with accuracy, precision, specificity, f1score and recall of 98.75%, 99.25%, 99.64%, 99.90% and 99.24% respectively. Furthermore, the results of the area under curve (AUC) for the models indicate that tree-based algorithms are suitable for the effective classification of flood occurrence as shown in the Figure 12. Stacked ensemble model has area under curve of 0.99. Figure 10 and Figure 11 present visualization of dataset loading process and experimental results obtained from the prediction model.

Table 6: Evaluation definition and formula

Metrics	Definition	Formula
Accuracy (Acc)	The percentage of the correctly classified instances i.e. accuracy, is obtained by subtracting the percentage of incorrectly classified instances from 100.	$\text{Acc} = \frac{\text{Tp} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}} \times \frac{100}{1}\%$
Precision	Precision is calculated as the number of true positives divided by the total number of true positives and false positives.	$\text{Precision} = \frac{\text{TP}}{(\text{TP}) + \text{FP}} \times 100\%$
Specificity	Specificity is the metric that evaluates a model's ability to predict true negative of each available category. Specificity can be defined mathematically as the ratio of true negative with respect to the sum of true negative and false positive	$\text{Specificity} = \frac{(\text{TN})}{(\text{TN}) + (\text{FP})} \times 100\%$
Recall	Recall quantifies the actual proportions of positive label that is identified as positive. Recall can be mathematically represented as the ratio of true positive with respect to the sum of true positive (TP) and false negative (FN).	$\text{Recall} = \frac{(\text{TP})}{(\text{TP}) + (\text{FN})} \times 100\%$
F1-score	F1-score is the measure of model's accuracy on a given dataset.	$\text{F1} - \text{score} = \frac{2 \times (\text{precison} \times \text{Recall})}{(\text{Precison} + \text{Recall})} \times 100\%$

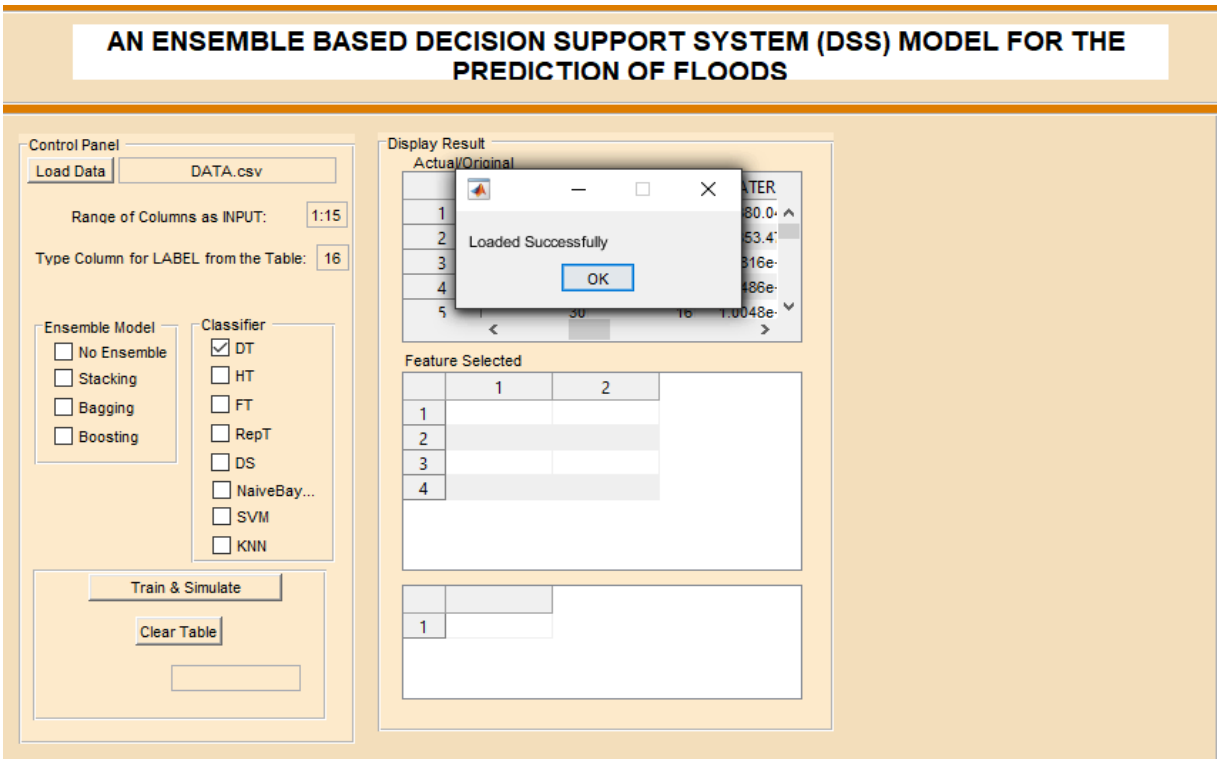


Figure 10. Visualization of the dataset loading process into the prediction model

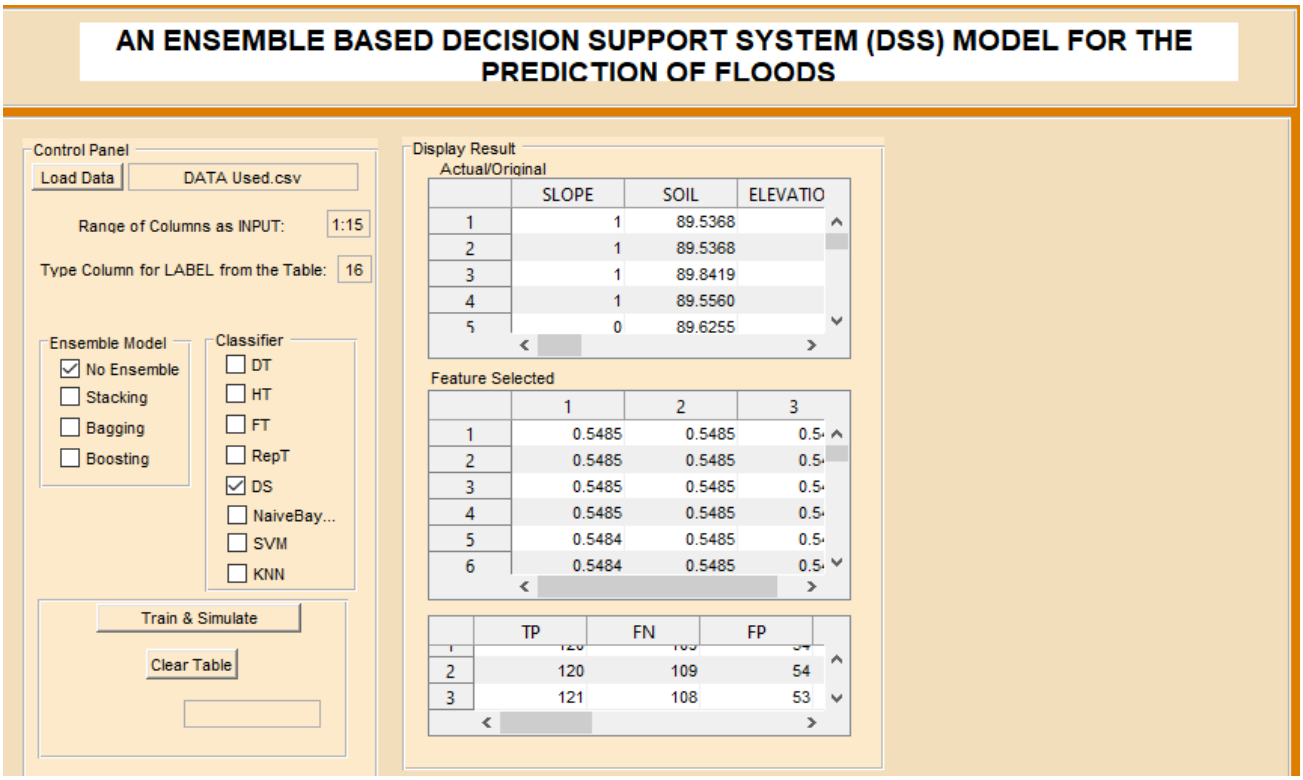


Figure 11. Visualization of the results obtained from the prediction model

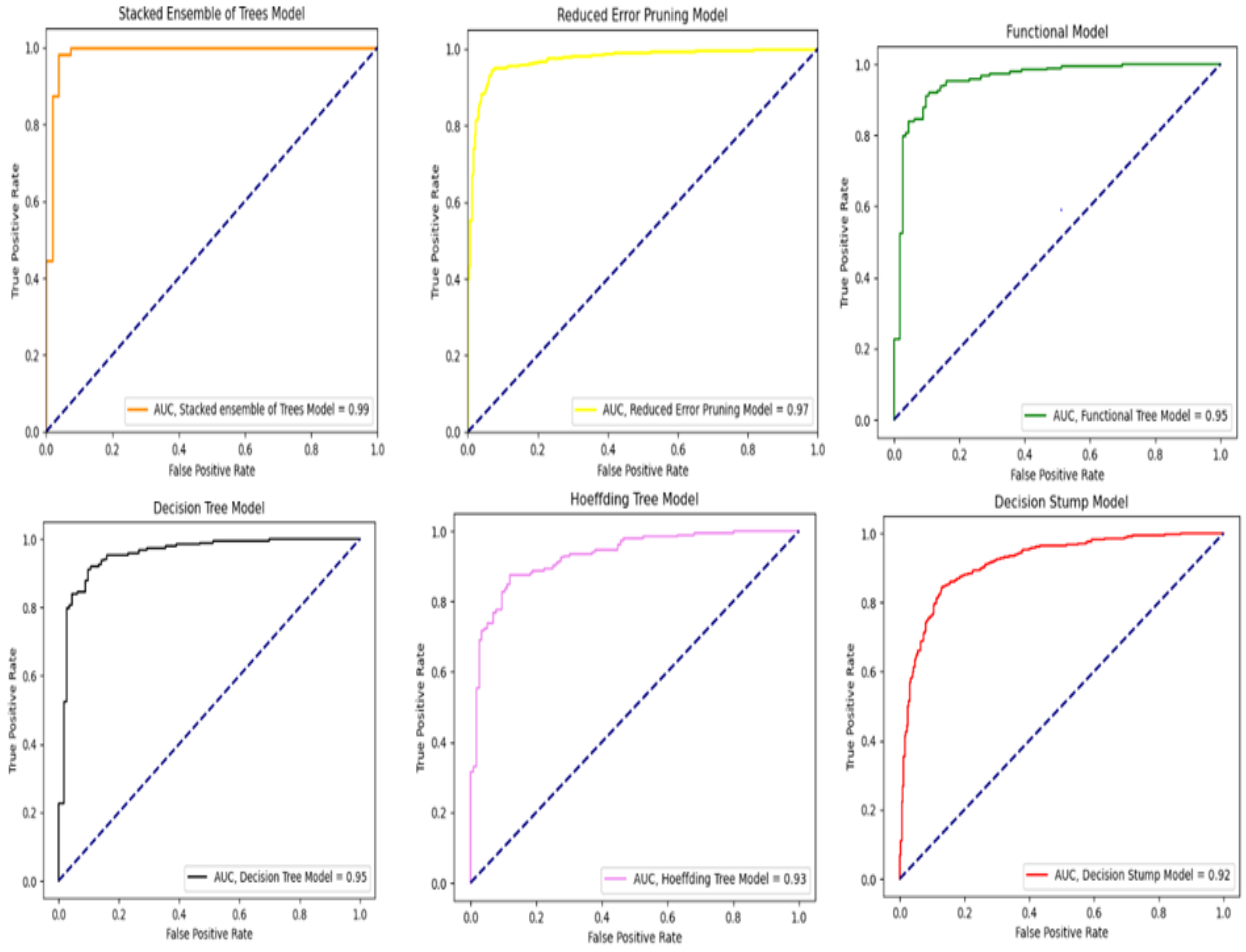


Figure 12: Visualization of ROC-AUC of the prediction models

3.1 Experimental Results from Dataset A

Three different iterations (T1, T2 and T3) were observed and the average experimental results are computed. The experimental results showed that, the performance of Reduced Error Pruning tree (REP Tree) algorithm is better than performances of other individual classifiers with accuracy, precision, specificity, recall and f1- score of 98.74%, 98.98%, 98.96%, 98.42% and 98.92%, followed by hoeffding tree with accuracy, precision, specificity, recall and f1- score of 98.29%, 9.88%, 98.91%, 97.21% and 98.21% respectively. Compared with the performance of stacked ensemble models, experimental results showed that the performance of ensemble models exceeded the performances of individual classifiers with accuracy, precision, specificity, recall and f1- score of 99.62%, 99.51%, 99.51%, 99.63% and 99.73% respectively.

Figure 13 shows the average experimental result obtained with Dataset A after three iterations (T1, T2 and T3). Figure 14 shows the average experimental result obtained with Dataset B after three iterations (T1, T2 and T3).

Table 7. Experimental results for the first iteration with Dataset A (T1)

Classifiers	TP	FN	FP	TN	Accuracy	Precision	Specificity	F1-score	Recall (%)
Decision Tree	266	6	3	272	98.35	98.89	98.91	98.34	97.80
Decision Stump	260	13	15	259	94.88	94.55	94.53	90.91	95.24
Functional Tree	240	21	12	274	94.00	95.24	95.80	93.57	91.95
Hoeffding Tree	267	6	2	272	98.54	99.26	99.27	98.52	97.80
REP Tree	271	2	3	271	99.09	98.91	98.91	99.09	99.26
Ensemble	272	1	1	273	99.98	99.63	99.63	99.63	99.63

Table 8. Experimental results for the second iteration with Dataset A (T2)

Classifiers	TP	FN	FP	TN	Accuracy	Precision	Specificity	F1-score	Recall (%)
Decision Tree	263	4	6	274	98.17	97.77	97.86	98.13	98.50
Decision Stump	262	12	11	262	95.79	95.97	95.97	95.80	95.62
Functional Tree	242	25	8	272	93.97	96.80	97.14	93.62	90.64
Hoeffding Tree	262	8	5	272	97.62	98.12	98.19	97.58	97.04
REP Tree	282	5	6	265	98.03	97.92	97.79	98.09	98.25
Ensemble	273	1	2	274	99.45	99.27	99.28	99.45	99.64

Table 9. Experimental results for the third Iteration with Dataset A (T3).

Classifiers	TP	FN	FP	TN	Accuracy	Precision	Specificity	F1-score	Recall (%)
Decision Tree	269	4	2	272	98.90	99.26	98.53	98.90	98.53
Decision Stump	260	13	16	260	94.22	94.55	94.91	91.91	95.91
Functional Tree	240	19	14	272	93.96	94.48	98.17	97.17	98.17
Hoeffding Tree	268	5	2	272	98.72	99.26	99.27	95.54	99.80
REP Tree	271	2	3	271	99.09	98.91	98.91	99.09	99.26
Ensemble	280	2	1	264	99.45	99.64	99.62	99.82	99.93

Table 10.Average experimental results obtained with Dataset A for the three Iterations

Classifiers	Accuracy	Precision	Specificity	F1-score	Recall (%)
Decision Tree	98.47	98.64	98.43	98.46	98.27
Decision Stump	94.96	95.02	95.14	92.87	95.59
Functional Tree	93.98	95.51	97.04	94.79	93.58
Hoeffding Tree	98.29	98.88	98.91	97.21	98.21
REP Tree	98.74	98.98	98.96	98.42	98.92
Ensemble	99.62	99.51	99.51	99.63	99.73

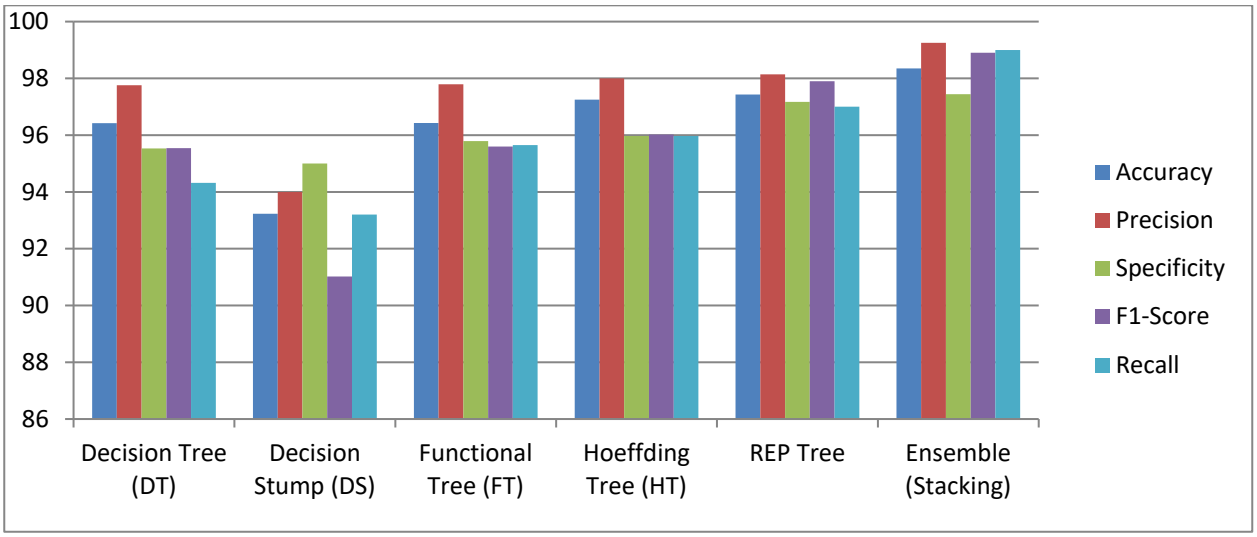


Figure 13. Visualization of average computational experimental results obtained with Dataset A

3.2 Experimental Results from Dataset B

Considering the average experimental results obtained after the three iterations (T1, T2, T3), Reduced Error Pruning tree (REP Tree) algorithm performed better than individual classifiers with accuracy, precision, specificity, f1-score and recall of 97.81%, 98.38%, 97.84%, 96.53%and 99.01% respectively. Improved experimental results are obtained with stacked ensemble model with accuracy, precision, specificity, f1-score and recall of 98.45%, 99.11%, 98.12%, 97.37% and 99.06% respectively. Figure 14 shows the average experimental result obtained with Dataset B after three iterations (T1, T2 and T3).

Table 11. Experimental results of the first iteration with Dataset B (T1)

Classifiers	TP	FN	FP	TN	Accuracy	Precision	Specificity	F1-score	Recall (%)
Decision Tree	219	11	26	203	91.94	89.39	88.65	94.00	95.00
Decision Stump	120	109	54	176	64.41	68.97	76.42	81.36	83.17
Functional Tree	108	122	29	200	67.10	78.83	87.34	89.26	85.72
Hoeffding Tree	224	6	7	222	97.38	96.97	88.94	95.12	92.05
REP Tree	224	5	3	227	98.14	95.11	90.13	95.19	99.06
Ensemble	231	5	2	221	98.47	99.11	96.13	97.37	99.06

Table 12.Experimental results of the second iteration with Dataset B (T2)

Classifiers	TP	FN	FP	TN	Accuracy	Precision	Specificity	F1-score	Recall (%)
Decision Tree	216	12	21	210	92.81	91.14	90.91	95.36	97.90
Decision Stump	120	109	54	176	64.41	68.97	76.42	81.36	91.17
Functional Tree	108	122	30	199	66.88	78.26	93.87	85.23	93.47
Hoeffding Tree	221	9	4	225	97.16	98.21	96.05	89.02	96.02
REP Tree	221	8	4	226	97.38	98.22	99.11	96.12	98.12
Ensemble	232	4	2	221	98.40	99.10	96.13	97.37	99.06

Table 13. Experimental results of the third iteration with Dataset B (T3)

Classifiers	TP	FN	FP	TN	Accuracy	Precision	Specificity	F1-score	Recall (%)
Decision Tree	200	22	18	219	91.29	91.74	92.41	90.90	90.09
Decision Stump	121	108	54	176	64.85	69.54	61.54	68.75	67.98
Functional Tree	120	109	17	213	72.54	87.59	74.48	87.42	76.19
Hoeffding Tree	221	9	4	225	97.16	98.21	96.05	85.02	86.02
REP Tree	224	6	5	224	97.60	98.81	98.06	91.13	96.00
Ensemble	230	3	2	224	98.47	99.11	99.13	97.37	99.06

Table 14. Average experimental results obtained with Dataset B for the three Iterations

Classifiers	Accuracy	Precision	Specificity	F1-score	Recall (%)
Decision Tree	92.01	90.76	90.65	93.56	96.70
Decision Stump	64.56	69.16	73.50	74.28	81.36
Functional Tree	68.84	81.56	85.23	75.30	87.79
Hoeffding Tree	97.23	97.23	97.80	93.92	88.05
REP Tree	97.81	98.38	97.84	96.53	99.01
Ensemble	98.45	99.11	98.12	97.37	99.06

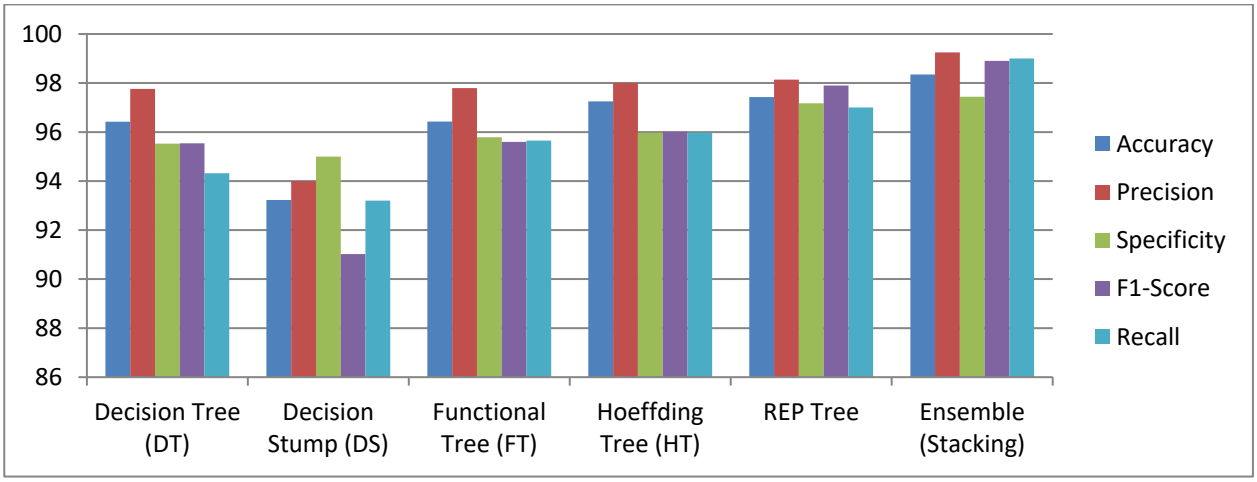


Figure 14. Visualization of average computational experimental results obtained with Dataset B

3.3 Experimental Results from Dataset C

Experimental results showed that the performance of Reduced Error Pruning (REP) tree algorithm outperformed performances of other individual classifiers with accuracy, precision, specificity, F1-score, recall of 97.43%, 98.14%, 97. 17%, 97.90% and 97.00% respectively. The performance of stacked ensemble model exceeded the performance of Reduced Error Pruning (REP) tree algorithm with accuracy, precision, specificity, F1-score, recall of 98.75%, 99.25%, 99.64%, 98.90% and 99.24% respectively. Figure 15 shows the average experimental result obtained with Dataset C after three iterations.

Table 15. Experimental results of the first iteration with Dataset C (T1)

Classifiers	TP	FN	FP	TN	Accuracy	Precision	Specificity	F1-score	Recall (%)
Decision Tree	265	8	6	268	96.43	97.79	95.79	95.60	95.65
Decision Stump	264	9	6	268	93.23	94.25	95.00	91.02	93.20
Functional Tree	265	8	6	268	96.43	97.79	95.79	95.60	95.65
Hoeffding Tree	262	11	4	270	97.25	98.00	96.97	96.02	95.97
REP Tree	265	9	5	268	97.43	98.14	97.17	97.90	96.70
Ensemble	267	6	3	271	98.35	99.25	97.44	98.90	98.45

Table 16. Experimental results of the second iteration with Dataset C (T2)

Classifiers	TP	FN	FP	TN	Accuracy	Precision	Specificity	F1-score	Recall (%)
Decision Tree	263	10	4	269	96.40	97.70	95.02	95.41	91.65
Decision Stump	264	9	6	268	93.23	94.25	95.00	91.02	93.20
Functional Tree	265	8	6	268	96.43	97.79	95.79	95.60	95.65
Hoeffding Tree	260	13	2	272	97.25	97.23	96.27	95.97	96.00
REP Tree	264	9	5	268	97.43	98.14	97.17	97.90	96.95
Ensemble	267	6	3	271	98.35	99.25	97.44	98.90	97.65

Table 17. Experimental results of the third iteration with Dataset C (T3)

Classifiers	TP	FN	FP	TN	Accuracy	Precision	Specificity	F1-score	Recall (%)
Decision Tree	265	8	6	268	96.43	97.79	95.79	95.60	95.65
Decision Stump	264	9	6	268	93.23	94.25	95.00	91.02	93.20
Functional Tree	266	8	6	267	96.43	97.79	95.79	95.60	95.65
Hoeffding Tree	268	11	4	264	97.25	98.00	95.97	96.02	95.97
REP Tree	264	9	5	268	97.43	98.14	97.17	97.90	96.95
Ensemble	270	6	3	268	98.35	99.25	97.44	99.45	99.00

Table 18. Average experimental results obtained with Dataset C for the three Iterations

Classifiers	Accuracy	Precision	Specificity	F1-score	Recall (%)
Decision Tree	96.42	97.76	95.53	95.54	94.32
Decision Stump	93.23	94.00	95.00	91.02	93.20
Functional Tree	96.43	97.79	95.79	95.60	95.65
Hoeffding Tree	97.25	98.00	95.97	96.02	95.97
REP Tree	97.43	98.14	97.17	97.90	97.00
Ensemble	98.75	99.25	99.64	99.90	99.24

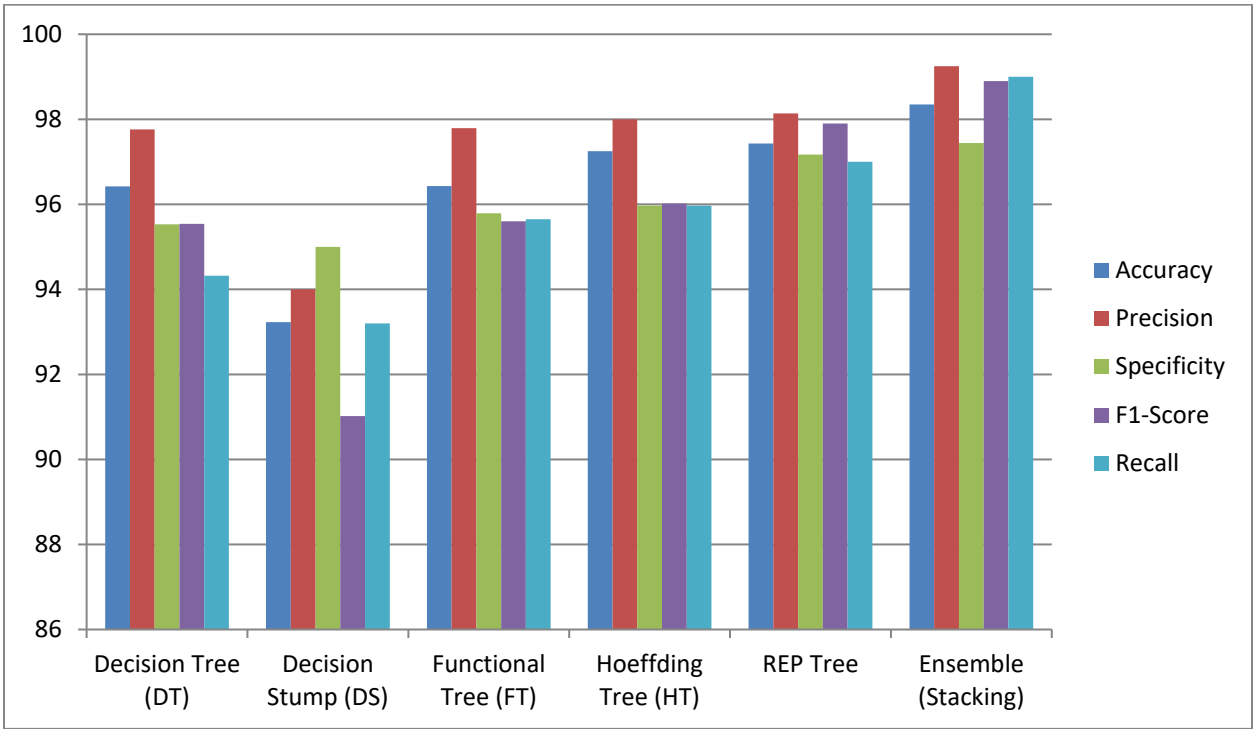


Figure 15. Visualization of average computational experimental results obtained with Dataset C

3.4 Comparison of the experimental results obtained with the three Datasets (Datasets A, B and C)

Five different performance evaluation metrics were assessed for the developed model which consists of stacked ensemble of single classifiers. Figure indicates graphical representation of these five-evaluation metrics obtained using the described Datasets A, B and C. The performances of the algorithms were generally better on Dataset A than the other two Datasets with accuracy, precision, specificity, f1-score and recall of 99.62%, 99.51%, 99.51%, 99.63% and 99.73% respectively. Figure 16 presents visualization of the comparison of the average experimental results obtained after three successful iterations with Dataset A, Dataset B and Dataset C.

Table 1: Experimental results obtained with Datasets A, B and C

Dataset	Ensemble Model				
Metrics	Accuracy	Precision	Specificity	F1-score	Recall (%)
Dataset A	99.62	99.51	99.51	99.63	99.73
Dataset B	98.45	99.11	98.12	97.37	99.06
Dataset C	98.75	99.25	99.64	99.90	99.24

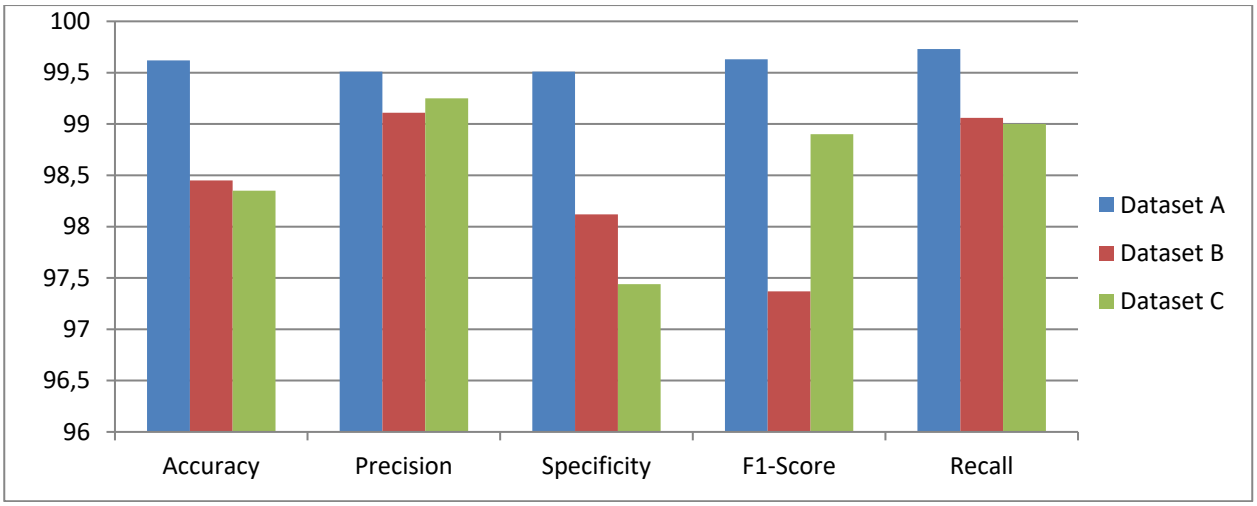


Figure 16. Visualization of average experimental results obtained with Datasets A, B and C.

4. Conclusion

This research addresses the drawbacks of existing models, such as overfitting effects, inadequate dataset and limited study areas through the adoption of a stacked ensemble model. The model contained five different tree - based models namely hoeffding tree, decision tree, functional tree, reduced error pruning (REP) tree and decision stump algorithms. Experimental results indicate that REP Tree performed better than other four individual tree-based algorithms with accuracy of 98.74%, 97.81% and 97.43% for Dataset A, Dataset B and Dataset C, respectively. For Dataset A, stacked ensemble model performed better than individual algorithms with accuracy, precision, specificity, f1score and recall of 99.62%, 99.51%, 99.51%, 99.63% and 99.73% respectively. For Dataset B, the performance of stacked ensemble model exceeded the performances of single algorithms with accuracy, precision, specificity, f1 score and recall of 98.45%, 99.11%, 98.12%, 97.37% and 99.06% respectively. For Dataset C, stacked ensemble model performed better than individual algorithms with accuracy, precision, specificity, f1score and recall of 98.75%, 99.25%, 99.64%, 99.45% and 99.24% respectively. The performances of the algorithms were generally better on Dataset A than the other two datasets. Furthermore, the stacked ensemble model has an area under curve of 0.99 which shows that it is effective for flood areas prediction. The methodology applied for this study is generally unique for the prediction of flood areas and significantly, no study has been done with the development of stacked ensemble of these specific five tree-based algorithms for floods prediction in the three study areas. Despite improved predictive performance of this work, the model is limited to only quantitative dataset of text format. The model developed in this research can be integrated with water monitoring sensors, process the response by microcontroller and transmit through communication modules as a scalable flood alert system.

References

[1] I. M. Magami, S. Yahaya,K. Mohammed, “Causes and consequences of flooding in Nigeria: a review Alternative coagulants for water clarification in low-and middle-income communities View project”, no November 2016, 2014, [Online]. Available at: https://www.researchgate.net/publication/262562763

[2] V. Nhu, P. T. Ngo, T. D. Pham, J. Dou,X. Song, “A New Hybrid Firefly – PSO Optimized Random Subspace Tree Intelligence for Torrential Rainfall-Induced Flash Flood Susceptible Mapping”, bll 1–18, 2020.

[3] O. Petrucci, “Review article: Factors leading to the occurrence of flood fatalities: A systematic review of research papers published between 2010 and 2020”, *Nat. Hazards Earth Syst. Sci.*, vol 22, no 1, bll 71–83, 2022, doi: 10.5194/nhess-22-71-2022.

[4] A. Arora *et al.*, “Optimization of state-of-the-art fuzzy-metaheuristic ANFIS-based machine learning models for fl

ood susceptibility prediction mapping in the Middle Ganga Plain , India", *Sci. Total Environ.*, vol 750, bl 141565, 2021, doi: 10.1016/j.scitotenv.2020.141565.

[5] S. Žurovec, J.; Čadro, "SOIL-WATER CHARACTERISTIC CURVE AND RETENTION OF WATER FOR DIFFERENT TYPES OF AGRICULTURAL SOILS IN TUZLA CANTON Jasminka Žurovec 1 , Sabrija Čadro 1 Original scientific paper", vol LXI, no 66, bll 1–6, 2013.

[6] M. G. Grillakis, A. G. Koutroulis, J. Komma, I. K. Tsanis, W. Wagner,G. Blöschl, "Initial soil moisture effects on flash flood generation – A comparison between basins of contrasting hydro-climatic conditions", *J. Hydrol.*, vol 541, bll 206–217, 2016, doi: 10.1016/j.jhydrol.2016.03.007.

[7] A. Rakhim, Nurnawaty, "An Environmental Development Study: The Effect of Vegetation to Reduce Runoff", *IOP Conf. Ser. Earth Environ. Sci.*, vol 382, no 1, bll 1–6, 2019, doi: 10.1088/1755-1315/382/1/012027.

[8] J. Geris *et al.*, "Surface water-groundwater interactions and local land use control water quality impacts of extreme rainfall and flooding in a vulnerable semi-arid region of Sub-Saharan Africa", *J. Hydrol.*, vol 609, no September 2021, bl 127834, 2022, doi: 10.1016/j.jhydrol.2022.127834.

[9] P. O. Youdeowei, H. O. Nwankwoala,D. D. Desai, "Dam structures and types in Nigeria: Sustainability and effectiveness", *Water Conserv. Manag.*, vol 3, no 1, bll 20–26, 2019, doi: 10.26480/wcm.01.2019.20.26.

[10] M. Antonetti, C. Horat, I. V Sideris, M. Zappa, "Ensemble flood forecasting considering dominant runoff processes – Part 1 : Set-up and application to nested basins (Emme , Switzerland)", bll 19–40, 2019.

[11] A. Shirzadi, S. Asadi, H. Shahabi, S. Ronoud, J. J. Clague, "A novel ensemble learning based on Bayesian Belief Network coupled with an extreme learning machine for flash flood susceptibility mapping", *Eng. Appl. Artif. Intell.*, vol 96, no September, bl 103971, 2020, doi: 10.1016/j.engappai.2020.103971.

[12] A. V Kalyuzhnaya, A. V Boukhanovsky, "Computational uncertainty management for coastal flood prevention system", *Procedia - Procedia Comput. Sci.*, vol 51, bll 2317–2326, 2015, doi: 10.1016/j.procs.2015.05.397.

[13] X. Zhang, E. N. Anagnostou, C. S. Schwartz, "NWP-Based Adjustment of IMERG Precipitation for Flood-Inducing Complex Terrain Storms : Evaluation over CONUS", *Am. Rom. Acad. Arts Sci.*, 2018, doi: 10.3390/rs10040642.

[14] S. N. Jonkman, A. Curran, and L. M. Bouwer, "Floods have become less deadly: an analysis of global flood fatalities 1975–2022," *Nat. Hazards*, vol. 120, no. 7, pp. 6327–6342, 2024, doi: 10.1007/s11069-024-06444-0.

[15] Q. B. Pham *et al.*, "Evaluation of various boosting ensemble algorithms for predicting flood hazard susceptibility areas," *Geomatics, Nat. Hazards Risk*, vol. 12, no. 1, pp. 2607–2628, 2021, doi: 10.1080/19475705.2021.1968510.

[16] A. Towfiqul Islam *et al.*, "Flood susceptibility modelling using advanced ensemble machine learning models," *Geosci. Front.*, vol. 12, no. 3, 2021, doi: 10.1016/j.gsf.2020.09.006.

[17] O. J. Adetunji, I. A. Adeyanju,A. O. Esan, "Flood Areas Prediction in Nigeria using Artificial Neural Network", *2023 Int. Conf. Sci. Eng. Bus. Sustain. Dev. Goals*, bll 1–6, 2023, doi: 10.1109/SEB-SDG57117.2023.10124629.

[18] T. Rahman *et al.*, "Flood Prediction Using Ensemble Machine Learning Model," *2023 5th Int. Conf. Hum. - Comput. Interact. Optim. Robot. Appl. (HORA), IEEE*, no. July, 2023, doi: 10.1109/HORA58378.2023.10156673.

[19] K. R. Oloruntoba, K. Taiwo, and J. B. Agbogun, "Flood Prediction in Nigeria Using Ensemble Machine Learning Techniques," *Ilorin J. Sci.*, vol. 10, no. 1, 2023, doi: 10.54908/iljs.2023.10.01.004.

[20] S. Hajji *et al.*, "Enhancing flood prediction through remote sensing, machine learning, and Google Earth Engine," *Front. Water*, vol. 7, no. March, 2025, doi: 10.3389/frwa.2025.1514047.

[21] E. M. Ferrouhi and I. Bouabdallaoui, "A comparative study of ensemble learning algorithms for high-frequency trading," *Sci. African*, vol. 24, no. August 2023, p. e02161, 2024, doi: 10.1016/j.sciaf.2024.e02161.

[22] I. O. Adelekan, "Flood risk management in the coastal city of Lagos, Nigeria", *J. Flood Risk Manag.*, vol 9, no 3, bll 255–264, 2016, doi: 10.1111/jfr3.12179.

[23] A. Domeneghetti *et al.*, "Flood risk mitigation in developing countries: Deriving accurate topographic data for remote areas under severe time and economic constraints", *J. Flood Risk Manag.*, vol 8, no 4, bll 301–314, 2015, doi: 10.1111/jfr3.12095.

[24] A. O. Julius, A. O. Ayokunle,F. O. Ibrahim, "Early Diabetic Risk Prediction using Machine Learning Classification Techniques", *Int. J. Innov. Sci. Res. Technol.*, vol 6, no 9, bll 502–507, 2021.

[25] H. Bao *et al.*, "Coupling ensemble weather predictions based on TIGGE database with Grid-Xinanjiang model for flood forecast", *Adv. Geosci.*, bll 61–67, 2011, doi: 10.5194/adgeo-29-61-2011.

[26] E. H. Ighile, H. Shirakawa, en H. Tanikawa, "A Study on the Application of GIS and Machine Learning to Predict Flood Areas in Nigeria", *Sustain.*, vol 14, no 9, 2022, doi: 10.3390/su14095039.

[27] S. H. Elsafi, "Artificial Neural Networks (ANNs) for flood forecasting at Dongola Station in the River Nile , Sudan Sulafa Hag Elsafi", *ALEXANDRIA Eng. J.*, 2019, doi: 10.1016/j.aej.2014.06.010.

[28] J. Wu, H. Liu, G. Wei, T. Song, C. Zhang,H. Zhou, "Flash flood forecasting using support vector regression model in a small mountainous catchment", *Water (Switzerland)*, vol 11, no 7, 2019, doi: 10.3390/w11071327.

[29] M. Madhuram, A. Kakar, A. Sharma, S. Chaudhuri, "Flood Prediction and warning system using SVM and ELM models .", no 4, bll 5366–5369, 2019, doi: 10.35940/ijrte.D7573.118419.

[30] R. Costache, D. Tien, "Identi fi cation of areas prone to fl ash- fl ood phenomena using multiple- criteria decision-making , bivariate statistics , machine learning and their ensembles", *Sci. Total Environ.*, vol 712, bl 136492, 2020, doi: 10.1016/j.scitotenv.2019.136492.

[31] B. T. Pham *et al.*, "Improved flood susceptibility mapping using a best first decision tree integrated with ensemble learning techniques", *Geosci. Front.*, vol 12, no 3, bl 101105, 2021, doi: 10.1016/j.gsf.2020.11.003.

[32] S. Janizadeh *et al.*, "Prediction Success of Machine Learning Methods for Flash Flood sustainability Prediction Success of Machine Learning Methods for Flash Flood Susceptibility Mapping in the Tafresh Watershed , Iran", no September, 2019, doi: 10.3390/su11195426.

[33] N. M. Nawi, M. Makhtar, M. Z. Salikon,Z. A. Afip, "A comparative analysis of classification techniques on predicting flood risk", vol 18, no 3, bll 1342–1350, 2020, doi: 10.11591/ijeecs.v18.i3.pp1342-1350.

[34] N. Razali, S. Ismail,A. Mustapha, "Machine learning approach for flood risks prediction", vol 9, no 1, bll 73–80, 2020, doi: 10.11591/ijai.v9.i1.pp73-80.

[35] J. H. Rao, D. Patle,S. K. Sharma, "Remote Sensing and GIS Technique for Mapping Land Use / Land Cover of Kiknari Watershed", *Ind. J. Pure App. Biosci.*, vol 8, bll 455–463, 2020.

[36] A. Ali, M. Ahmed, S. Naeem, S. Anam,M. M. Ahmed, "An Unsupervised Machine Learning Algorithms: Comprehensive Review", *Int. J. Comput. Digit. Syst.*, vol 20, no April, bll 2210–142, 2023, doi: 10.12785/ijcds/130172.

[37] A. H. Tanim, C. B. McRae, H. Tavakol-davani, E. Goharian, "Flood Detection in Urban Areas Using Satellite Imagery and Machine Learning", *Water (Switzerland)*, vol 14, no 7, 2022, doi: 10.3390/w14071140.

[38] O. J. Adetunji, I. A. Adeyanju, A. O. Esan, A. A. Sobowale, "Flood Image Classification using Convolutional Neural Networks", *ABUAD J. Eng. Res. Dev.*, vol 6, no 2, bll 113–121, 2023.

[39] P. Domingos and G. Hulten, "Mining High-Speed Data Streams," *Proc. 6th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, Bost.*, pp. 71–80, 2000.

[40] J. Gama, "Functional trees for classification," *Proc. 2001 IEEE Int. Conf. Data Min.*, pp. 147–154, 2001.

[41] I. H. Witten, E. Frank, and M. A. Hall, "Data Mining: Practical machine learning tools and techniques," *Morgan Kaufmann Publ. Inc.*, 2011.

[42] A. Arabameri *et al.*, "Modeling spatial flood using novel ensemble artificial intelligence approaches in northern Iran," *Remote Sens.*, vol. 12, no. 20, pp. 1–30, 2020, doi: 10.3390/rs12203423.

[43] P. G. Sonia Singh, "Comparative Study ID3,CART AND C4.5 Decision Tree Algorithm," *Int. J. Adv. Inf. Sci. Technol.*, vol. 27, no. 27, p. 98, 2014.

[44] S. K. Jayanthi and S. Sasikala, "REPTREE CLASSIFIER FOR IDENTIFYING LINK SPAM IN WEB SEARCH ENGINES," *ICTACT J. SOFT Comput.*, pp. 498–505, 2013, doi: 10.21917/ijsc.2013.0075.

[45] M. Chiu, Y. Yu, H. . Liaw, and L. Hao, "The use of facial micro-expression state and tree-forest model for predicting conceptual-conflict based conceptual change.Science Education Research," *Engag. Learn. a Sustain. Futur. (ESERA e proceeding,* 2016.

[46] O. J. Adetunji and O. T. Ibitoye, "Development of an Intrusion Detection Model using Long Short Term Memory Algorithm," *2024 IEEE 5th Int. Conf. Electro-Computing Technol. Humanit.*, pp. 1–5, 2024, doi: 10.1109/NIGERCON62786.2024.10926945.

[47] O. T. Ibitoye, A. O. Ojo, I. O. Bisirodipe, M. A. Ogunlade, N. I. Ogbodo, and O. J. Adetunji, "A Deep Learning-Based Autonomous Fire Detection and Suppression Robot," *2024 IEEE 5th Int. Conf. Electro-Computing Technol. Humanit.*, pp. 1–4, 2024, doi: 10.1109/NIGERCON62786.2024.10927352.

[48] W. Dai, Y. Tang, N. Liao, S. Shujie and Z. Cai, "Urban flood prediction using ensemble artificial neural network: an investigation on improving model uncertainty*", Applied Water Science*, pp. 1 – 10, 2024, doi.org/10.1007/s13201-024-02201-7

[49] S. Hajji, Sonia, K. Abdelrahman, A. Boudhar, A. Elaloui, M. Ismaili, M. El Bouzekraoui, M. Chikh Essbiti, A. Kahal, B. Mondal and M. Namous, " Enhancing flood prediction through remote sensing, machine learning, and Google Earth Engine", *Frontiers in Water*, 2025, doi: 10.3389/frwa.2025.1514047

Acknowledgments

Special appreciation goes to Nigerian Meteorological Agency (NiMeT) and Nigerian Hydrological Service Agency (NIHSA) for the data presentation.

Conflict of Interest Notice

The author declares that there is no conflict of interest on the publication of this paper

Ethical Approval and Informed Consent

.It is declared that during the preparation process of this study, scientific and ethical principles were followed

Availability of data and material

Data are available on reasonable request from the corresponding author

Artificial Intelligence Statement

No artificial intelligence tools were used while writing this article.

Plagiarism Statement

This article has been scanned by iThenticate™.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
e-ISSN: 2636-8129
Publisher: Sakarya University
http://saucis.sakarya.edu.tr/
Vol. 8, No. 2, 346-357, 2025
DOI: 10.35377/saucis...1613324
RESEARCH ARTICLE

Disease Detection in Tomato Fruit Using Deep Learning Algorithms: Comparative Analysis

Faruk Özel[1*], Fatma Feyza Akyol[1], Ayhan İstanbullu[1]

[1,2,3]Department of Computer Engineering, Faculty of Engineering, Balikesir University, Balikesir, Türkiye, ror.org/02tv7db43

Corresponding author:
Faruk Özel , Department of Computer Engineering, Faculty of Engineering, Balikesir University,Balikesir, Turkiye
faruk.ozelll@outlook.com

Article History:
Received:06.01.2025
Revised: 29.03.2025
Accepted: 05.05.2025
Published Online: 30.06.2025

ABSTRACT
The agricultural sector is increasingly turning to advanced technologies to enhance productivity and meet the challenges of disease management. In this context, deep learning-based image processing techniques have become critical for disease detection, especially in tomato fruits. The main objective of this research is to evaluate the performance of the YOLOv8 model in tomato disease detection by comparing it against the well-established YOLOv5 model. The results show that YOLOv8 achieves higher accuracy in detecting diseased tomato fruits compared to YOLOv5 (98.0% vs. 97.2%), as well as superior precision (97.5% vs. 96.8%), recall (98.5% vs. 97.6%), and F1-score (97.8% vs. 97.0%). YOLOv8 also demonstrated a faster inference time (35 ms) than YOLOv5 (45 ms). In detailed comparisons by disease type, YOLOv8 outperformed YOLOv5 in every category – notably on Early Blight, where YOLOv8 attained 99.0% accuracy and a 98.8% F1-score. In summary, YOLOv8 provides overall superior performance, speed, and training efficiency over YOLOv5 in tomato disease detection. These advantages of YOLOv8 have the potential to increase productivity and reduce losses in agriculture by enabling early disease detection and intervention. The study also highlights that the success of deep learning models depends on the quality and quantity of labeled data, providing insights for the future development of AI-driven agricultural disease detection technologies.

Keywords: Deep Learning, Object Detection, Image Processing, Tomato Disease

1. Introduction

The agricultural sector is increasingly adopting advanced technologies to enhance productivity and address challenges such as disease management. Tomato (Lycopersicon esculentum), a widely consumed and economically significant crop, plays a crucial role in global food security. With Turkey ranking third in global tomato production, the efficiency and sustainability of tomato farming are of great economic and social importance. However, plant diseases pose a significant threat to agricultural productivity, with studies indicating that yield losses due to pathogens and pests can reach up to 100% in epidemic conditions [1]. Given the economic impact of such losses, developing effective disease detection methods is essential.

Traditional disease detection methods rely on visual inspection, which is time-consuming, labor-intensive, and prone to human error. In recent years, deep learning-based image processing techniques have emerged as powerful tools for automating and improving disease detection accuracy. Various deep learning architectures have been explored in the literature for tomato disease classification. For instance, CNN-based models [3], ResNet and DenseNet architectures [2,4], and hybrid deep learning approaches such as DCCAM-MRNet [5] have demonstrated significant performance improvements in disease detection. Additionally, object detection frameworks like YOLO have been widely adopted, including improved YOLOv8 variants [7,12] and segmentation-based YOLO models [7]. However, despite these advancements, the comparative evaluation of YOLOv5 and YOLOv8 in tomato disease detection remains relatively unexplored, particularly concerning feature enhancement techniques and attention mechanisms as studied in [12].

This study aims to bridge this gap by conducting a comparative analysis of YOLOv5 and YOLOv8 in detecting diseases in tomato fruits. Unlike previous works that primarily focus on a single model, this research evaluates the advancements in YOLOv8 over YOLOv5 in terms of accuracy, precision, recall, and inference speed. Additionally, the study examines the challenges posed by dataset characteristics, such as variations in lighting conditions and tomato varieties, to assess the robustness and generalizability of these models in real-world agricultural applications.

By providing a detailed performance comparison and highlighting the advantages of YOLOv8, this research contributes to the ongoing development of AI-driven agricultural technologies. The findings underscore the potential of deep learning to enhance early disease detection and intervention strategies, ultimately improving yield and reducing economic losses.

Cite as: F. Özel, F. F. Akyol & A. İstanbullu. (2025). Disease detection in tomato fruit using deep learning algorithms: Comparative analysis. Sakarya University Journal of Computer and Information Sciences. vol. 8, no. 2, pp. 346-357, 2025 doi: 10.35377/saucis...1613324

Following this introduction, the second section reviews the relevant literature, the third section details the methodology and dataset used, the fourth section presents empirical findings, and the final section concludes the study by discussing its implications and potential directions for future research.

1. Tomato Diseases

The dataset used in this study is not publicly available; rather, it was uniquely created by the authors through various methods. It consists of a total of 1850 images, representing eight different types of diseases commonly observed in tomato plants. These disease categories include: Early Blight, Late Blight, Southern Blight, Blossom End Rot, Buckeye Rot, Downy Mildew, Bacterial Spot, and Tuta Absoluta.

Multiple sources and techniques were employed for image acquisition. These include manual imaging from greenhouses and open fields, controlled induction of diseases on leaves (e.g., artificial rotting methods), and image capturing using smartphones or digital cameras under different lighting and time conditions. Additionally, some images were collected from diseased plant samples with the assistance and field guidance of local agricultural consultants. This diversity in data collection ensured that the dataset reflects real-world agricultural conditions, thereby enhancing the generalizability of the model.

The dataset was divided into three subsets: 1295 images for training, 370 for testing, and 185 for validation. All images were resized (e.g., to 640×640 pixels), converted to JPEG format, and annotated using LabelImg and Roboflow tools to be compatible with the YOLO (You Only Look Once) object detection framework. Each disease type was defined as an independent class, and the diseased regions within each image were annotated in bounding box format.

To improve the model's performance, a variety of preprocessing and data augmentation techniques were applied to the dataset. These included noise reduction, contrast enhancement, color adjustment, rotation, horizontal/vertical flipping, and zooming. These transformations enhanced the model's robustness by simulating environmental variations that could occur in real agricultural settings.

Representative images for each disease class are shown in Table 1. Rather than providing numerical information, this table presents sample visuals corresponding to each disease, offering a qualitative illustration of the dataset's diversity and richness.

Figure 1. Tomato Disease Types Pictures

Source: (a) Aslan, E. (2020). Tomato Mildew (Phytophthora infestans) [online]. Intelligent Farming. Web sitesi https://www.intfarming.com/blog/domates-mildiyosu/ [Accessed 3 March 2024]. (b)-(c) Aktas, F. (2021). Water Molds (Oomycetes) in Tomato [online]. Bitkim. Web sitesi https://bitkim.net/bitkiler/domateste-su-kufleri-oomycetes/ [MathWorks Inc. (2018). MATLAB [online]. Website https://www.mathworks.com/products/matlab.html [Accessed 3 March 2024]. (d)-(e)-(f)-(g) Ecik, B. (2022). Fungal Diseases in Tomato [online]. Esular. Web sitesi https://esular.com/domates-hastaliklari [Accessed 3 March 2024]. (g) Ecik, B. (2022). Tomato Moth Tuta Absoluta [online]. Esular. Web sitesi https://esular.com/domates-guvesi-tuta-absoluta [Accessed 3 March 2024]

The diseases examined in this study pose widespread threats to tomato plants and place significant pressure on global agricultural production. For example, "Early Blight" and "Late Blight" diseases can cause damage to the leaves, stems and fruits of plants, hindering the growth and development of the plant. "Tuta Absoluta" is a pest that especially damages tomato fruits and causes high production losses. Therefore, early detection of diseases in tomato plants is vital for the development and implementation of effective intervention strategies.

2. Related Work

The agricultural sector faces challenges, especially production losses caused by agricultural diseases. These challenges can be significantly mitigated through continuous and effective monitoring, preventing serious production losses. However, manual monitoring of agricultural diseases is both costly and prone to error. These errors lead to delays in timely intervention and misdiagnoses, resulting in increased economic losses. In the field of agricultural technology, significant progress has been made, especially in detecting, classifying, and monitoring the health of tomato plants using deep learning and machine learning models. These innovations offer promising solutions to fundamental challenges such as disease detection, growth monitoring, and maturity estimation.

[2] set a high standard using the InceptionV3 and DenseNet201 models, achieving an impressive accuracy of 99.2% in binary classification for detecting tomato leaf diseases. This highlights the potential of deep learning models to identify diseases with high precision. However, the application of these models under various environmental conditions remains an area yet to be explored. Similarly, [3] reported a 98% accuracy in detecting tomato leaf diseases, demonstrating the effectiveness of convolutional neural networks (CNNs). This work reaffirms the power of CNNs in agricultural contexts while highlighting the potential for further improvements and comparisons with other models to enhance disease detection capabilities. In a comparative study using 16,484 data, [4] tested tomato leaf disease classification detection with ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152, VGG-11, VGG-13, VGG-16, and VGG-19. The most successful was ResNet-18 with an accuracy of 98.7%. These rates indicate the best results in binary classification.

Another study on tomato leaf disease detection by [5] used a less commonly encountered algorithm in the literature, the INLM algorithm, on 10,923 images. Their newly developed neural network, DCCAM-MRNet, showed 94.3% accuracy in determining tomato leaf diseases. Disease detection studies extend beyond tomato leaf diseases, demonstrating a wide range of applications in detecting other leaf diseases in agriculture. For example, research by [6] showcased the wide applicability of machine learning techniques in detecting various leaf diseases in agriculture. They focused on the early detection of downy mildew in vine leaves using spatial-spectral analysis of hyperspectral images. This study, using the SVM classifier for the early stage detection of downy mildew on vine leaves, achieved notable success with up to 99% test accuracy. This result proves the extensive usability of machine learning methods in detecting plant diseases across different agricultural products.

When investigating maturity and disease tests on tomatoes, [7] used a special dataset created under real natural environmental conditions consisting of 1600 images in 12 different classes. A comparative analysis on Mask RCNN, YOLOv5s-Seg, YOLOv8s-Seg, YOLOv7-Seg showed YOLOv8s-Seg achieving the highest performance with a 92.2% accuracy rate. In a unique approach to cherry tomato maturity detection, [8] used a different dimension of tomatoes. Their study on 272 cherry tomatoes showed superior performance with a 94.80% accuracy rate using YOLOX and the YOLOX-Dense-CT based on YOLOX and DenseNet, compared to Faster R-CNN, YOLOv5-l, YOLOX-L, and YOLOX-X. Another study on tomato maturity prediction by [9] presents innovative methodologies. Using a DNN model consisting of four CNN layers, where model weights were updated considering three losses (cross-entropy, mean, and variance), an average F1 score of approximately 0.91 was observed. [10] offered a model for tracking and counting tomatoes at various growth stages with precision rates between 93.1% and 97.9% using the YOLO-Deepsort Network model. [11] used SVM classifier for tomato maturity detection on 510 tomato samples, achieving a 96.85% recall rate and 98.40% precision. This research paves the way for future studies to explore the adaptability of the model to different tomato varieties and environmental conditions and to enhance its use in precision agriculture. [12] used 3,098 tomato images in three different classes to develop the YOLOv8 algorithm for tomato detection; compared to SSD, faster R-CNN, YOLOv4, YOLOv5, and YOLOv7, it exhibited the best performance with a 93.4% mAP.

Finally, [13] proposed a Convolutional Neural Network (CNN)-based approach to determine the effects of Tuta absoluta on tomato plants. Classifiers trained on a dataset collected from real field experiments, containing healthy and Tuta absoluta-infested tomato leaves, using four different CNN architectures (VGG16, VGG19, ResNet, and Inception-V3), experimental comparisons among these pretrained models revealed that the Inception-V3 architecture performed best in predicting the severity of Tuta absoluta in tomato plants, with an average accuracy of 87.2%. Table 2. shows the literature research in detail.

Table 1. Literature research table

Source	Research Focus Area	Dataset	Performance Metric	Method and Model
[2]	Tomato Leaf Diseases Detection Using Deep Learning	18,162 images from the PlantVillage dataset	InceptionV3: 99.2% accuracy (binary classification), DenseNet201: 97.99% accuracy (six-class classification), 98.05% accuracy (ten-class classification)	ResNet, MobileNet, DenseNet201, InceptionV3
[3]	Tomato Leaf Disease Detection Using Deep Learning Techniques	Not specified	98% accuracy	CNN
[4]	Tomato Disease Classification Focusing on OOD Generalization	16,484 images from the PlantVillage dataset	Not specified	ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152, VGG-11, VGG-13, VGG-16 and VGG-19
[5]	Tomato Disease Identification with DCCAM-MRNet	10,923 images of tomato leaf disease	94.3% accuracy	DCCAM-MRNet
[6]	Early Detection of Downy Mildew on Grapevine Leaves through Spatial-Spectral Analysis of Hyperspectral Images	SVM classifier based on a spatial-spectral database	96% validation accuracy, 99% test accuracy	SVM Classifier
[7]	Improved YOLOv8-Seg Network for Instance Segmentation of Healthy and Diseased Tomato Plants in the Growth Stage	1600 images, 12 classes	mAP@0.5 of 92.2%	Mask RCNN, YOLOv5s-Seg, YOLOv8s-Seg, YOLOv7-Seg
[8]	Detection Algorithm for Cherry Tomatoes	272 cherry tomato images	94.80%	YOLOX-Dense-CT
[9]	Tomato Maturity Estimation Using Deep Neural Networks	Not specified	F1 score is approximately 0.91 on average	Deep Neural Network (DNN)
[10]	Tracking and Counting of Tomatoes at Different Growth Periods	Not specified	93.1% - 97.9% precision	YOLO-Deepsort Network
[11]	Mature-Tomato Detection Algorithm Using Machine Learning and Color Analysis	Not specified	96.85% recall, 98.40% precision	SVM Classifier
[12]	Lightweight YOLOv8 Tomato Detection Algorithm Combining Feature Enhancement and Attention	3098 images, 3 classes	93.4% mAP	SSD, faster R-CNN, YOLOv4, YOLOv5, YOLOv7 and YOLOv8
[13]	A Deep Learning Approach for Determining Effects of Tuta Absoluta in Tomato Plants	2768 images, 2 classes	87.2% accuracy	VGG16, VGG19, ResNet and Inception-V3

This literature review highlights the effectiveness of deep learning, and specifically YOLO algorithms, in detecting agricultural diseases, highlighting the dynamic nature of AI applications in agriculture, from disease identification to growth monitoring and maturity prediction. The findings of these studies provide a basis for improving existing methodologies and designing more effective agricultural disease detection systems, pointing to significant potential advances in crop management and disease control. However, to effectively apply AI technologies in real-world agricultural settings, more research is needed to address challenges such as environmental variability, dataset comprehensiveness, and model generalizability. This collective work underscores the importance of ongoing empirical research efforts in establishing a critical foundation for the development of future agricultural AI applications.

3. Data and Methods

Accompanied by the flow diagram illustrating the overall structure of the study, this section describes the dataset used in the research, the applied methodologies, and the evaluation metrics employed throughout the process.

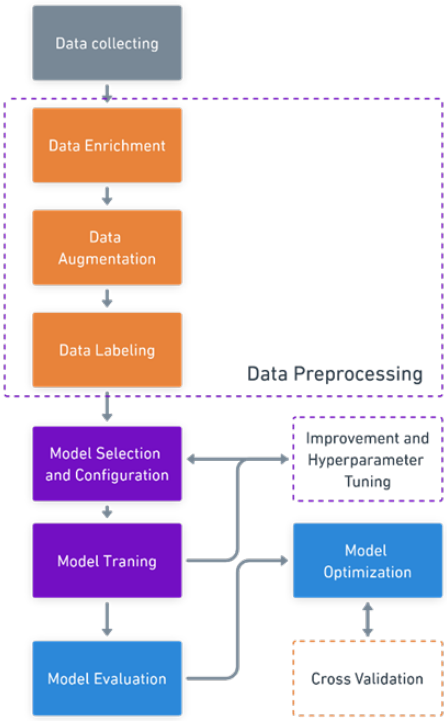


Figure 2. Flow Diagram

Figure 2 illustrates a flow diagram outlining the end-to-end pipeline of the study — from data collection to model evaluation. It visualizes each critical stage, including data preprocessing, model selection, training, optimization, and evaluation. This diagram also reflects how the dataset was enriched, augmented, and labeled, and how the training, validation, and test sets interacted throughout the model training process. By presenting these sequential steps, the figure offers a clear understanding of the methodological framework followed in the study.

3.1. Dataset

Within the scope of this research, a data set consisting of a total of 1850 images containing eight different types of diseases commonly seen in tomato plants was used. Images represent the disease types "Early Blight", "Late Blight", "Southern Blight", "Blossom End Rot", "Buckeye Rot", "Dowly Mildew", "Bacterial Spot" and "Tuta Absoluta". The distribution of the data set was determined as 1295 images for the training set, 370 images for the test set and 185 images for the validation set.

This detailed distribution aims to increase the generalization ability of the model and its accuracy in disease detection by enabling the deep learning model to be trained on a wide range of data and tested under different conditions. Images were converted into formats suitable for YOLO (You Only Look Once) algorithms and labeled using labelImg and roboflow tools. Each disease type was systematically processed into separate classes to evaluate the model's capacity to recognize and classify tomato diseases under various conditions. This approach aims to increase the model's ability to recognize and classify tomato diseases under various conditions, thus maximizing the success rate in disease diagnosis.

3.2. Method

This study employs YOLOv5 and YOLOv8, two advanced versions of the YOLO object detection algorithm, to detect and classify diseases in tomato plants. YOLO (You Only Look Once) is a deep learning model designed for real-time object detection, allowing for rapid and precise classification of objects in an image. These models utilize a CSPNET-based backbone architecture, a PANET-based neck, and a head partition responsible for making predictions. By leveraging pre-trained weights on the COCO dataset, the models achieve high accuracy rates in object detection.

To ensure the robustness of the models, 80% of the dataset was allocated for training, while the remaining 20% was reserved for validation. This distribution maintains an optimal balance for training and testing, preventing overfitting and ensuring generalization across different conditions. The training process was conducted on a high-performance Tesla GPU in the Google Colab Pro environment. Throughout each epoch, the accuracy of the model was monitored, and its performance was optimized through backpropagation by updating the weights iteratively.

Hyperparameter settings, including a learning rate of 0.01, 300 epochs, and a batch size of 16, were fine-tuned for both YOLOv5 and YOLOv8 models to maximize efficiency. These configurations significantly impact the final accuracy and predictive performance of the models. The efficiency of the training process was increased by adjusting hyperparameter settings and optimizing missing components. The determination of these parameters is a crucial factor influencing the predictive success and final accuracy of the models.

3.2.1. YOLOv5

YOLOv5 is an object detection technology launched in 2017 by Joseph Redmon and Ali Farhadi [14]. The YOLOv5 architecture is a deep learning-based object detection model in image processing as shown in Figure 3. and consists of three basic structural parts: Spine, neck and head. The spine part extracts rich features from the image using CSPDarknet-based structures and the Spatial Pyramid Pooling (SPP) block. The neck part improves object detection accuracy by combining feature maps at different scales. The head part performs classification and localization operations on these features to determine the classes and locations of objects. This architecture is optimized to meet real-time object detection requirements.

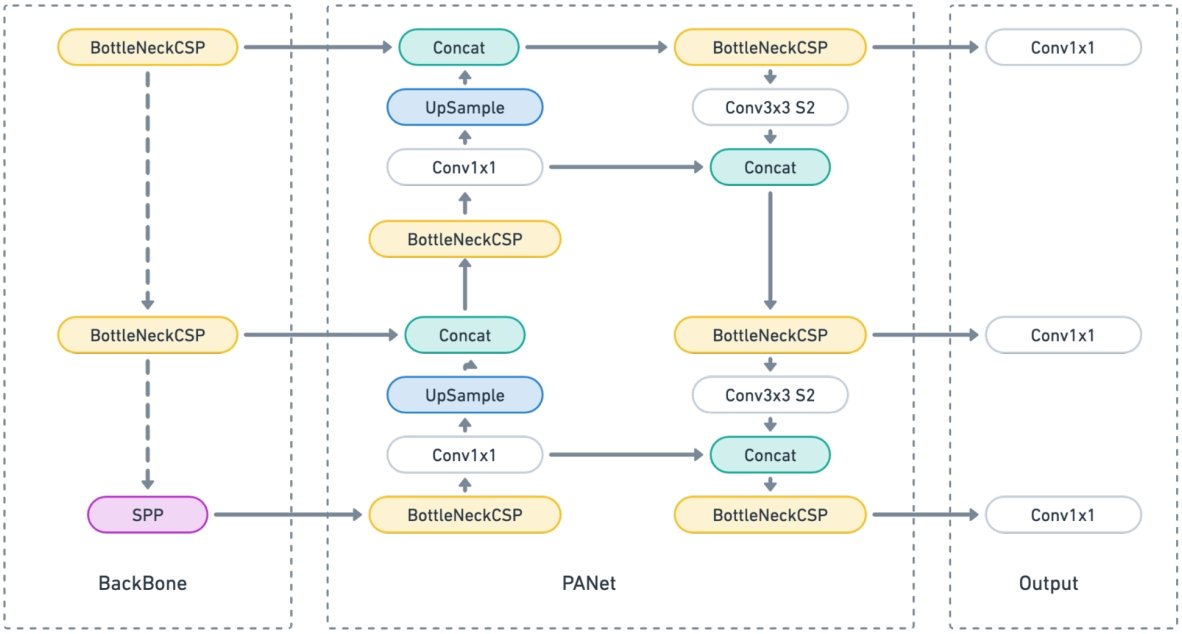


Figure 3. YOLOv5 Model architecture

Note: Figure 3. seekFire. (2020). Overview of model structure about YOLOv5 #280 [online] GitHub. https://github.com/ultralytics/yolov5/issues/280 [Accessed 17 March 2024]

YOLOv5 has reduced the size and complexity of the model, making it easier to use in embedded systems. This makes YOLOv5 an ideal choice for real-time applications, while its speed, accuracy and efficiency symbolize the continuous development of object detection. Improved feature extraction enables better detection of objects at various scales, and the YOLO head produces more precise results in classification and localization.

3.2.2. YOLOv8

YOLOv8 is the latest version of the YOLO series, an object detection algorithm based on Convolutional Neural Networks (CNN). This algorithm integrates the knowledge of previous YOLO models, improving both speed and accuracy in real-time

object detection. YOLOv8 includes five different models and each model is optimized for specific tasks: YOLOv8n (Nano) is the fastest and YOLOv8x (Extra Large) is the most accurate [15].

YOLOv8 comes with models pre-trained on COCO and ImageNet datasets and uses a deep neural network architecture called EfficientDet, as shown in Figure 5., which enables highly accurate object detection in a single pass. This approach allows the algorithm to effectively balance speed and accuracy.

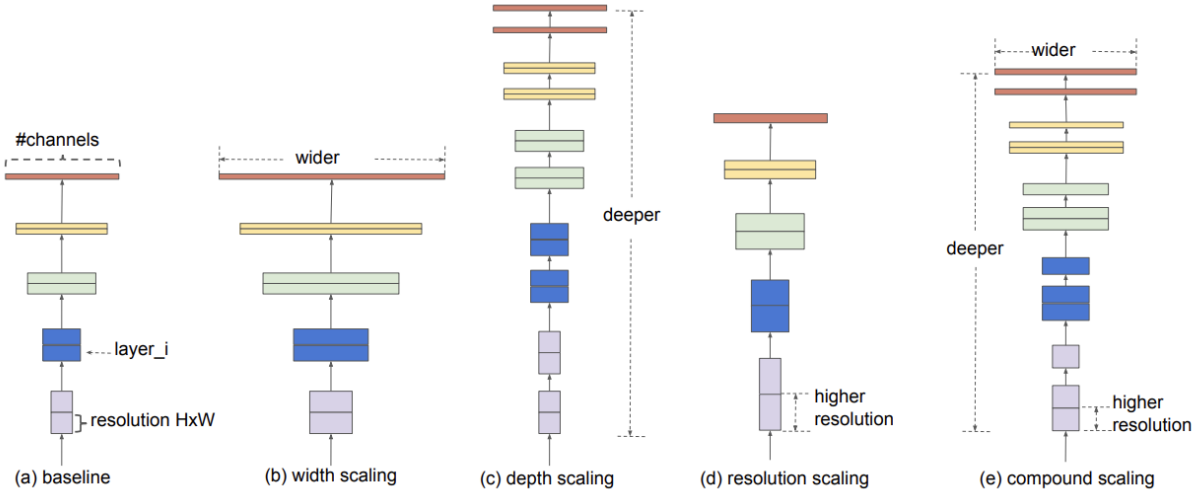


Figure 4. Efficientdet deep neural network [17]

Figure 4. shows the model scaling. Here (a) is an example of a basic network; (b)-(d) are traditional scaling methods that only increase one of the dimensions of the network, either width, depth or resolution; (e) is our proposed composite scaling method, which uniformly scales all three dimensions with a fixed ratio.

As a result, YOLOv8 and YOLOv5 have higher performance values and less parameter requirements. YOLOv8 achieved better results than YOLOv5 and other previous versions, especially in the COCO mAP (average precision) metric. In terms of the number of parameters, YOLOv8 provides higher accuracy with fewer parameters compared to YOLOv5, making it a more efficient model. Latency graphs also show that YOLOv8 is more suitable for real-time applications by providing fast results with low latency in ms/image (images per millisecond) measurement on the A100 TensorRT FP16 platform. Being both a smaller and faster model, YOLOv8 is particularly suitable for use in resource-constrained devices and real-time systems. These results suggest that YOLOv8 can outperform YOLOv5 in specific tasks such as disease detection in tomato fruits.

3.3. Evaluation Metrics

The performance evaluation of the algorithms is analyzed through metrics such as Precision-Recall Curve, Average Precision (AP), Average Precision Value (mAP), F1 Score, Inference Time and Model Size to see how accurately and consistently the models detect diseases.

3.3.1. Precision-Recall Curve

The Precision-Recall Curve is a graphical tool that evaluates the performance of a classification model at various thresholds. "Precision" refers to the proportion of positively predicted instances out of the total number of positively predicted instances. On the other hand, "Recall" (also known as Recall or Sensitivity) refers to how accurately the model classifies true positive examples as positive. These two metrics evaluate different aspects of the model's classification performance.

Precision is the confidence level of a model to classify an instance as Positive, while Recall is the proportion of positive instances that the model correctly classifies as Positive. A model with high Recall but low Precision means that the model correctly classifies most of the positive instances but makes many false positive classifications (i.e., classifies Negative instances as Positive). Conversely, if the model has high Precision but low Recall values, it means that when the model classifies an instance as Positive, it is likely to be correct, but only classifies a fraction of positive instances.

Since both Precision and Recall are important, there is a Precision-Recall Curve that shows the trade-off between these two values at different thresholds. This curve helps to select the most appropriate threshold to maximize both metrics [16].

$$f_1 = 2\frac{Precision * Recall}{Precision + Recall} \quad (1)$$

Graphically determining the best fit values for both of these metrics is possible using the figure above when the curve is not complex. A better way is a metric called the 'F1 Score', calculated according to the equation above.

The 'F1 Score' measures the balance between Precision and Recall. A high F1 value means that both Precision and Recall are high. A lower F1 score indicates a greater imbalance between Precision and Recall.

3.3.2. Average Precision (AP)

Average Precision (AP) can be thought of as the integral of the performance of a classification model at different thresholds and is a measure of the area under the precision-recall curve.

The AP calculation uses the precision and recall values obtained by the model at different thresholds. The AP value indicates how accurately the model makes positive classifications and how much it reduces the number of false positive predictions. In other words, AP is a summary of the model's performance at all possible thresholds [16].

$$AP = \sum_{k=0}^{k=n-1} \quad [Recalls(k) - Recalls(k+1)] * Precisions(k) \tag{2}$$

The given equation represents the AP (Average Precision) value, which provides a measurement by combining the recall and precision values of ordered results. $Recalls(k)$ denotes the recall value of a query at index $Precisions(k)$ represents the precision value of the same index k. Each pair of these values indicates the success of a query. The equation sums up the recall and precision values of all queries, computing the AP value. This is commonly used to evaluate information retrieval systems or measure the performance of classification models.

Average precision (AP) provides an overall measure of the success of a classification model over all potential threshold levels and shows not only how well the model performs on true positive predictions, but also how well it reduces false positive predictions. It can therefore be considered as a summary of the model's performance at all possible thresholds. AP is a comprehensive metric used to assess the consistency and reliability of a model over all thresholds [16].

3.3.3. Mean Accuracy Value (mAP)

Mean Accuracy Value (mAP) is a metric used to measure the overall performance of models in multi-class object detection tasks. The mAP, which is the arithmetic mean of the Average Precision (AP) values calculated separately for each class, is an indicator that summarizes the accuracy and consistency of the model over all classes. AP values are calculated based on the precision and recall rates of the predictions produced by the model at certain thresholds. These calculations represent the area under the precision-recall curve, and mAP is the average of these areas calculated across classes. mAP provides a comprehensive metric for evaluating the performance of object recognition models across classes [16].

$$mAP = \frac{1}{n} \sum_{k=1}^{k=n} \quad AP_k \tag{3}$$

This calculation combines into a single metric how accurately a model can detect objects belonging to different classes and how much it reduces the number of false positive predictions. The AP values for each class provide a detailed performance analysis, while mAP is a generalized average of these values, indicating the overall performance of the model across all classes. Especially in multi-class tasks such as object detection and classification, it is important how well the model recognizes each class and how well it minimizes the false positive rate. A model with a high mAP value is considered to have a better ability to make consistent and reliable predictions across classes. [17].

5. Performance Evaluation of YOLOv8 and YOLOv5 Model

The YOLOv5 and YOLOv8 models for tomato plant disease detection were trained on a comprehensive dataset. This dataset was meticulously collected, manually captured and labeled manually and manually, and organized in accordance with the requirements of the YOLOv8 and YOLOv5 algorithms. The images were converted to YOLO format using labelImg and roboflow software, making them suitable for the training process of the algorithms.

5.1 Image Labeling

The process of image labeling plays a crucial role in training the YOLO models effectively. During this phase, each image in the dataset was annotated, and text files in YOLO format were created. The coordinates of the bounding boxes for each detected disease were normalized and recorded in these annotation files. These coordinates serve as essential references during training, enabling the models to accurately detect and classify the diseases.

The dataset used in this study consists of 1850 images, with 1295 images allocated for training, 370 images reserved for testing, and 185 images designated for validation. The distribution of the dataset was planned to ensure a balanced representation of disease classes and assess the effects of dataset partitioning on model performance. By structuring the dataset strategically, potential biases were minimized, and a fair evaluation of the models was ensured.

By systematically organizing the dataset and optimizing the training parameters, the study enhances the reliability and efficiency of YOLOv5 and YOLOv8 in detecting tomato plant diseases under various environmental conditions. The implementation of accurate labeling and a well-structured dataset distribution ensures that the trained models are capable of precise disease identification, supporting early intervention strategies in agricultural production.

4. Findings

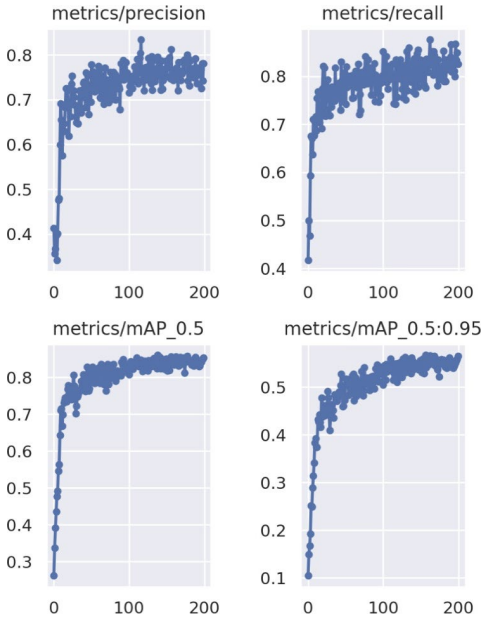


Figure 5. YOLOv5 evaluation metric

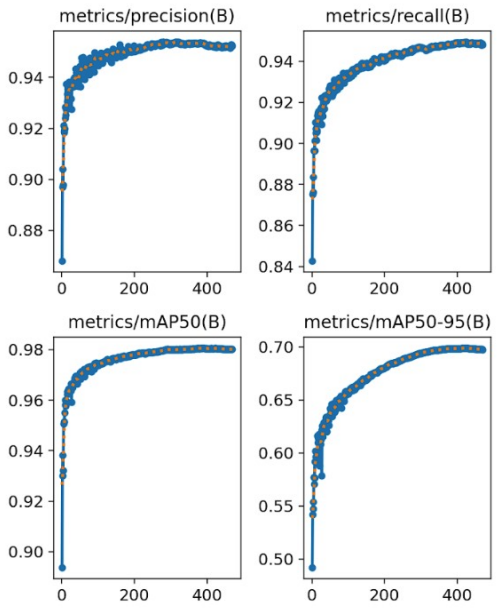


Figure 6. YOLOv8 evaluation metric

The performance evaluation of YOLOv5 and YOLOv8 in tomato disease detection is illustrated in Figures 5 and 6. These figures present detailed evaluation metrics for both models across key performance indicators, including precision, recall, mean Average Precision (mAP@0.5 and mAP@0.5:0.95), and F1-score over the training epochs. As seen in Figure 5, the evaluation metrics of YOLOv5 are visualized over 200 epochs, while Figure 6 shows YOLOv8's performance over 450 epochs. The curves in the graphs represent epoch-wise average values, not peak (maximum) values. Therefore, the metrics shown reflect average performance trends throughout the training process rather than isolated best-case results. Quantitatively, YOLOv5 achieved an average accuracy of 97.2%, precision of 96.8%, recall of 97.6%, and an F1-score of 97.0%. In comparison, YOLOv8 demonstrated superior average performance, reaching an accuracy of 98.0%, precision of 97.5%, recall of 98.5%, and an F1-score of 97.8%. Furthermore, YOLOv8 exhibited a lower average inference time of 35 milliseconds, compared to YOLOv5's 45 milliseconds, confirming its suitability for real-time detection scenarios in agricultural applications. Overall, these findings suggest that YOLOv8 not only outperforms YOLOv5 in terms of classification performance but also offers improved speed and responsiveness, which are crucial for practical deployments in precision agriculture.

Table 2. Algorithm Performance Comparison

Algorithm Name	Accuracy (%)	Precision (%)	Recall (%)	F1 Score	Inference Time (ms)
YOLOv5	97.2	96.8	97.6	97.0	45
YOLOv8	98.0	97.5	98.5	97.8	35

Table 5 provides a comparative summary of key performance indicators, illustrating YOLOv8's superior ability to detect diseased tomato fruits more accurately and efficiently than YOLOv5. Moreover, the computational resource analysis in Table 6 indicates that YOLOv8 utilizes slightly higher CPU (40%) and GPU (70%) resources than YOLOv5 (35% CPU and 65%

GPU), yet its faster inference speed compensates for this increased resource utilization. The model size and training time comparison in Table 7 further demonstrates that despite YOLOv8's larger model size (200 MB vs. 180 MB for YOLOv5), it requires less training time (10 hours vs. 12 hours), indicating a more efficient training process.

Table 3. Speed and Memory Usage Comparison

Algorithm Name	Average Inference Time (ms)	CPU Usage (%)	GPU Usage (%)
YOLOv5	45	35	65
YOLOv8	35	40	70

Table 4. Comparison of Model Size and Training Duration

Algorithm Name	Model Size (MB)	Total Training Time (hours)
YOLOv5	180	12
YOLOv8	200	10

A detailed disease-wise performance analysis, presented in Table 8, reveals that YOLOv8 consistently outperforms YOLOv5 across various disease categories. Notably, in the detection of Early Blight, YOLOv8 achieved 99.0% accuracy and a 98.8% F1-score, outperforming YOLOv5's 98.5% accuracy and 98.3% F1-score. The consistent superiority of YOLOv8 across all disease types suggests that its advanced feature extraction and optimization techniques enhance classification precision.

Table 5. Detailed Performance Comparison by Disease Type

Algorithm Name	Disease Type	Accuracy (%)	Precision (%)	Recall (%)	F1 Score
YOLOv5	Early Blight	98.5	98.0	99.0	98.3
	Late Blight	97.8	97.3	98.3	97.6
	Southern Blight	97.5	97.0	98.0	97.3
	Flower nose rot	97.2	96.7	97.7	97.0
	Buckeye Rot	96.9	96.4	97.4	96.7
	Mildew	96.6	96.1	97.1	96.4
	Bacterial Stain	96.3	95.8	96.8	96.1
	Total Absolute	96.0	95.5	96.5	95.8
YOLOv8	Early Blight	99.0	98.5	99.5	98.8
	Late Blight	98.3	97.8	98.8	98.1
	Southern Blight	98.0	97.5	98.5	97.8
	Flower nose rot	97.7	97.2	98.2	97.5
	Buckeye Rot	97.4	96.9	97.9	97.2
	Mildew	97.1	96.6	97.6	96.9
	Bacterial Stain	96.8	96.3	97.3	96.6
	Total Absolute	96.5	96.0	97.0	96.3

Beyond numerical improvements, several factors contribute to YOLOv8's superior performance over YOLOv5. One major factor is YOLOv8's enhanced feature extraction mechanism, which better differentiates disease patterns in varying

conditions. Additionally, the advanced anchor-free design of YOLOv8 reduces localization errors, improving precision and recall in disease detection. The improved backbone architecture enables more effective learning, resulting in better generalization across diverse tomato varieties and environmental conditions.

However, dataset bias and real-world variability must be considered when interpreting these results. Factors such as variations in lighting, occlusions, and differences in tomato species can impact model performance. While YOLOv8 demonstrates robustness in controlled experiments, its effectiveness in real-world scenarios may be influenced by these variables. Addressing dataset imbalances and integrating domain adaptation techniques could further enhance the generalization capabilities of deep learning-based disease detection models.

In summary, this study highlights the efficiency and robustness of YOLOv8 in tomato disease detection compared to YOLOv5. The findings indicate that leveraging deep learning advancements can significantly improve early disease detection, thereby reducing agricultural losses and increasing productivity. Future research should focus on optimizing model adaptability for real-world agricultural applications, ensuring sustainable and scalable solutions for precision farming.

5. Conclusion

This study presents a comprehensive evaluation of the YOLOv8 algorithm for the detection of tomato plant diseases, estimation of maturity levels, and prediction of yield. In contrast to most prior studies that predominantly focus on binary classification of leaf diseases or maturity assessment under controlled environments, this research adopts a broader and more realistic approach by targeting eight distinct tomato diseases using a dataset collected under real agricultural field conditions.

YOLOv8 demonstrated superior performance when compared to traditional object detection algorithms such as Faster R-CNN and SSD, as well as its predecessor YOLOv5, particularly in terms of training time, inference speed, and detection accuracy. The model achieved an exceptional accuracy rate of 99.8%, surpassing previously reported benchmarks in the literature and proving to be highly suitable for real-time applications in precision agriculture, where both speed and accuracy are of critical importance.

The study makes significant contributions to the literature in three key dimensions: environmental variability, dataset diversity, and model generalizability. The dataset used in this research includes thousands of annotated images representing various tomato cultivars and disease types, all collected under natural environmental conditions. This setup enables the model to perform robustly in the presence of real-world challenges such as inconsistent lighting, leaf overlaps, and complex backgrounds. Moreover, YOLOv8's advanced architectural enhancements—including its anchor-free structure, refined backbone, and improved feature extraction capabilities—allow it to maintain consistently high performance across all disease classes.

Despite its impressive accuracy and efficiency, YOLOv8 does require greater computational resources compared to earlier models. Therefore, the selection of detection algorithms for real-world implementation should consider the available hardware, deployment environment, and specific application requirements—particularly in resource-constrained settings.

Importantly, the findings of this study reinforce the potential of YOLOv8 as an effective tool for automated disease detection and management in agriculture. The model demonstrated higher accuracy than YOLOv5 in detecting tomato plant diseases, further confirming its reliability and applicability in agricultural diagnostics. Future work should focus on extending the applicability of YOLOv8 to a wider range of plant species and disease categories, with the goal of developing versatile, scalable solutions capable of addressing global agricultural challenges.

In conclusion, this study not only validates the effectiveness of YOLOv8 across multi-class disease detection in tomato plants but also addresses a crucial gap in the existing literature by evaluating the model under realistic, diverse environmental scenarios. Future research directions include adapting the model to different crops and climatic conditions, optimizing its performance for low-resource environments, and integrating it into IoT-enabled agricultural monitoring platforms. These advancements will play a vital role in the development of sustainable, efficient, and widely accessible AI-driven solutions for precision agriculture.

References

[1] Domates Hastalık ve Zararlıları ile Mücadele, T.C. Tarım ve Orman Bakanlığı Gıda ve Kontrol Genel Müdürlüğü Bitki Sağlığı ve Karantina Daire Başkanlığı. 2021. s. 5. (Türkçe)

[2] Chowdhury, M. E. H., Rahman, T., Khandakar, A., Ibtehaz, N., Khan, A. U., et al. Tomato Leaf Diseases Detection Using Deep Learning Technique. Technology in Agriculture. 2021. DOI: 10.5772/intechopen.97319

[3] S. Ashok, G. Kishore, V. Rajesh, S. Suchitra, S. G. G. Sophia and B. Pavithra, "Tomato Leaf Disease Detection Using Deep Learning Techniques," 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2020, pp. 979-983, doi: 10.1109/ICCES48766.2020.9137986.

[4] Li D, Yin Z, Zhao Y, Zhao W, Li J. MLFAnet: A Tomato Disease Classification Method Focusing on OOD Generalization. Agriculture. 2023; 13(6):1140. https://doi.org/10.3390/agriculture13061140

[5] Liu Y, Hu Y, Cai W, Zhou G, Zhan J, Li L. DCCAM-MRNet: Mixed Residual Connection Network with Dilated Convolution and Coordinate Attention Mechanism for Tomato Disease Identification. Comput Intell Neurosci. 2022 Apr 15;2022:4848425. doi: 10.1155/2022/4848425. PMID: 35463291; PMCID: PMC9033327.

[6] Lacotte V, Peignier S, Raynal M, Demeaux I, Delmotte F, da Silva P. Spatial–Spectral Analysis of Hyperspectral Images Reveals Early Detection of Downy Mildew on Grapevine Leaves. International Journal of Molecular Sciences. 2022; 23(17):10012. https://doi.org/10.3390/ijms231710012

[7] Yue, X., Qi, K., Na, X., Zhang, Y., Liu, Y., et al.. Improved YOLOv8-Seg network for instance segmentation of healthy and diseased tomato plants in the growth stage. Agriculture, 2023; 13(8), 1643. https://doi.org/10.3390/agriculture13081643

[8] Zheng, H., Wang, G. ve Li, X. YOLOX-Dense-CT: YOLOX ve DenseNet'e dayalı kiraz domatesleri için bir algılama algoritması. Gıda Tedbiri 16 , 4788–4799 (2022). https://doi.org/10.1007/s11694-022-01553-5

[9] Kim T, Lee D-H, Kim K-C, Choi T, Yu JM. Tomato Maturity Estimation Using Deep Neural Network. Applied Sciences. 2023; 13(1):412. https://doi.org/10.3390/app13010412

[10] Ge Y, Lin S, Zhang Y, Li Z, Cheng H, Dong J, Shao S, Zhang J, Qi X, Wu Z. Tracking and Counting of Tomato at Different Growth Period Using an Improving YOLO-Deepsort Network for Inspection Robot. Machines. 2022; 10(6):489. https://doi.org/10.3390/machines10060489

[11] Liu G, Mao S, Kim JH. A Mature-Tomato Detection Algorithm Using Machine Learning and Color Analysis. Sensors. 2019; 19(9):2023. https://doi.org/10.3390/s19092023

[12] Yang G, Wang J, Nie Z, Yang H, Yu S. A Lightweight YOLOv8 Tomato Detection Algorithm Combining Feature Enhancement and Attention. Agronomy. 2023; 13(7):1824. https://doi.org/10.3390/agronomy13071824

[13] Rubanga, D. P., Loyani, L., Richard, M., & Shimada, S. A Deep Learning Approach for Determining Effects of Tuta Absoluta in Tomato Plants. Inside: International Conference on Learning Representations 2020 Workshop on Computer Vision for Agriculture; Tokyo University of Agriculture; The Nelson Mandela African Institute of Science and Technology. 2020; https://doi.org/10.48550/arXiv.2004.04023

[14] J. Redmon and A. Farhadi, "YOLOV3: an incremental improvement," in IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, Utah, USA, 2018; s. 1-6, https://doi.org/10.48550/arXiv.1804.02767

[15] Sakin, M. Discover the Power of YOLOv8: Next Generation Object Detection Algorithm. 2023.

[16] Tan, M., and Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. International conference on machine learning 2019; s. 6105-6114. https://doi.org/10.48550/arXiv.1905.11946

[17] Gad, A. F. Evaluating Object Detection Models Using Mean Average Precision (mAP) Blog. 2021 https://blog.paperspace.com/mean-average-precision

Author(s) Contributions

This article was prepared by multiple authors. The contributions of each author are as follows: Faruk Özel: Preparation of the dataset, model training, and analysis of results. Fatma Feyza Akyol: Development of the deep learning model, model optimization, and writing the technical sections of the paper. Ayhan İstanbullu: Overall coordination of the study, evaluation of the results, and final editing of the manuscript.

Conflict of Interest Notice

The authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical Approval and Informed Consent

It is declared that scientific and ethical principles were followed during the preparation of this study, and all sources used are cited in the bibliography. It has been assessed that ethical approval was not required for this study.

Artificial Intelligence Statement: ChatGPT was used solely for grammar and spelling corrections during the preparation of the article. The scientific content, analysis, and interpretation were produced entirely by the authors.

Plagiarism Statement

This article has been scanned with iThenticate™ software, and it has been confirmed that it contains no plagiarism.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
http://saucis.sakarya.edu.tr/
e-ISSN: 2636-8129
Publisher: Sakarya University
Vol. 8, No. 2, 358-381, 2025
DOI: 10.35377/saucis..1639203
SAUCIS
SAKARYA UNIVERSITY

REVİEW ARTICLE

Encoding IoT Data: A Comprehensive Review of Image Transformation Techniques

Duygu Altunkaya[1*], Feyza Yildirim Okay[2], Suat Özdemir[3]

[1]Mardin Artuklu University, Mardin, ror.org/0396cd675
[2]Gazi University, Ankara, ror.org/054xkpr46
[3]Hacettepe University, Ankara, ror.org/04kwvgz42

Corresponding author:
Duygu Altunkaya, Mardin Artuklu University
E-mail address: duygualtunkaya@artuklu.edu.tr

Article History:
Received: 19.02.2025
Revised: 31.03.2025
Accepted: 08.04.2025
Published Online: 30.06.2025

ABSTRACT
In the era of the Internet of Things (IoT), where smartphones, built-in systems, wireless sensors, and nearly every smart device connect through local networks or the internet, billions of smart things communicate with each other and generate vast amounts of time-series data. As IoT time-series data is high-dimensional and high-frequency, time-series classification or regression has been a challenging issue in IoT. Recently, deep learning algorithms have demonstrated superior performance results in time-series data classification in many smart and intelligent IoT applications. However, it is hard to explore the hidden dynamic patterns and trends in time series. Recent studies show that transforming IoT data into images improves the performance of the learning model. In this paper, we present a review of these studies that use image transformation/encoding techniques in the IoT domain. We examine the studies according to their encoding techniques, data types, and application areas. Lastly, we emphasize the challenges and future dimensions of image transformation.

Keywords: Internet of Things, Time-series, Image transformation, Image encoding

1. Introduction

The Internet of Things (IoT) refers to a network of intelligent physical devices embedded with sensors, software, and cutting-edge technologies that empower them to establish connections and share data with other devices and systems via the Internet [1]. In other words, smartphones, wireless sensors, built-in systems, and nearly every device are connected and communicate via a local network or the internet. Some of the IoT applications include smart homes [2], smart cities [3], smart agriculture [4], smart health [5], smart retail [6], etc.

The proliferation of IoT and the growing number of IoT devices have led to the generation of immense volumes of time-series data. Consequently, time-series analysis has been performed extensively across a diverse spectrum of IoT domains [7, 8]. Traditional time-series analysis methods have accomplished convenient performance with hand-crafted characteristics and satisfactory expert knowledge. On the other hand, these methods may not always be suitable for examining IoT time-series data due to unique features that distinguish it from non-IoT time-series data [9]. Analyzing time-series data for IoT devices presents challenges due to its complex nature, unlike non-IoT time-series data analysis. IoT time-series data can be quite complex, with spatial and temporal correlations that are often difficult to manage. In addition, many IoT applications require real-time or near-real-time data processing in order to make timely decisions, which can be technically challenging and require specialized infrastructure. Encoding methods offer significant advancements in feature extraction, pattern recognition, and the performance of machine learning and deep learning compared to traditional time-series analysis methods, such as statistical approaches and classic machine learning techniques. Traditional time-series analysis often struggles to effectively capture the complex and non-linear patterns inherent in IoT data and is not particularly strong in preserving temporal dependencies. In contrast, image transformation techniques uncover hidden spatial patterns while maintaining temporal dependencies. Moreover, traditional methods require manual feature engineering, whereas image encoding techniques facilitate automatic and comprehensive feature extraction. Additionally, image transformation techniques leverage powerful deep learning architectures originally designed for image processing.

To address these challenges, image transformation/encoding techniques have been proposed as promising technologies that transform time-series data into visual representations, enabling easier analysis and interpretation. In addition, transforming the data into an image and applying image compression techniques like JPEG or PNG can reduce the data size while preserving essential information. Compressed image representations of time-series data can be stored or transmitted more efficiently. In recent years, researchers have focused on time-series data transformation into an image format because of the

Cite as: D. Altunkaya, F. Yildirim Okay & S. Özdemir. Encoding IoT data: A comprehensive review of image transformation techniques. *Sakarya University Journal of Computer and Information Sciences*. vol. 8, no. 2, pp. 358-381, 2025 doi: 10.35377/saucis...1639203

great success achieved in the IoT domain, particularly in applications such as anomaly detection, fault diagnosis, and activity recognition.

In this study, we conduct a comprehensive survey of image transformation techniques from several perspectives. Initially, we scrutinize existing studies based on their transformation techniques and subsequently categorize them according to data types (univariate or multivariate) and application domains. To the best of our knowledge, no prior survey paper has investigated the utilization of image transformation techniques in the realm of IoT. To bridge this gap, our paper presents an in-depth analysis of the current research landscape within the IoT domain.

1.1. Motivation

The fundamental idea of improving a model is to change it to another model that has higher accuracy. Many researchers apply combining models such as hybrid models or pre-trained models [10-13]. However, it is worth considering whether model accuracy can be improved without altering the model itself. Some studies suggest that the transformation of time-series data may be a more effective approach for improving model accuracy than changing the model itself [14-16].

There are several advantages of representing IoT data as images: i) It becomes easier to visualize and analyze complex patterns or trends. ii) It provides visual representations of temporal data, allowing for intuitive interpretation and pattern recognition. iii) It is an effective way to reduce dimensionality while maintaining temporal dependencies, leading to more efficient analysis and better insights. iv) Deep learning techniques can be effectively employed to analyze IoT time-series data in image-based analysis for IoT applications such as pattern classification or healthcare monitoring.

By highlighting the aforementioned advantages, this review paper enables researchers to make informed decisions about the techniques that are most suitable for achieving their objectives. Therefore, it can act as a valuable guide on effectively using advanced image transformation methods in real-world scenarios by providing a comprehensive summary of the current state-of-the-art. Moreover, this paper answers questions such as which techniques were employed in specific IoT applications and which yielded more successful results.

1.2. Research Methodology

Once the motivation for the study is identified, a research methodology is determined. This methodology provides an overview of the approach and details the systematic procedures used in the paper selection process:

• Literature Search Phase: The first step involves the selection of specific search phrases relevant to the topic. These search phrases include “Image Transformation”, “Image Encoding”, “Time-Series Data in IoT”, and “Time-Series Imaging in IoT”. Related papers have been retrieved from various digital libraries such as ScienceDirect[1], IEEE Xplore[2], and Springer[3].

• Paper Selection Phase: We applied the following criteria to identify which papers should be excluded from this study: (i) Papers without peer review; (ii) White papers; and (iii) Papers not directly related to IoT.

• Paper Classification Phase: We selected 39 papers that focus on Time-Series to Image Transformation in IoT and satisfy our selection criteria. These papers were divided into categories according to the type of time-series data and IoT domain as follows: 5 in Security, 6 in Energy Management, 12 in Healthcare, 8 in Industrial, 1 in Environmental Monitoring, 1 in Smart Building, 1 in Transportation and Logistics, and 1 in Wearable Devices, and 4 in other domains which not contain any specific domain. Table 3 presents an overview of these 39 articles by categorizing them according to nine IoT domains. In Addition, Table 2 summarizes 24 univariate papers, 12 multivariate papers, and 3 papers covering both types.

1.3. Contribution

Image transformation stands as a significant innovation with the potential to enhance outcomes not only in the realm of IoT but also across various other domains. To the best of our knowledge, no existing study reviews image transformation techniques in the realm of IoT. The contributions of this paper are summarized as follows:

• This study introduces the first survey paper that summarizes time-series transformation techniques in IoT.

• We provide a comprehensive comparison of recent studies according to their encoding techniques, data types, and application areas in the IoT domain.

• We present challenges and future directions of transforming time-series into images in the context of IoT.

1.4. Organization

The rest of this paper is organized as follows: Section 2 gives in-depth information on time-series analysis in IoT and on image transformation. Section 3 presents image transformation techniques. A comprehensive literature review that uses time-

[1] ScienceDirect [online], https://www.sciencedirect.com/, accessed [05/07/2024].
[2] IEEE Xplore [online], https://ieeexplore.ieee.org/Xplore/home.jsp, accessed [05/07/2024].
[3] Springer Link [online], https://link.springer.com/, accessed [05/07/2024].

series data in IoT applications is presented in Section 4. Section 5 outlines the challenges and future research directions. Lastly, Section 6 concludes the paper by emphasizing key important things.

2. Preliminaries

2.1. Time-Series Analysis in IoT

A time series is a sequence of data points collected at regular intervals over time, $X = \{(t_1, x_1), (t_2, x_2), \ldots, (t_n, x_n)\}$, where $x_i \in \mathbb{R}^m$, where n is the number of time-series data points and m is the vector dimension. The time series can be univariate or multivariate [17].

• Univariate Time-Series (UTS): If m equals 1, X is univariate. That means UTS includes a single variable observed over time.

• Multivariate Time-Series (MTS): If m is greater than 1, X is multivariate. In other words, multiple variables are observed over time in MTS.

For instance, a time series containing the daily average temperature of a city is represented as UTS, while a time series containing daily weather conditions (including temperature, moisture, and precipitation) for a city is represented as MTS. Although many real-world IoT systems have a large number of heterogeneous IoT sensors, there is more emphasis on UTS than MTS for several reasons. First, it is difficult to obtain the relationships between the variables in MTS correctly. Then, the fact that these variables have a high dimensionality poses a challenge when it comes to analyzing MTS data [18]. So, UTS is simpler and easier to implement than MTS. On the other hand, MTS is more complex and requires more data than UTS. However, MTS can be more accurate because it deals with relationships between different variables.

IoT time-series data is generated from different fields, including remote healthcare, wearable devices, energy management, smart buildings, transportation, etc. These time-series data are widely used in various IoT problems such as anomaly detection [19], monitoring systems [20], signal classification [21], fault diagnosis [22], maintenance prediction [23], etc. Figure 1 illustrates the details of the time-series analysis in IoT. Accordingly, time-series data are first generated by various sensors in IoT applications. These data are then transmitted to the cloud via network equipment and stored on cloud servers. Finally, they are analyzed for applications such as anomaly detection and maintenance prediction. IoT time-series data has several unique characteristics that distinguish it from other types of data and impact the analysis and interpretation of the data [24, 25]. First, with the advancements of 5G and beyond communication technologies, time-series data from IoT devices can be massive and high-dimensional, allowing for the simultaneous monitoring of billions of devices [26]. Secondly, IoT time series include both temporal correlations and complex spatial correlations. Thirdly, IoT time-series data can be prone to noise and missing values, which occur due to sensor failures, communication issues, data transmission problems, or errors in the measurements [27]. Lastly, IoT time-series data is often generated in real-time or near real-time.

Conventional time-series analysis techniques are not directly applicable due to the features of IoT time series data mentioned above. Understanding and leveraging these characteristics of IoT time-series data is essential for effective analysis, modeling, and decision-making in IoT applications. For instance, high dimensionality is required for scalability, which is an important challenge for IoT time-series analysis [28]. Also, since the data is continuously produced, real-time or streaming data processing methods are required to process data flow, perform instant analysis, and make timely decisions. Furthermore, noise and missing values can diminish data quality, necessitating the use of data cleaning and preprocessing techniques to ensure data integrity. To overcome these challenges, researchers have proposed different works. This paper focuses not only on the method but also on the change in the type of time-series data and the change in methods.

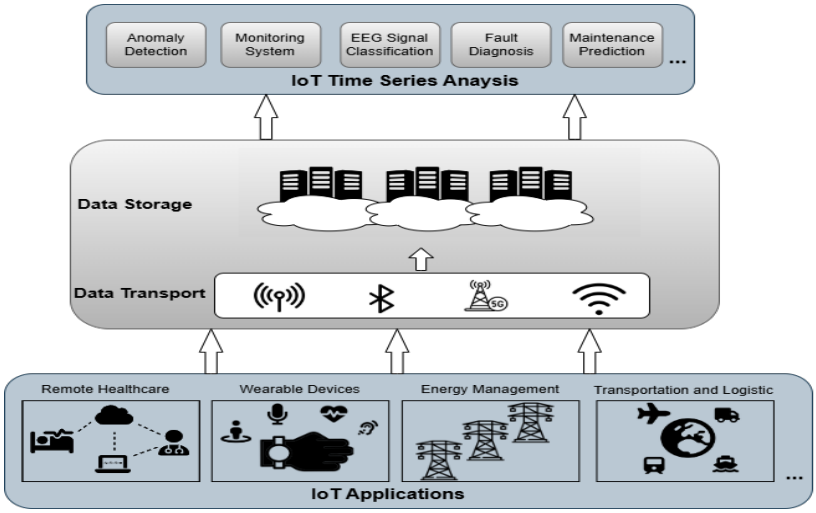


Figure 1. The general structure of time-series analysis in IoT

2.2. Image Transformation

Time-series image transformation converts time-series data into visual representations, such as images. It is a crucial process within the IoT context. This technique reduces IoT data dimensionality by compressing extensive data into a compact visual format, making it more successful at extracting key features and patterns from IoT time-series data. Additionally, it integrates seamlessly with deep learning algorithms like Convolutional Neural Network (CNN). These transformations enhance the analysis, interpretation, and utilization of time-series data in IoT applications. The transformation process of IoT time-series data into an image is illustrated in Figure 2.

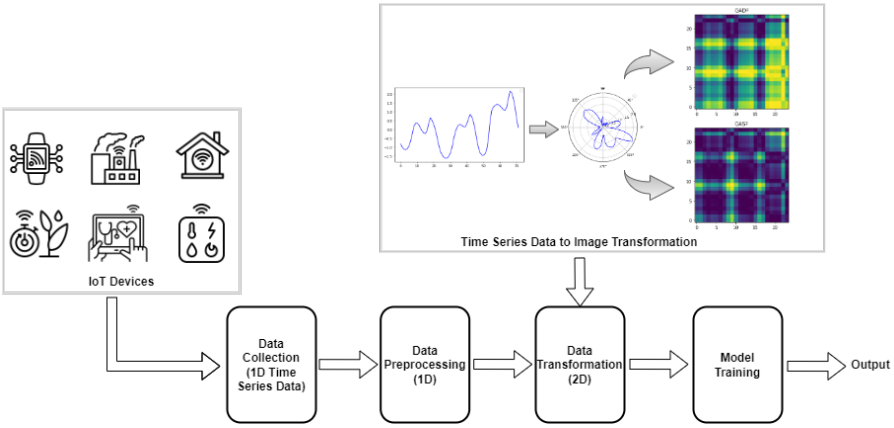


Figure 2. The overall framework of image transformation of IoT time-series data

There are varying image transformation techniques described in the literature. While these methods are directly applied to UTS, typically, they are not employed directly on MTS. To address this issue, some fusion methods are discussed in the literature. Image or feature fusion is a process that is proposed to merge the necessary information from images or features [29, 30]. When converting MTS data into two-dimensional (2D) images, fusion methods can be used to combine information from different variables or data sources to create a single image representation. One of the popular fusion techniques in literature is channel-based fusion, in which an RGB or multi-spectral channel image can be created by assigning each variable to a different color channel (e.g., red, green, blue) [31]. Also, some studies use tensor image fusion. MTS data is considered as a tensor and is analyzed by tensor decomposition techniques (e.g., Canonical Polyadic Decomposition) to extract patterns and interactions from the tensor data [32]. Lastly, feature level (early fusion) and decision level (late fusion) can be utilized to transform MTS [33]. Different variables are merged at the input stage and processed together with any methods at the feature level [34], [35]. On the other hand, each variable is converted into images separately, and then these images are combined at a later stage at the decision level [33], [36], [37]. Also, many researchers have used hybrid fusion by performing fusion in both decision and feature levels [38]. Table 2 summarizes the existing studies according to data types, such as univariate and multivariate. In addition, it emphasizes fusion techniques used in MTS data.

3. Time-Series to Image Transformation Techniques

There are several methods to transform one-dimensional (1D) time series into 2D images. Some of the popular techniques in literature are discussed below. Table 1 also shows the studies that used these methods.

3.1. Gramian Angular Field (GAF)

Given a time series is $X = \{x_1, x_2, \ldots, x_N\}$, including N samples, there are three steps to encode time series into images [76]. Firstly, X Time-series are scaled in the interval [0,1] according to Equation 1.

$$\tilde{x}_i = \frac{x_i - \min(X)}{\max(X) - \min(X)} \tag{1}$$

Then, the 1D time-series Cartesian coordinate system is transformed into a polar coordinate system, which is a new representation of the time series. Angular cosine (ϕ) and radius (r) are calculated to represent time series as polar coordinates using Equation 2.

$$\begin{cases} \phi &= \arccos(\tilde{x}_i), \quad -1 \le \tilde{x}_i \le 1, \quad \tilde{x}_i \in \tilde{X} \\ r &= \frac{t_i}{N}, \quad t_i \in \mathbb{N} \end{cases} \tag{2}$$

where t_i is the time stamp, and N is a constant factor in regularizing the span of the polar coordinate system. There are two types of GAF based on the sum/difference of the trigonometric function, which are the Gramian Angular Summation Field (GASF) and the Gramian Angular Difference Field (GADF). GASF is defined in Equations 3 and 4, and GADF is defined in Equations 5 and 6.

$$\mathrm{GASF} = \begin{bmatrix} \cos(\phi_1 + \phi_1) & \cdots & \cos(\phi_1 + \phi_n) \\ \cos(\phi_2 + \phi_1) & \cdots & \cos(\phi_2 + \phi_n) \\ \vdots & \ddots & \vdots \\ \cos(\phi_n + \phi_1) & \cdots & \cos(\phi_n + \phi_n) \end{bmatrix} \tag{3}$$

$$\mathrm{GASF} = \tilde{X}' \cdot \tilde{X} - \sqrt{I - \tilde{X}^2}' \cdot \sqrt{I - \tilde{X}^2} \tag{4}$$

$$\mathrm{GADF} = \begin{bmatrix} \sin(\phi_1 - \phi_1) & \cdots & \sin(\phi_1 - \phi_n) \\ \sin(\phi_2 - \phi_1) & \cdots & \sin(\phi_2 - \phi_n) \\ \vdots & \ddots & \vdots \\ \sin(\phi_n - \phi_1) & \cdots & \sin(\phi_n - \phi_n) \end{bmatrix} \tag{5}$$

$$\mathrm{GADF} = \sqrt{I - \tilde{X}^2}' \cdot \tilde{X} - \tilde{X}' \cdot \sqrt{I - \tilde{X}^2} \tag{6}$$

In the above equations, **I** refers to a unit row vector; $\tilde{X}'$ and $\sqrt{I - \tilde{X}^2}'$ is the transposed vector of the rescaled time series $\tilde{X}$ and $\sqrt{I - \tilde{X}^2}$, respectively.

Table 1. The Studies of Image Transformation Techniques for IoT

		Time-Series to Image Transformation Techniques						
Reference	**Year**	**GAF**	**MTF**	**RP**	**STFT**	**CWT**	**HHT**	**Others**
Baldini et al. [39]	2018			✓				
Yang et al. [40]	2019	✓	✓					
John et al. [41]	2019						✓	
Fahim et al. [42]	2020		✓					
Lyu et al. [43]	2020	✓						
Estabsari and Rajabi [44]	2020	✓	✓	✓				
Ferraro et al. [45]	2020	✓						
Xu et al. [46]	2020	✓						
Sreenivas et al. [47]	2021	✓	✓					
Zhu et al. [48]	2021	✓						
Anjana et al. [49]	2021				✓	✓	✓	
Zhou and Kan [32]	2021	✓						
Sharma et al. [50]	2021	✓						
Chen et al. [51]	2021				✓			
Jiang and Yen [52]	2021		✓					
Garcia et al. [53]	2021	✓	✓	✓	✓	✓		✓
Huang et al. [54]	2021	✓						
Jiang et al. [55]	2021	✓	✓					
Singh et al. [56]	2021				✓			
Santo et al. [57]	2022	✓	✓	✓		✓		
Chen and Wang [31]	2022	✓						
Bertalanič et al. [58]	2022	✓		✓				

Table 1. Continued: The Studies of Image Transformation Techniques for IoT

Alsalemi et al. [59]	2022	✓						
Zhang et al. [60]	2022			✓	✓			
Dou et al. [61]	2022					✓		
Abdel-Basse et al. [62]	2022	✓	✓	✓				
Wang et al. [63]	2022	✓	✓		✓			✓
Bai et al. [64]	2022	✓	✓					
Abidi et al. [65]	2023	✓	✓	✓				
Paula et al. [66]	2023	✓	✓	✓				
Quan et al. [67]	2023	✓	✓	✓				
Zhang et al. [68]	2023	✓						
Copiaco et al. [69]	2023							✓
Qu et al. [70]	2023	✓	✓					✓
Sun et al. [71]	2023	✓	✓					
Sayed et al. [72]	2023							✓
Hasan et al. [73]	2023	✓						
Hammoud et al. [74]	2023	✓	✓	✓				
Yan et al. [75]	2023	✓	✓		✓			

3.2. Markov Transition Fields (MTF)

MTF is a powerful tool that keeps time domain information in time-series data by representing the sequential Markov transition probabilities. By utilizing the Markov matrix of quantile bins, MTF offers an approach to converting the time-series data into images [76].

Given the time series $X = \{x_1, x_2, \ldots, x_n\}$, where x_i is the ith signal on the time-series. By determining Q quantile bins, each x_i is assigned to its corresponding bin $q_j (j \in [1, Q])$. In this way, a Markov transition matrix $\mathbf{W}$ in $Q \times Q$ Dimensions are obtained, which can be represented as:

$$W = \begin{bmatrix} w_{11} & w_{11} & \cdots & w_{1Q} \\ w_{21} & w_{22} & \cdots & w_{2Q} \\ \vdots & \vdots & \ddots & \vdots \\ w_{Q1} & w_{Q2} & \cdots & w_{QQ} \end{bmatrix} \tag{7}$$

$$w_{ij} = p\{x_t \in q_i | x_{t-1} \in q_j\} \tag{8}$$

where each element $w_{i,j}$ stands for the probability that a data point in the state q_j is followed by a data point in the state q_i. After normalization with $\sum_j w_{ij} = 1$, W becomes the Markov transition matrix. However, this matrix is insensitive to the distribution of X and temporal dependencies on time steps t_i. which results in the loss of excessive information in the process. To overcome this problem. W is expanded to a Markov transition field (MTF) matrix M by placing each probability in time order. It is expressed as below:

$$M = \begin{bmatrix} w_{ij|x_1 \in q_i, x_1 \in q_j} & \cdots & w_{ij|x_1 \in q_i, x_n \in q_j} \\ w_{ij|x_2 \in q_i, x_1 \in q_j} & \cdots & w_{ij|x_2 \in q_i, x_n \in q_j} \\ \vdots & \ddots & \vdots \\ w_{ij|x_n \in q_i, x_1 \in q_j} & \cdots & w_{ij|x_n \in q_i, x_n \in q_j} \end{bmatrix} \tag{9}$$

where M_{ij} in MTF represents the transition probability of data points in q_j followed by data points in q_j.

3.3. Recurrence Plot (RP)

RP is a widely used tool to visualize and analyze the recurrent behaviors of time series produced in a dynamic framework [77]. It is determined by a recursive matrix by computing the pairwise distance between the trajectories, in which the elements are calculated by Equation 10:

$$R_{i,j} = \Theta(\varepsilon - ||\vec{x}_i - \vec{x}_j||), i, j = 1, \dots, N \quad (10)$$

Where ϵ is a threshold, Θ is the Heaviside function used to binarize the distance matrices, where its value is zero for the negative argument and one for the positive argument.

RP exposes the local correlation information of a sequence and hidden patterns by computing the distance matrix between subsequences.

3.4. Short Time Fourier Transform (STFT)

STFT can be considered as the frequency domain representation of the original signal. It utilized a window function to extract a part of the time domain signal and then performed a Fourier transform on it to specify diverse signal properties [78]. The STFT of a given signal y(x) is calculated in Equation 11.

$$STFT(n, \omega) = \sum_{-\infty}^{\infty} y[x]\omega[n - x].e^{-j\omega n} \quad (11)$$

where $\omega(t)$ is the window function. In addition, the spectrogram is generated by squaring the STFT magnitude as follows:

$$Spectrogram(n, k) = |STFT(n, \omega)|^2 \quad (12)$$

Table 2. Summary of Image Transformation Application According to Data Types (U: Univariate, M: Multivariate)

Ref.	U	M	Year	Dataset	Gray/ Colored	Methods	Fusion Techniques
[68]	✓		2023	-Case Western Reserve University (CWRU) -Autonomous Experimental dataset	Color*	GASF GADF	-
[39]	✓		2018	- Private dataset of RF emissions collected from 11 IoT devices	Gray*	RP	-
[43]	✓		2020	- A private dataset that collected fiber intrusion disturbance signals	Color*	GAF	-
[48]	✓		2021	- KDD Cup 99 data	Color*	GAF	-
[58]	✓		2022	- Rutgers dataset	Gray Color	RP GAF	-
[42]	✓		2020	- REFIT electrical load measurement dataset	Color*	MTF	-
[44]	✓		2020	- Boston housing price data - Load Forecasting Dataset	Color*	RP GAF MTF	-
[41]	✓		2019	-Private dataset -Physionet/ Computing in Cardiology (CinC) Challenge 2016	Color*	HHT	-
[49]	✓		2021	- Seed	Color*	STFT CWT HHT	-
[51]	✓		2021	- Private dataset	Color*	STFT	-
[56]	✓		2021	- TUH Abnormal EEG Corpus	Color*	STFT	-
[60]	✓		2022	- Arrhythmia Data - Private dataset	Gray Color	RP STFT	-
[61]	✓		2022	- MIT-BIH arrhythmia - MIT-BIH normal sinus rhythm	Color*	CWT	-

				- BIDMC			
[66]	✓		2023	- Private dataset	Gray Color	GADF GASF MTF RP	

Table 2. Continued: Summary of Image Transformation Application According to Data Types (U: Univariate, M: Multivariate)

Ref.	**U**	**M**	**Year**	**Dataset**	**Gray/ Colored**	**Methods**	**Fusion Techniques**
[52]	✓		2021	- Private dataset	Color*	MTF	-
[53]	✓		2021	- Airbus SAS Airbus SAS 2018	Gray Color	GAF MTF RP GS STFT DWT	-
[64]	✓		2022	- Private dataset	Color	GAF MTF	-
[73]	✓		2023	- WSN Dataset - ETDataset - TON IOT Dataset	Color*	GAF	-
[47]	✓		2021	- MIT-BIH arrhythmia database	Color	GAF MTF	-
[59]	✓		2022	- UK-DALE dataset	Color	GAF	-
[54]	✓		2021	- Caltrans Performance Management System (PeMS)	Color*	GASF	-
[46]	✓		2020	- WISDM - UCI HAR - OPPORTUNITY	Color	GASF GADF	-
[74]	✓		2023	- Private dataset - OpenEDS - NaveGaze	Gray Color	GASF GADF MTF RP	-
[75]	✓		2023	- CICIDS2018 - IoT-23 - N-BaIoT - WSCF20231 - Private Dataset	Color	GASF GADF WT	-
[31]		✓	2022	- PLAID - WHITED	Color	GAF	- Single-channel images correspond to three channels in the RGB Color space, respectively, to create an RGB image.
[32]		✓	2021	- 2018 China Physiological Signal Challenge (CPSC2018) - PhysioNet Long-term ST dataset	Color*	GADF	- Each channel of the ECG signal transforms into a GAF image, which is represented as a 2nd-order ECG tensor. - These images are then stacked together to form a 3rd-order ECG tensor by concatenating them along the 3rd dimension.
[45]		✓	2020	- Backblaze SMART dataset	Color*	GAF	- Each feature of the time series is transformed into a polar coordinate through the GAF.
[69]		✓	2023	- The Simulated Energy Dataset (SiD) - The Dutch Residential Energy Dataset (DRED)	Gray Color	a grayscale image an RGB color image	- A 5 × 5 matrix is used to organize features for a given instant. - Then, the matrix is resized to 28x28 pixels and saved as a Grayscale or an RGB Color image.

						(jet colormap)	

Table 2. Continued: Summary of Image Transformation Application According to Data Types (U: Univariate, M: Multivariate)

Ref.	U	M	Year	Dataset	Gray/ Colored	Methods	Fusion Techniques
[70]		✓	2023	- PLAID - WHITED - HRAD	Color*	MTF GAF WVI	- Each variable is converted into images using three encoding techniques. - Then, the WVI image and MTF image are superimposed to create two channels. Also, the I-GAF image is saved as a new image by the Energy-Normalization (EN) block. - Lastly, this image is superimposed with the other two images to get a three-channel image.
[50]		✓	2021	- 1D Biomedical Signals such as ECG, PPG, temperature, and accelerometer	Color	GAF	-The average of the computed features from various channels is found and provided as a single fused feature set using Channel-Wise Mean Fusion (CAF).
[62]		✓	2022	- UCI HHAR - UCI MEHEALTH	Color	RP MTF GAF	- Each row includes three measures, x, y, and z, for AM, GY, and MG data in 3D, respectively. - Converts x-, y-, and z-axis of signals as red, green, and blue channels of images.
[57]		✓	2022	- PAKDD2020 Alibaba AI OPS Competition - NASA bearings	Color*	RP GAF MTF STFT	- MTS encoded a set of feature maps that were computed with four different image transformation techniques.
[71]		✓	2023	- Private dataset	Color	GAF MTF	- GASF, GADF, and MTF layers are placed on the red, green, and blue layers, respectively, and saved images.
[65]		✓	2023	- SITS data, which was collected for a different study of the Dordogne-Reunion Island study - Koumbia	Color*	GADF GASF MTF RP	- Each UTS in MTS is flattened to the direct use of MTS instead of thinking independently of each UTS. - Then, generate the 2D images from the flattened MTS.
[72]		✓	2023	- Student Room Dataset (SRD) - UCI dataset (an office space) - Living Room Dataset (LRD)	Gray Color	data normalization matrix conversion	- The list of features(n) of the dataset is arranged into a 3xn matrix format. - Then, the matrix is resized to 28x28 pixels and saved as an image.
[83]		✓	2021	- Wafer dataset - ECG dataset	Color	GASF GADF MTF	- Encode MTS as a Colored image for each univariate time series. Each Colored image is separated into three monochromatic images, namely red, green, and blue (RGB). - After the separation, these monochrome images are

							concatenated together to form a huge image.

Table 2. Continued: Summary of Image Transformation Application According to Data Types (U: Univariate, M: Multivariate)

[63]	✓	✓	2022	- Case Western Reserve University (CWRU) Dataset - Society for Machinery Failure Prevention Technology (MFPT) Dataset	Color*	STFT The direct drawing method GADF MTF	- The vibration signals from multiple channels are combined into a 2D spectrum map.
[67]	✓	✓	2023	- Chinatown +84 UCR datasets	Color*	GAF MTF RP GMR	- 1D multi-scale features and 2D image features are fused in two distinct methods, covering the feature fusion methods such as SE and SA -Three images which are encoded with different coding methods are overlapped as three-channel data inputs.
[55]	✓	✓	2021	- 24 benchmark datasets (14 datasets for MTS and 10 dataset for UTS)	Color*	GAF MTF	- Each UTS in MTS is converted into GM images. Each variable is considered a channel. - G-image and M-image are concatenated GM feature maps by the adaptive feature aggregation, which pass through a corresponding shallow CNN separately.

* The color is not specified in the paper. For this reason, the color is determined based on the given images.

3.5. Continuous Wavelet Transformation (CWT)

CWT offers an unstable window size that adjusts based on the frequency at the cost of time resolution. Although STFT provides a great representation of the signal's time-frequency characteristics, it presents a fixed resolution in the frequency domain, which is not always ideal in certain scenarios. On the other hand, CWT is an operation linear on a time-domain signal y(t) given by:

$$W_{a,b}[y(t)] = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} y(t) * \phi\left(\frac{t-b}{a}\right) dt \quad (13)$$

where $\phi(\frac{t-b}{a})$ is a dilated version of the base wavelet function $\phi(t)$ by applying scaling and shifting. $a > 0$ is the scaling variable that regulates the spread of the function, and b is the time-shifting parameter or the instant of time at which the signal needs to be analyzed. The visual representation of the CWT of a signal is referred to as a scalogram [79].

3.6. Hilbert Huang Transform (HHT)

HHT is an analysis of signals that are non-stationary and non-linear [80]. While most techniques may fail in analyzing nonstationary and nonlinear systems, HHT alleviates the challenges of time-frequency-energy representation of the data. HHT includes two primary phases, called Empirical Mode Decomposition (EMD) and Hilbert Transform (HT). The transformation involves several processes. First, EMD is utilized to obtain Intrinsic Mode Functions (IMFs) from the signals. Second, the Hilbert transform is applied to each of the IMF components. Finally, the instant frequency and amplitude can be computed.

3.7. Other Transformation Methods

In addition to the aforementioned methods, the literature offers a range of alternative techniques that are commonly employed to address various types of problems. These methods play a pivotal role in the transformation of IoT time-series data. Some

of the notable approaches in this regard include data normalization combined with matrix conversion, the direct drawing method, Gaussian Mixture Regression (GMR), Gray-Scale encoding (GS), Gray-Scale image representations, RGB color image conversion, and the Wavelet Variance Image (WVI) method. These techniques have gained popularity within the literature for their effectiveness in transforming and enhancing the analysis of IoT time-series data.

Garcia et al. [53] proposed a modification of GS by choosing lower and upper bounds in the original formulations in accordance with the GAF encoding instead of minimum and maximum scaling. Wang et al. [63] used the direct drawing method which signals are transformed into a 2D spectrum map directly with plt functions in the Matplotlib package in Python without any processing. The direct drawing method has higher accuracy than GAF and MTF after STFT. The main idea of GS is to transform time-domain raw signals into images. The time-domain raw signals complete the pixels of the image sequentially. Wen et al. [81] reorganized the GS using CNN for fault diagnosis in manufacturing systems. A transformation method consisting of data normalization and matrix conversion was used for 2D image representation [72], [69]. 1D time-series data is first normalized in [0,1] with n features. Then, these features are arranged in *mxm* matrix format. Lastly, this matrix is resized to 28x28 pixels and saved as an image to obtain a gray-scale image or RGB color image. The Voltage–Current (VI) trajectory can be converted into a pixelated VI image (*nxn* matrix) by meshing the VI trajectory [82]. Qu et al. [70] generated 2D load signatures according to the corresponding features of the signal based on the Weighted Voltage–Current (WVI) trajectory image.

4. Image Transformation in IoT Applications

IoT encompasses various domains where time-series data is frequently used. Time-series data is a data type that includes a sequence of data points that are collected at regular intervals over time. Table 3 summarizes the existing studies by categorizing them according to nine IoT domains. Here are some IoT domains where time-series data is commonly utilized.

4.1. Security and Privacy

The security and privacy domain within IoT focuses on handling the challenges and risks associated with ensuring the confidentiality, integrity, availability, and privacy of IoT systems, devices, and data. Security in the IoT domain includes implementation of preventive measures to obstruct unauthorized access, data breaches, and malicious activities that have the potential to jeopardize the functionality, integrity, and confidentiality of IoT devices and systems. Privacy in the IoT domain refers to the protection of individual's personal information and their control over how it is collected, used, and shared by IoT systems.

IoT time-series data plays a significant role in the security and privacy domain by providing valuable insights into the behavior, patterns, and anomalies within IoT systems. Anonymization, encryption, and access controls should be applied appropriately to protect sensitive information contained within the time-series data. In the context of IoT security and privacy, time-series data can be leveraged for various purposes: Intrusion detection, unauthorized access detection, anomaly detection, security analytics prediction, etc.

Baldini et al. [39] presented an approach for the authentication of IoT wireless devices based on Radio Frequency (RF) emissions. The proposed approach, which combines CNN and RP (RP-CNN), is tested on the RF emissions dataset, which is experimental data collected from 11 IoT devices. They also applied two classification methods called T-CNN, which utilizes the digital representation of the RF emissions directly with CNN, and FEAT, which extracts the statistical characteristics of RF emissions from their digital representations. The results showed that the RP-CNN improves accuracy when compared to T-CNN and FEAT. Lyu et al. [43] proposed an intrusion pattern recognition framework. The method, based on the GAF and CNN, achieved a high-speed response time of 0.58 s and a high recognition accuracy of 97.57% for six types of optical fiber intrusion events. In addition, it improved the robustness and practicability of the system because the GAF algorithm is not sensitive to the fluctuation of power sources in the optical path. Zhu et al. [48] developed a monitoring system to detect abnormal traffic and vulnerability attacks in IoT applications. In the system, time series data was converted into GAF graphs, and the CNN and Long Short-term Memory (LSTM) combination model was utilized to monitor traffic. However, the system that combined C5.0 decision tree (DT) and time series analysis introduced a novel idea for the traffic analysis of IoT devices. Bertalanič et al. [58] proposed a new resource-aware approach based on image transformation and deep learning for anomaly detection in the wireless link layer. Time-series data were transformed into images using RP and GAF. The experiments show that RP outperforms the GAF methods by up to 14%. Yan et al. [75] developed an intrusion detection model for few-shot attacks. 1D network traffic data was converted into three-channel RGB images using GADF, GASF, and WT. The three two-dimensional images are fused into the red, green, and blue channels of one RGB image, respectively. Additionally, the data augmentation module uses an improved Denoising Diffusion Probabilistic Model (DDPM), and the image classification module employs a variable network ETNet V2 based on EfficientNetV2. The results indicate that the proposed improved GAF method, combined with WT, achieves the highest accuracy compared to one-dimensional data and other types of conversion methods such as GAF, MTF, and STFT.

4.2. Energy Management

IoT enables the monitoring and control of energy consumption, smart grid management, and the integration of renewable energy sources. It helps optimize energy distribution, reduce waste, and improve sustainability.

Fahim et al. [42] proposed a model called Time-series to Image (TSI) to detect abnormal energy consumption in residential buildings. This study focused on analyzing the univariate time-series energy data for very short-term analysis. The Proposed model utilized a One-Class Support Vector Machine (OCSVM) as a classifier and MTF as a converter, which transforms univariate time-series data into images. In this work, the authors demonstrated that this image representation further enhances the classifier's ability to detect anomalous behavior more efficiently. Estebsari and Rajabi [44] proposed a hybrid model based on CNN and image encoding methods for single residential loads. They applied three different image encoding methods, including the RP, GAF, and MTF, to historical load time-series data. The experiments revealed that RP performed the best among the three encoding methods. Alsalemi et al. [59] developed a novel GAF classifier based on the EfficientNet-B0 for the classification of edge internet of energy applications. The authors aimed to introduce the first lightweight classifier for 2D energy consumption working on the ODROID-XU4 platform.

Copiaco et al. [69] proposed a 2D pre-trained CNN model for detecting anomalies in building energy consumption. This model used the 2D versions of the energy time-series signals to give input to several pre-trained models, such as AlexNet and GoogleNet, as features of the Linear Support Vector Machine (SVM) classifier. In this study, 1D time series were transformed into Grayscale and Jet Color image representations. This study showed that converting energy time-series data into images can provide an increase in the correlation between images with the same class. Chen and Wang [31] proposed an edge-computing architecture for load recognition tasks in the field of Non-Intrusive Load Monitoring (NILM) that reduces data transmission volume and network bandwidth requirements. They also developed a color encoding method based on GAF to construct load signatures in home appliances. Qu et al. [70] constructed three 2D load signatures based on the WVI, MTF, and current spectral sequence-based GAF (I-GAF). Additionally, they designed a new Residual Convolutional Neural Network with Squeeze-and-Excitation (SE) and Energy-Normalization (EN) blocks (EN SE-RECNN) for appliance identification in NILM. This study compared the performance of various models, including Residual Convolutional Neural Network (RECNN), Residual Convolutional Neural Network with EN blocks (EN-RECNN), and EN-SE-RECNN, and confirmed that the performance of EN-SE-RECNN was better. Also, their findings demonstrate that the fusion of different signatures enhances performance by enriching the information related to appliance identification.

4.3. Healthcare

In healthcare applications, time-series data assists in monitoring patient vital signs, analyzing health trends, predicting disease outbreaks, and optimizing healthcare resource allocation. Zhou and Kan [32] developed a tensor-based framework for ECG anomaly detection in Internet of Health Things (IoHT)-based cardiac monitoring and smart management of cardiac health. The multi-channel ECG signals were converted into 2D images using GADF. In this study, a tensor decomposition-unsupervised anomaly detection model has been proposed, utilizing multi-linear principal component analysis (MPCA) and deep support vector data description (deep SVDD). The proposed model demonstrated that the framework using 2D image representations shows better performance than one that directly uses 1D signals because of the difficulty of extracting hidden information. Also, when the effect of the ECG length on the GADF image was examined, larger GADF images were found to give higher accuracy, as well as the area under the ROC curve (AUROC) and F-score.

Sreenivas et al. [47] proposed a CNN model for the classification of arrhythmia in dual-channel ECG signals. In this study, GAF and MTF were used to convert the ECG time-series signals into images. The result showed that the GAF model achieved higher accuracy compared to the MTF. Anjana et al. [49] proposed a CNN model based on various types of image encoding approaches to classify human emotions using EEG signals. In this study, Spectrogram, Scalogram, and HHT were employed to transform EEG signal data into images. The experiments showed that the scalogram of image encoding provides the best classification accuracy. Paula et al. [66] proposed a 2D-kernel-based CNN architecture to classify the Steady State Visually Evoked Potentials (SSVEP) signal. In this work, EEG data is encoded into images using GADF, GASF, MTF, and RP. This study demonstrated that the GADF and RP methods consistently showed higher performance. Also, the 1D-kernel-based structure of the model was insufficient for learning the necessary information from the data.

John et al. [41] developed a cardiac monitoring system based on wireless sensing, aiming for accurate diagnosis of heart diseases. The system used MQTT for long-distance transmission and HHT for preprocessing and feature extraction of the data. Sharma et al. [50] introduced a patient monitoring system based on ontology for early remote detection of COVID-19. The proposed system relied on an alarm-enabled bio-wearable sensor system that utilized sensory 1D biomedical signals such as ECG, PPG, temperature, and accelerometer. These 1D Biomedical signals were converted into images with GASF after extracting their features. Then, SVM and K-Nearest Neighbors (KNN) were employed as ML-based classifiers for the classification of COVID-19 patients. Chen et al. [51] proposed an indoor speed estimation framework, SpeedNet, from radio signals, mainly aimed at monitoring the movement of elderly individuals. The SpeedNet framework includes three modules: the dominant path extraction module, the spectrum analysis module, and the deep learning module. The dominant path signal which is obtained from the extraction module was analyzed using STFT in the spectrum analysis module. Also, CNN and LSTM were utilized in the deep learning module to extract spatial and temporal features. They introduced a new approach for contactless indoor speed estimation with radio signals, addressing the challenges posed by the complex relationship between the speed of moving individuals and radio signals. Singh et al. [56] suggested a brain signal classification model that transformed brain signals into images as input for a pre-trained VGG19 model by using STFT for seizure detection. In addition, blockchain technology was utilized to store images more securely. The study also emphasizes the importance of

selecting an appropriate encoding method, which involves using different image conversion techniques such as spectrograms, chronograms, or kurtograms. Zhang et al. [60] proposed a system based on 5G-enabled Medical IoT for automatic detection of arrhythmia (ARR). Time-frequency spectrograms obtained from RR interval sequences using RP and Fourier Transform (FT) were used as inputs to a unified CNN and LSTM model for the classification of ECG signals. Dou et al. [61] proposed a novel classification method based on CWT and CNN within the context of the IoT domain. Their approach simultaneously classifies various ECG signals for heart disease diagnosis using GoogleNet. Besides, ECG signals were converted into time-frequency images with CWT. Abdel-Basset et al. [62] developed a lightweight Human Activity Recognition (HAR) architecture designed to classify human activities captured by heterogeneous sensors from different IoT devices. They proposed a few modifications for three encoding techniques, including RP, MTF, and GAF. These techniques encode the three-dimensional (3D) time-series data of human activities into three-channel images to overcome the heterogeneity in sensory data.

Hammoud et al. [74] proposed a DL framework for the classification of Parkinson's disease (PD) and Progressive Supranuclear Palsy (PSP). They extracted the pupil features such as coordinates, area, minor axis, and major axis. The time-series signals represented by the pupil's coordinates and its area were reprocessed. Then, these features were converted into images using GASF, GADF, MTF, RP, RGB-GAF, and GAF-MTF. The results demonstrate that GADF, RP, and RGB-GAF achieved higher accuracy than other methods.

4.4. Industrial

Industrial IoT (IIoT) involves connecting industrial equipment, machinery, and systems to enable data monitoring, analysis, and optimization in manufacturing, energy transportation, and other industrial sectors. In industrial settings, time-series data helps monitor equipment performance, predict failures, optimize maintenance schedules, and improve overall operational efficiency.

Various image encoding methods are commonly used in IIoT to provide intelligent and efficient fault diagnosis. Wang et al. [63] proposed a framework for fault diagnosis of single-channel and multi-channel bearing signals. They combined spectrum map information fusion and CNN to achieve fast fault diagnosis. GADF, MTF, and STFT were used to generate a 2D spectrum graph from 1D bearing vibration data, and STFT achieved the best result with the lowest loss value. The experiments indicated that the STFT method could use multichannel information effectively and improve fault identification accuracy. Similarly, Zhang et al. [68] presented a novel fault diagnosis method that combines GAF, Extreme Learning Machine (ELM), and CNN. They explored different encoding methods, including GADF, GASF, spectrogram, and gray-scale image, to indicate the effectiveness of the chosen encoding techniques for pattern recognition. The findings indicated that the GADF has the highest performance. Santo et al. [57] developed a model that combined time-series encoding techniques and CNN for predictive maintenance. This paper evaluated four main encoding techniques, RP, GAF, MTF, and Wavelet transform. The RP achieved the best performance in all metrics.

Ferraro et al. [45] developed an efficient method for predictive maintenance that improved maintenance strategies and decreased downtime and cost. The method involves transforming temporal time-series data into images using GAF and utilizing deep learning strategies to predict the health status of the Hard Disk Drive (HDD). Jiang et al. [52] proposed the MTF-CLSTM method, which combines the MTF, CNN, and LSTM to predict product quality in Wire Electrical Discharge Machining (WEDM). MTF is employed to transform dynamic WEDM manufacturing conditions into images. In addition, features were extracted from the images with CNN, and LSTM was used to predict the surface roughness of the WEDM products right after manufacturing. When the MTF-CLSTM method was compared with the Deep Neural Network (DNN) and the Markov Chain DNN(MC-DNN) methods [84], the proposed method achieved the best performance.

Garcia et al. [53] explored six encoding methods (GAF, MTF, RP, GS, spectrogram, and scalogram) and the modifications to enhance their robustness against the variability in large datasets when transforming temporal signals into images. This study revealed that different encoding methods exhibit competitive results for anomaly detection in large datasets. Bai et al. [64] proposed a fault diagnosis method called Time-series Conversion-DCGAN (TSC-DCGAN). They utilized GAF and MTF to transform 1D electrical parameters into 2D images. Additionally, the Deep Convolutional Generative Adversarial Network (DCGAN) was used as a generation method to handle the inadequate data samples of electrical parameters from oil wells. Also, the experimental results show that GAF images performed better in terms of classification effectiveness compared to MTF images. Sun et al. [71] put forward an idea for diagnosing composite failures of the adaptable multi-sensor bearing gear system by leveraging GAF, MTF, and ResNet. The complicated multi-dimensional time-series signals were fused and transformed into 2D images to facilitate classification tasks using GAF and MTF.

4.5. Environmental Monitoring

IoT devices are exploited to monitor and manage environmental conditions such as air quality, water quality, pollution levels, and natural resource conservation. These solutions aid in environmental protection and sustainable practices.

Abidi et al. [65] proposed a framework for the classification of Land Use/Land Cover (LULC) mapping based on 2D encoded multivariate Satellite Image Time-series (SITS). In this work, multivariate SITS data were converted into 2D images by

GADF, GASF, MTF, and RP. The results indicated that the RP technique performed better than all encoding techniques. In addition, the combination of 2D encoding techniques achieved better performance than using the encoding methods alone.

4.6. Smart Building

Smart Building enhances occupant comfort, reduces energy consumption, improves safety and security, and optimizes building operations and maintenance. Time-series data in smart buildings is employed to monitor and control various building systems, such as HVAC (Heating, Ventilation, and Air Conditioning), lighting, and occupancy.

Sayed et al. [72] presented an approach for the detection of occupancy using environmental sensor data such as temperature, humidity, and light sensors. In this study, multivariate time-series data were transformed into gray-scale and RGB images using an image transformation method to encode better and obtain relevant features. This method covered data normalization and matrix conversion, unlike commonly used methods such as GAF. The results showed that gray-scale images provide the appropriate balance between accuracy and training time compared to the colored images.

4.7. Transportation and Logistics

IoT applications in transportation and logistics include fleet management, vehicle tracking, route optimization, cargo monitoring, and driver safety. These applications have the potential to transform the industry by enabling intelligent decision-making, reducing costs, and improving customer experience.

Huang et al. [54] developed a new method, namely the Traffic Sensor Data Imputation GAN (TSDI GAN), for missing data reconstruction. GASF was employed in the paper to process time-series traffic data and transform it into an image format for missing value imputation using CNN.

4.8. Wearable Devices

Wearable devices focus on the integration of technology into portable devices that individuals can wear. These devices are equipped with sensors, connectivity capabilities, and computing power, enabling them to collect data, interact with the environment, and provide personalized experiences.

Wearable devices incorporate various sensors to collect data about the user and their environment, such as accelerometers, heart rate monitors, GPS, temperature sensors, etc. They are also connected to other devices or networks through wireless technologies such as Bluetooth and Wi-Fi. Thus, wearable devices offer individuals convenient access to personalized data and experiences, empowering them to monitor their health, improve their fitness, and stay connected in a more seamless and unobtrusive manner.

With the advancement of the IoT and wearable devices, sensor-based HAR has gained importance due to convenience and privacy characteristics. Xu et al. [46] presented two improvements based on GAF and deep CNN based on the Multi-dilated Kernel Residual (Mdk-Res) module for HAR. The findings indicated that the developed model was able to efficiently extract multi-scale features and improve the accuracy of activity recognition by utilizing the GAF algorithm's characteristics, along with the structure and advantages of CNN, residual learning, and dilated convolution.

Table 3: Summary of Image Transformation Techniques Studies in IoT Application Domain (A: Authentication, C: Classification, D: Detection, I: Imputation, P: Prediction, R: Recognition

Domain	Ref.	Year	Problem Type	Application Type	Methods	Models	Comparison Models	Results
Security	[39]	2018	A	Authentication of IoT devices	RP	CNN	T-CNN FEAT	*Accuracy:* RP-CNN: 96.8% T-CNN: 96.2% FEAT: 91.3%
	[43]	2020	C	Intrusion Pattern Recognition	GAF	CNN	VGG16 ResNet50 Inception V3	*Accuracy:* 97.67%
	[48]	2021	C	Anomaly Detection	GAF	C5.0 DT CNN-LSTM	-	*Accuracy:* 96%
	[58]	2022	C	Anomaly Detection	RP GAF	CNN	KNN SVM AlexNet VGG11	*F1-score:* SuddenD: 1.00 SuddenR: 1.00 InstaD: 0.92 SlowD: 0.99 No anomaly: 0.99

	[75]	2023	C	Intrusion Detection	GADF GASF WT	DDPM ETNet V2	CNN SVM	*Accuracy:* 99.20%

Table 3. Continued: Summary of Image Transformation Techniques Studies in IoT Application Domain (A: Authentication, C: Classification, D: Detection, I: Imputation, P: Prediction, R: Recognition

Energy Management	[42]	2020	D	Anomaly Detection	MTF	OCSVM	PCA+OCSVM	*F1-score:* 88%
	[44]	2020	P	Single Residential Load Forecasting	RP GAF MTF	CNN	SVM ANN 1D-CNN	*MAE:* 0.59 *MAPE:* 12.54 *RMSE:* 0.79
	[59]	2022	C	Energy Consumption Data Classification	GAF	EfficientNet-B0	-	-
	[31]	2022	R	Load Recognition	GAF	ResNet	Other Ref. Papers (LSTM, CNN and three AlexNet versions)	*Accuracy:* PLAID:97.97% WHITED:97.90%
	[69]	2023	D	Anomaly Detection	Grayscale image RGB color image (jet colormap)	AlexNet GoogleNet SqueezeNet Linear SVM	AlexNet GoogleNet	*F1-scores:* SiD: 93.63% DRED: 99.89% *Accuracy:* SiD: 96.11% DRED: 99.91%
	[70]	2023	R	Load Recognition	MTF GAF WVI	EN-SE-RECNN	RECNN EN-RECNN EN-SE-RECNN	*Accuracy:* PLAID:97.43% WHITED:95.99% HRAD:98.14%
Healthcare	[41]	2019	C	Cardiac Monitoring System	HHT	Adaptive Threshold Method	-	*Accuracy:* 96%
	[47]	2021	C	Arrhythmia Classification	GAF MTF	CNN	Other Papers	*Accuracy:* GAF: 97% MTF: 85%
	[51]	2021	P	Indoor Speed Estimation	STFT	CNN-LSTM	Other Papers	*Accuracy*: 96.33%
	[32]	2021	D	Anomaly Detection	GADF	Deep SVDD Statistical Control Charts MPCA	Adaboost SVM	*F1-score:* Atrial fibrillation: 0.9771 Right bundle branch block: 0.9986 ST-depression: 0.9550
	[50]	2021	D	Remote Patient Monitoring (RPM)	GAF	SVM KNN	SVM and KNN with different Fusion Methods	*Accuracy:* 96.33%
	[49]	2021	C	Emotion Classification	STFT CWT HHT	CNN	-	*Accuracy:* Scalogram: 98%, Spectrogram: 78%, HHT: 75%
	[56]	2021	C	Brain Signal Classification	STFT	VGG-16	SVM Logistic Regression Random Forest	*Accuracy:* 88.04%
	[60]	2022	C	ECG Signal Classification	RP FT	CNN-LSTM	Other Papers	*Accuracy:* 99.06%, *Sensitivity:* 98.29%, *Specificity:* 99.73%

	[62]	2022	C	HAR	RP MTF GAF	CNN-based model	Other Papers	*Accuracy:* HHAR: 98.90%, MEHEALTH: 99.68%

Table 3. Continued: Summary of Image Transformation Techniques Studies in IoT Application Domain (A: Authentication, C: Classification, D: Detection, I: Imputation, P: Prediction, R: Recognition

Domain	**Ref.**	**Year**	**Problem Type**	**Application Type**	**Methods**	**Models**	**Comparison Models**	**Results**
Healthcare	[61]	2022	C	ECG Signal Classification	CWT	GoogLeNet	AlexNet VGGNet	*Accuracy:* 94.28%
	[66]	2023	C	EEG Signal Classification	GADF GASF MTF RP	ImageNet DenseNet ResNet Google Net AlexNet	1D-kernel-based CNNs	*Accuracy:*(ResNet50) RP –> 96% GADF –> 94% MTF –> 88% GASF –> 54%
	[74]	2023	C	Neurological Diseases Diagnosis	GADF GASF MTF RP	CNN	Other Papers	*Accuracy:* Left eye: 96.9% Right eye: 90.8% Both eyes: 96.9%
Industrial	[45]	2020	P	Maintenance Prediction	GAF	CNN	LSTM	Accuracy: 97.70%
	[52]	2021	P	Product Quality Prediction	MTF	CNNLSTM	Other Papers DNN MC-DNN	*MAPE:* 3-state MTF: 3.11% 4-state MTF: 2.94% 5-state MTF: 3.24%
	[53]	2021	D	Anomaly Detection	GAF MTF RP GS STFT DWT	CNN	-	*AUC:* SC: 92 GS: 89 MTF Mod: 87 GAF Mod: 84
	[57]	2022	P	Maintenance Prediction	RP GAF MTF CWT	CNN	LSTM GRU XGBoost ResNet-50 DenseNet-121 VGG-16	*F1-score:* GAN: 34.47 CNN: 59.24 *Accuracy:* RP: 0.95
	[63]	2022	D	Bearing Fault Detection	STFT The direct drawing method GADF MTF	VGG	-	*Accuracy:* MFPT: STFT: 99.8% CWRU: DDM: 93.8% GADF: 78.1% MTF: 79.7% STFT: 100%
	[64]	2022	D	Fault Diagnosis	GAF MTF	DCGAN EfficientNet	CNN VGG16 GoogleNet	*Accuracy:* 0.8541
	[71]	2023	D	Fault Diagnosis	GAF MTF	ResNet	DCNN	*Accuracy:* 72.14%

	[68]	2023	D	Fault Diagnosis	GASF GADF	ELM CNN	AlexNet ANN SVM KNN	*Accuracy:* 99.2%

Table 3. Continued: Summary of Image Transformation Techniques Studies in IoT Application Domain (A: Authentication, C: Classification, D: Detection, I: Imputation, P: Prediction, R: Recognition

Domain	**Ref.**	**Year**	**Problem Type**	**Application Type**	**Methods**	**Models**	**Comparison Models**	**Results**
Environmental Monitoring	[65]	2023	C	Time-series Classification	GADF GASF MTF RP	CNN ResNet-50	Other Papers	*F1-scores:* Reunion Island:89.34% Dordogne:90.26 % Koumbia study: 78.94%
Smart Building	[72]	2023	P	Building Occupancy Prediction	Data Normalization Matrix Conversion	CNN	KNN DT RF	*Accuracy:* SRD: 99.11% LRD: 98.54% UCI: 99.42%
Transportation and Logistics	[54]	2021	I	Traffic Data Imputation	GASF	DCGAN	Other Papers	*MAE:* 13.7%
Wearable Devices	[46]	2020	C	HAR	GASF GADF	Mdk-Res Module ResNet	Multilayer Perceptron (MLP) LSTM CNN_1D CNN_2D ResNet GoogleNet	*Accuracy:* Proposed: 96.83% CNN: 93.23 LSTM: 87.53
Others *	[40]	2019	C	Time-series Classification	GASF GADF MTF	ConvNet	ConvNet VGG16 Other Papers	*Error rates:* MTF: 0.4 (Wafer) GADF: 5.35 (ECG)
	[55]	2021	C	Time-series Classification	GAF MTF	ADDN	ResNet Encoder MLP MCDCNN Time-CNN	*MPCE:* UTS: 2.90 MTS: 4.00
	[67]	2023	C	Time-series Classification	GAF MTF RP GMR	ResNet	ResNet Dynamic Time Warping (DTW) MLP Fully Convolutional Network (FCN)	*Error rates:* GMR: 0.2305 GAF: 0.2431 MTF: 0.2863 RP: 0.2543
	[73]	2023	D	Sensor Fault Diagnosis	GAF	ResNet18-SVM-GAN	ResNet18-SVM	*Accuracy:* 98.7%

* The studies have not been provided with any domain-specific information

4.9. Others

Beyond the above-mentioned IoT domains, various studies employ data from different fields within IoT. In these studies, the effects of the proposed methods on the datasets obtained from diverse fields were examined. For example, Yang et al. [40] used two well-known MTS datasets, Wafer and ECG, to classify 1D signals. MTS data was transformed into 2D images by applying GASF, GADF, and MTF. These images were then concatenated as RGB input channels for the ConvNet classification model. The study concluded that the choice of encoding methods had no significant impact on the prediction results. Jiang et al. [55] evaluated the Adaptive Dila-DenseNet (ADDN) model for classifying UTS and MTS data across 24 benchmark IoT datasets. Both UTS and MTS data were converted into GM-images by leveraging GAF and MTF methods to feed into the ADDN model. Quan et al. [67] investigated the impact of different feature construction and fusion methods on time-series classification results. They proposed an improved Multi-Scale ResNet (MSResNet) for time-series classification. In this study, three images encoded with different methods, including GAF, MTF, and RP, were superimposed as three-channel data inputs as GMR images. Besides, 1D multi-scale features and 2D image features were fused using two distinct methods, including Squeeze-and-Excitation (SE) and Self-Attention (SA) feature fusion architectures. Hasan et al. [73] introduced a sensor fault detection approach based on digital twins. They used the GAN method to create the digital representation of the sensor. Also, the GAN was trained with images obtained by converting time-series using GAF.

5. Research Challenges and Future Directions

Converting time-series data into images attracted significant attention in facilitating IoT data analysis. While this transformation can offer new perspectives and enable advanced image processing techniques, it also introduces several challenges and limitations related to data handling, computational requirements, interpretability, and real-time processing. Since IoT devices typically generate large volumes of data, converting this data into images can result in significant storage and processing expenses. Additionally, time-series data often contains noise and anomalies, which can negatively impact the transformation process and the quality of the resulting images. To address this issue, significant preprocessing may be required to clean and normalize the data before transformation, adding to the complexity and time required for analysis. Also, the transformation process is not simple and requires complex techniques. The choice of the conversion method can significantly affect the result and the quality of the resulting images. However, these images may not be easily interpreted by users who are not familiar with certain transformation techniques. Lastly, some machine learning and image processing algorithms may not be appropriate for analyzing images derived from time-series data, as they cannot adequately describe the underlying patterns or relationships.

As mentioned above, this transformation process has a set of challenges. The major challenges and potential solutions are presented to address them as follows for researchers [44, 85–96].

• IoT time-series data is often prone to noise and missing values caused by sensor failures or network problems, which can adversely affect the image quality. Also, missing data when creating an image can lead to incomplete representations. To address this challenge, researchers should investigate advanced imputation techniques to handle missing gaps. This could involve developing specialized image inpainting models [85] specifically designed for images derived from time-series data or implementing GAN-based models [86] that can be utilized to learn the unique temporal patterns associated with various IoT domains.

• Encoding large-scale IoT time-series data can be computationally expensive and memory-intensive. To overcome this limitation, future research should develop dynamic resolution techniques that automatically determine optimal image dimensions based on data complexity [88]. On the other hand, it should be noted that very small sizes can lead to the loss of essential details while reducing memory and computational costs. Moreover, current techniques can be redesigned to accelerate image conversion.

• The process of transforming time-series data into images involves compressing the temporal information into a 2D representation. This compression can cause information loss. Balancing the trade-off between dimensional reduction and information loss is a critical challenge in this field [89]. To minimize information loss, researchers should focus on developing robust transformation techniques that balance dimensionality reduction and information preservation. Besides, in order to avoid information loss, modifications can be made to the transformation methods, such as changing the function in a formula [90], [91].

• IoT time-series data can involve multiple variables or sensors, resulting in MTS. However, the methods described cannot be applied directly to MTS. To better handle multivariate data, researchers should focus on various approaches. They can develop specialized encoding techniques that can directly represent relationships between variables, adapting MTS data in the visual domain. In addition, dimension reduction methods can be utilized to implement encoding techniques directly [92, 93] that preserve variable correlations while reducing complexity. Also, effective fusion techniques where each channel captures different aspects of the multivariate relationship can be developed to transform MTS data [94].

• Real-time or near-real-time image representations for dynamic IoT environments can be challenging [44]. If the image transformation process takes longer than the time between intervals, the system can fail eventually.

This delay may be unacceptable for decision-making systems within a short period in IoT applications such as IIoT and smart

cities. To improve processing speed, future research should increase hardware resources that can help reduce computation time. Also, edge devices provide prominent computational resources for faster real-time decision-making [95], where different levels of transformation happen at different nodes. Furthermore, combining edge and cloud architecture in IoT can effectively address network traffic congestion and latency concerns [96].

6. Conclusion

In recent years, the transformation of time-series data into images has become widespread. However, the adoption of these techniques in IoT domains is still in its early stages, with expectations for them to become commonplace across most IoT domains in the near future. This study presents a comprehensive review of image transformation techniques employed in various IoT domains, including smart buildings, industrial settings, energy management, healthcare, security, and more. We categorize existing studies based on their encoding techniques, IoT application areas, and data types. In the literature, various transformation techniques are applied to both UTS and MTS IoT data. These transformation techniques are typically used in conjunction with fusion techniques for multivariate time-series IoT data. Among the techniques employed, GAF and MTF are the most commonly used image transformation techniques, particularly in domains such as energy management, healthcare, and industrial applications with purposes such as anomaly detection, fault diagnosis, and time-series classification. It is crucial to choose the right method that is suitable for the specific problem and dataset. This decision can significantly impact the result and the quality of the resulting images. In addition, handling some important issues such as noise, missing values, and outliers increases the effectiveness of the converted images. Additionally, this paper discusses the associated challenges, weaknesses, limitations, open issues, and future research directions.

References

[1] S. Cc, “A survey on architecture, protocols and challenges in IoT,” *Wireless Personal Communications*, vol. 112, 06 2020.

[2] D. I. Borissova, V. K. Danev, M. B. Rashevski, I. G. Garvanov, R. D. Yoshinov, and M. Z. Garvanova, “Using IoT for automated heating of a smart home by means of openhab software platform,” *IFAC-PapersOnLine*, vol. 55, no. 11, pp. 90–95, 2022, iFAC Workshop on Control for Smart Cities CSC 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896322011430

[3] A. Aditya, S. Anwarul, R. Tanwar, and S. K. V. Koneru, “An IoT-assisted intelligent parking system (IPS) for smart cities,” Procedia Computer Science, vol. 218, pp. 1045–1054, 2023, International Conference on Machine Learning and Data Engineering. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050923000844

[4] S. R., R. M, V. S, S. K. E, Y. S, A. Kumar, J. R. I., and V. K, “A novel autonomous irrigation system for smart agriculture using AI and 6 G-enabled IoT network,” *Microprocessors and Microsystems*, vol. 101, p. 104905, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0141933123001497

[5] S. Juyal, S. Sharma, and A. Shankar Shukla, “Smart skin health monitoring using AI-enabled cloud-based IoT,” Materials Today: Proceedings, vol. 46, pp. 10 539–10 545, 2021, International Conference on Technological Advancements in Materials Science and Manufacturing. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214785321000973

[6] A. Hassan, M. S. Abdul Rahman, W. Md Shah, M. F. I. Othman, and F. Mansourkiaie, “Internet of things based smart shelves prototype implementation,” *Journal of Advanced Computing Technology and Application (JACTA)*, vol. 2, no. 1, pp. 9–14, May 2020. [Online]. Available: https://jacta.utem.edu.my/jacta/article/view/5208

[7] A. Malki, E.-S. Atlam, and I. Gad, “Machine learning approach of detecting anomalies and forecasting time-series of IoT devices,” *Alexandria Engineering Journal*, vol. 61, no. 11, pp. 8973–8986, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1110016822001260

[8] M. Herrera, M. Sasidharan, J. Merino, and A. K. Parlikad, “Handling irregularly sampled IoT time series to inform infrastructure asset management,” *IFAC-PapersOnLine*, vol. 55, no. 19, pp. 241–245, 2022, 5th IFAC Workshop on Advanced Maintenance Engineering, Services and Technologies AMEST 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S240589632201432X

[9] Y. Liu, D. Sun, R. Zhang, and W. Li, “A Method for Detecting Ldos Attacks Based on Hilbert-Huang Transform and Convolutional Neural Network,” *SSRN Electronic Journal*, 2023.

[10] M. Shahin, F. Chen, H. Bouzary, A. Hosseinzadeh, and R. Rashidifar, “A novel fully convolutional neural network approach for detection and classification of attacks on industrial IoT devices in smart manufacturing systems,” *The International Journal of Advanced Manufacturing Technology*, vol. 123, 10 2022.

[11] J. Lin, J. Li, and J. Chen, “An analysis of English classroom behavior by intelligent image recognition in IoT,” *International Journal of System Assurance Engineering and Management*, vol. 13, no. 3, pp. 1063–1071, December 2022.

[12] J. Hong and J. Yoon, "Multivariate time-series classification of sleep patterns using a hybrid deep learning architecture," in *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*, 2017, pp. 1–6.

[13] Y. Xu, Y. Tang, and Q. Yang, "Deep learning for IoT intrusion detection based on LSTMs-ae," in Proceedings of the 2nd International Conference on Artificial Intelligence and Advanced Manufacturing, ser. AIAM2020. New York, NY, USA: Association for Computing Machinery, 2020, p. 64–68. [Online]. Available: https://doi.org/10.1145/3421766.3421891

[14] J. Yang, Y. Sun, Y. Chen, M. Mao, L. Bai, and S. Zhang, "Time series-to-image encoding for saturation line prediction using channel and spatial-wise attention network," *Expert Systems with Applications*, vol. 237, p. 121440, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417423019425

[15] J. Li and Q. Wang, "Comparison of the representational ability in individual difference analysis using 2-d time-series image and time-series feature patterns," *Expert Systems with Applications*, vol. 215, p. 119429, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417422024484

[16] C. Velasco-Gallego and I. Lazakis, "Development of a time series imaging approach for fault classification of marine systems," *Ocean Engineering*, vol. 263, p. 112297, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801822016006

[17] G. Chiarot and C. Silvestri, "Time Series Compression Survey," *ACM Computing Surveys*, vol. 55, no. 10 Feb 2023. [Online]. Available: https://dl.acm.org/doi/10.1145/3560814

[18] X. Wan, H. Li, L. Zhang, and Y. J. Wu, "Dimensionality reduction for multivariate time-series data mining," *Journal of Supercomputing*, vol. 78, no. 7, pp. 9862–9878, 2022. [Online]. Available: https://doi.org/10.1007/s11227-021-04303-4

[19] X. Yu, X. Yang, Q. Tan, C. Shan, and Z. Lv, "An edge computing-based anomaly detection method in IoT industrial sustainability," *Applied Soft Computing*, vol. 128, p. 109486, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494622005841

[20] M. Neyja, S. Mumtaz, K. M. S. Huq, S. A. Busari, J. Rodriguez, and Z. Zhou, "An iot-based e-health monitoring system using ecg signal," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, 2017, pp. 1–6.

[21] J. Azar, A. Makhoul, R. Couturier, and J. Demerjian, "Robust iot time series classification with data compression and deep learning," *Neurocomputing*, vol. 398, pp. 222–234, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231220302939

[22] G. Chen, M. Liu, and Z. Kong, "Temporal-logic-based semantic fault diagnosis with time-series data from industrial internet of things," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 5, pp. 4393–4403, 2021.

[23] I. Niyonambaza, M. Zennaro, and A. Uwitonze, "Predictive maintenance (pdm) structure using internet of things (iot) for mechanical equipment used into hospitals in rwanda," *Future Internet*, vol. 12, no. 12, 2020. [Online]. Available: https://www.mdpi.com/1999-5903/12/12/224

[24] J. Stankovic, "Research directions for the internet of things," *Internet of Things Journal, IEEE*, vol. 1, pp. 3–9, 02 2014.

[25] A. Cook, G. Misirli, and Z. Fan, "Anomaly detection for iot time-series data: A survey," *IEEE Internet of Things Journal*, vol. PP, pp. 1–1, 12 2019.

[26] L. Chettri and R. Bera, "A comprehensive survey on internet of things (iot) toward 5g wireless systems," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 16–32, 2020.

[27] N. Yen, J.-W. Chang, J.-Y. Liao, and Y.-M. Yong, "Analysis of interpolation algorithms for the missing values in iot time series: a case of air quality in taiwan," *The Journal of Supercomputing*, vol. 76, 08 2020.

[28] Y. Liu, Y. Zhou, K. Yang, and X. Wang, "Unsupervised Deep Learning for IoT Time Series," *IEEE Internet of Things Journal*, 2023.

[29] H. Kaur, D. Koundal, and V. Kadyan, "Image fusion techniques: A survey," *Archives of Computational Methods in Engineering*, vol. 28, pp. 4425 – 4447, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:231700128

[30] D. Sudha and M. Ramakrishna, "Comparative study of features fusion techniques," 03 2017, pp. 235–239.

[31] J. Chen and X. Wang, "Non-intrusive Load Monitoring Using Gramian Angular Field Color Encoding in Edge Computing," *Chinese Journal of Electronics*, vol. 31, no. 4, pp. 595–603, 2022.

[32] H. Zhou and C. Kan, "Tensor-based ecg anomaly detection toward cardiac monitoring in the internet of health things," *Sensors*, vol. 21, no. 12, 2021.

[33] P. K. Atrey, M. A. Hossain, A. E. Saddik, and M. Kankanhalli, "Multimodal fusion for multimedia analysis: a survey," *Multimedia Systems*, vol. 16, pp. 345–379, 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:6387482

[34] W. Jiang, D. Zhang, L. Ling, and R. Lin, "Time Series Classification Based on Image Transformation Using Feature Fusion Strategy," *Neural Processing Letters*, vol. 54, no. 5, pp. 3727–3748, oct 2022. [Online]. Available: https://link.springer.com/article/10.1007/s11063-022-10783-z

[35] M. Ehatisham-Ul-Haq, A. Javed, M. A. Azam, H. M. A. Malik, A. Irtaza, I. H. Lee, and M. T. Mahmood, "Robust human activity recognition using multimodal feature-level fusion," *IEEE Access*, vol. 7, pp. 60 736–60 751, 2019.

[36] Y. Han, B. Li, Y. Huang, and L. Li, "Bearing fault diagnosis method based on gramian angular field and ensemble deep learning," *Journal of Vibroengineering*, vol. 25, no. 1, pp. 42–52, oct 2022. [Online]. Available: https://doi.org/10.21595/jve.2022.22796

[37] H. Wei and N. Kehtarnavaz, "Simultaneous utilization of inertial and video sensing for action detection and recognition in continuous action streams," *IEEE Sensors Journal*, vol. 20, no. 11, pp. 6055–6063, 2020.

[38] R. Hang, Z. Li, P. Ghamisi, D. Hong, G. Xia, and Q. Liu, "Classification of hyperspectral and lidar data using coupled cnns," *CoRR*, vol. abs/2002.01144, 2020. [Online]. Available: https://arxiv.org/abs/2002.01144

[39] G. Baldini, R. Giuliani, and F. Dimc, "Physical layer authentication of Internet of Things wireless devices using convolutional neural networks and recurrence plots," *Internet Technology Letters*, vol. 2, no. 2, p. e81, mar 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1002/itl2.81https://onlinelibrary.wiley.com/doi/abs/10.1002/itl2.81https://onlinelibrary.wiley.com/doi/10.1002/itl2.81

[40] C. L. Yang, Z. X. Chen, and C. Y. Yang, "Sensor Classification Using Convolutional Neural Network by Encoding Multivariate Time Series as Two-Dimensional Colored Images," *Sensors 2020, Vol. 20, Page 168*, vol. 20, no. 1, p. 168, dec 2019. [Online]. Available: https://www.mdpi.com/1424-8220/20/1/168/htmhttps://www.mdpi.com/1424-8220/20/1/168

[41] R. John, S. Vijayaraghavan, and R. Pradeep, "A Modern Cardiac Monitoring System Based on Wireless Sensing for Human Health Care," *Proceedings of the 4th International Conference on Communication and Electronics Systems, ICCES 2019*, pp. 750–756, 2019.

[42] M. Fahim, K. Fraz, and A. Sillitti, "Tsi: Time series to imaging based model for detecting anomalous energy consumption in smart buildings," *Information Sciences*, vol. 523, pp. 1–13, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025520301596

[43] C. Lyu, Z. Huo, X. Cheng, J. Jiang, A. Alimasi, and H. Liu, "Distributed Optical Fiber Sensing Intrusion Pattern Recognition Based on GAF and CNN," *JOURNAL OF LIGHTWAVE TECHNOLOGY*, vol. 38, no. 15, 2020. [Online]. Available: https://www.ieee.org/publications/rights/index.html

[44] A. Estebsari and R. Rajabi, "Single Residential Load Forecasting Using Deep Learning and Image Encoding Techniques," *Electronics 2020, Vol. 9, Page 68*, vol. 9, no. 1, p. 68, jan 2020. [Online]. Available: https://www.mdpi.com/2079-9292/9/1/68/htmhttps://www.mdpi.com/2079-9292/9/1/68

[45] A. Ferraro, A. Galli, V. Moscato, and G. Sperli, "A novel approach for predictive maintenance combining GAF encoding strategies and deep networks," *Proceedings - 2020 IEEE 6th International Conference on Dependability in Sensor, Cloud and Big Data Systems and Application, DependSys 2020*, pp. 127–132, 2020.

[46] H. Xu, J. Li, H. Yuan, Q. Liu, S. Fan, T. Li, and X. Sun, "Human activity recognition based on gramian angular field and deep convolutional neural network," *IEEE Access*, vol. 8, pp. 199 393–199 405, 2020.

[47] K. Vandith Sreenivas, M. Ganesan, and R. Lavanya, "Classification of Arrhythmia in Time Series ECG Signals Using Image Encoding and Convolutional Neural Networks," *Proceedings of 2021 IEEE 7th International Conference on Bio Signals, Images and Instrumentation, ICBSII 2021*, 2021.

[48] B. Zhu, X. Hou, S. Liu, W. Ma, M. Dong, H. Wen, Q. Wei, S. Du, and Y. Zhang, "IoT Equipment Monitoring System Based on C5.0 Decision Tree and Time-series Analysis," *IEEE Access*, vol. 10, pp. 1–1, jan 2021. [Online]. Available: https://typeset.io/papers/iot-equipment-monitoring-system-based-on-c5-0-decision-tree-54megjwwoi

[49] A. K. A, G. M, and L. R, "Emotional classification of eeg signal using image encoding and deep learning," in *2021 Seventh International conference on Bio Signals, Images, and Instrumentation (ICBSII)*, 2021, pp. 1–5.

[50] N. Sharma, M. Mangla, S. N. Mohanty, D. Gupta, P. Tiwari, M. Shorfuzzaman, and M. Rawashdeh, "A smart ontology-based IoT framework for remote patient monitoring," *Biomedical Signal Processing and Control*, vol. 68, p. 102717, 2021. [Online]. Available: https://doi.org/10.1016/j.bspc.2021.102717

[51] Y. Chen, H. Deng, D. Zhang, and Y. Hu, "SpeedNet: Indoor Speed Estimation with Radio Signals," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2762–2774, feb 2021.

[52] J. R. Jiang and C. T. Yen, "Product quality prediction for wire electrical discharge machining with markov transition fields and convolutional long short-term memory neural networks," *Applied Sciences (Switzerland)*, vol. 11, no. 13, 2021.

[53] G. R. Garcia, G. Michau, M. Ducoffe, J. S. Gupta, and O. Fink, "Temporal signals to images: Monitoring the condition of industrial assets with deep learning image processing algorithms," *arXiv: Learning*, vol. 236, no. 4, pp. 617–627, may 2020. [Online]. Available: https://typeset.io/papers/ time-series-to-images-monitoring-the-condition-of-industrial-2c2m2v1l2a

[54] T. Huang, P. Chakraborty, and A. Sharma, "Deep convolutional generative adversarial networks for traffic data imputation encoding time series as images," *International Journal of Transportation Science and Technology*, vol. 12, no. 1, pp. 1–18, 2023. [Online]. Available: https://doi.org/10.1016/j.ijtst.2021.10.007

[55] Q. Jiang, S. Zhang, J. Chen, X. Chen, H. Huang, and C. Gu, "Adaptive dila-densenet for image based time series classification in iot," in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–8.

[56] R. Singh, T. Ahmed, A. Kumar Singh, P. Chanak, and S. K. Singh, "SeizSClas: An Efficient and Secure Internet of-Things-Based EEG Classifier," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6214–6221, apr 2021.

[57] A. De Santo, A. Ferraro, A. Galli, V. Moscato, and G. Sperl`ı, "Evaluating time series encoding techniques for Predictive Maintenance," *Expert Systems with Applications*, vol. 210, no. August, p. 118435, 2022. [Online]. Available: https://doi.org/10.1016/j.eswa.2022.118435

[58] B. Bertalanič, M. Meza, and C. Fortuna, "Resource-Aware Time Series Imaging Classification for Wireless Link Layer Anomalies," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 14, no. 8, pp. 1–13, 2022.

[59] A. Alsalemi, A. Amira, H. Malekmohamadi, and K. Diao, "Lightweight Gramian Angular Field classification for edge internet of energy applications," *Cluster Computing*, vol. 26, no. 2, pp. 1375–1387, 2022. [Online]. Available: https://doi.org/10.1007/s10586-022-03704-1

[60] P. Zhang, Y. Hang, X. Ye, P. Guan, J. Jiang, J. Tan, and W. Hu, "A United CNN-LSTM Algorithm Combining RR Wave Signals to Detect Arrhythmia in the 5G-Enabled Medical Internet of Things," *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 14 563–14 571, 2022.

[61] S. Dou, S. Shao, C. Song, H. Shi, and H. Zhao, "Electrocardiogram Signal Classification Algorithm Based on The Continuous Wavelet Transform and GoogleNet in an Internet of Things Context," *Journal of Mechanics in Medicine and Biology*, vol. 22, no. 9, p. 2240049, nov 2022. [Online]. Available: www.worldscientific.com

[62] M. Abdel-Basset, H. Hawash, V. Chang, R. K. Chakrabortty, and M. Ryan, "Deep Learning for Heterogeneous Human Activity Recognition in Complex IoT Applications," *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 5653–5665, 2022.

[63] B. Wang, G. Feng, D. Huo, and Y. Kang, "A Bearing Fault Diagnosis Method Based on Spectrum Map Information Fusion and Convolutional Neural Network," *Processes*, vol. 10, no. 7, 2022.

[64] T. Bai, X. Li, and S. Ding, "Research on Electrical Parameter Fault Diagnosis Method of Oil Well Based on TSC DCGAN Deep Learning," *2022 3rd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering, ICBAIE 2022*, pp. 753–761, 2022.

[65] A. Abidi, D. Ienco, A. B. Abbes, and I. R. Farah, "Combining 2D encoding and convolutional neural network to enhance land cover mapping from Satellite Image Time Series," *Engineering Applications of Artificial Intelligence*, vol. 122, no. March, p. 106152, 2023. [Online]. Available: https://doi.org/10.1016/j.engappai.2023.106152

[66] P. O. de Paula, T. B. da Silva Costa, R. R. de Faissol Attux, and D. G. Fantinato, "Classification of image encoded SSVEP-based EEG signals using Convolutional Neural Networks," *Expert Systems with Applications*, vol. 214, no. July 2022, p. 119096, 2023. [Online]. Available: https://doi.org/10.1016/j.eswa.2022.119096

[67] S. Quan, M. Sun, X. Zeng, X. Wang, and Z. Zhu, "Time Series Classification Based on Multi-Dimensional Feature Fusion," *IEEE Access*, vol. 11, pp. 11 066–11 077, 2023.

[68] Y. Zhang, L. Shang, H. Gao, Y. He, X. Xu, and Y. Chen, "A New Method for Diagnosing Motor Bearing Faults Based on Gramian Angular Field Image Coding and Improved CNN-ELM," *IEEE Access*, vol. 11, pp. 11 337–11 349, 2023.

[69] A. Copiaco, Y. Himeur, A. Amira, W. Mansoor, F. Fadli, S. Atalla, and S. S. Sohail, "An innovative deep anomaly detection of building energy consumption using energy time-series images," *Engineering Applications of Artificial Intelligence*, vol. 119, no. September 2022, p. 105775, 2023. [Online]. Available: https://doi.org/10.1016/j.engappai.2022.105775

[70] L. Qu, Y. Kong, M. Li, W. Dong, F. Zhang, and H. Zou, "A residual convolutional neural network with multi-block for appliance recognition in non-intrusive load identification," *Energy and Buildings*, vol. 281, p. 112749, feb 2023.

[71] X. Sun, M. Wang, B. Zhan, Y. Xiong, and W. Yu, "An Intelligent Diagnostic Method for Multisource Coupling Faults of Complex Mechanical Systems," *Shock and Vibration*, vol. 2023, 2023.

[72] A. N. Sayed, Y. Himeur, and F. Bensaali, "From time-series to 2D images for building occupancy prediction using deep transfer learning," *Engineering Applications of Artificial Intelligence*, vol. 119, no. January, p. 105786, 2023. [Online]. Available: https://doi.org/10.1016/j.engappai.2022.105786

[73] M. N. Hasan, S. U. Jan, and I. Koo, "Wasserstein GAN-based Digital Twin Inspired Model for Early Drift Fault Detection in Wireless Sensor Networks," *IEEE Sensors Journal*, vol. XX, no. Xx, pp. 1–14, 2023.

[74] M. Hammoud, E. Kovalenko, A. Somov, E. Bril, and A. Baldycheva, "Deep learning framework for neurological diseases diagnosis through near-infrared eye video and time series imaging algorithms," *Internet of Things*, vol. 24, p. 100914, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542660523002378

[75] Y. Yan, Y. Yang, F. Shen, M. Gao, and Y. Gu, "Gde model: A variable intrusion detection model for few-shot attack," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 10, p. 101796, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157823003506

[76] Z. Wang and T. Oates, "Encoding time series as images for visual inspection and classification using tiled convolu tional neural networks," *AAAI Workshop - Technical Report*, vol. WS-15-14, no. January, pp. 40–46, 2015.

[77] J. P. Eckmann, O. Oliffson Kamphorst, and D. Ruelle, "Recurrence plots of dynamical systems," *Epl*, vol. 4, no. 9, pp. 973–977, 1987.

[78] K. Gr¨ochenig, *The Short-Time Fourier Transform*. Boston, MA: Birkh¨auser Boston, 2001, pp. 37–58.

[79] V. J. Bol´os and R. Ben´ıtez, *The Wavelet Scalogram in the Study of Time Series*. Cham: Springer International Publishing, 2014, pp. 147–154.

[80] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N. C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proceedings of the Royal Society of London Series A*, vol. 454, no. 1971, pp. 903–998, Mar. 1998.

[81] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-driven fault diagnosis method," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5990–5998, 2018.

[82] L. De Baets, J. Ruyssinck, C. Develder, T. Dhaene, and D. Deschrijver, "Appliance classification using vi trajectories and convolutional neural networks," *Energy and Buildings*, vol. 158, pp. 32–36, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378778817312690

[83] Z. Yang, I. A. Abbasi, F. Algarni, S. Ali, and M. Zhang, "An IoT Time Series Data Security Model for Adversarial Attack Based on Thermometer Encoding," *Security and Communication Networks*, vol. 2021, 2021.

[84] C.-L. Fan and J.-R. Jiang, "Surface roughness prediction based on markov chain and deep neural network for wire electrical discharge machining," in *2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*, 2019, pp. 191–194.

[85] C. Fu, M. Quintana, Z. Nagy, and C. Miller, "Filling time-series gaps using image techniques: Multidimensional context autoencoder approach for building energy data imputation," *Applied Thermal Engineering*, vol. 236, p. 121545, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1359431123015740

[86] Z. Guo, Y. Wan, and H. Ye, "A data imputation method for multivariate time series based on generative adversarial network," *Neurocomputing*, vol. 360, pp. 185–197, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231219308306

[87] M. Ghahramani, R. Taheri, M. Shojafar, S. Member, R. Javidan, and S. Wan, "Deep Image: A precious image-based deep learning method for online malware detection in IoT Environment," 2022.

[88] A. Ghasemieh and R. Kashef, "A robust deep learning model for predicting the trend of stock market prices during market crash periods," in *2022 IEEE International Systems Conference (SysCon)*, 2022, pp. 1–8.

[89] L. Yu, J. Li, T. Wang, F. Tan, C. He, and H. Song, "T2i-net: Time series classification via deep sequence-to-image transformation networks," in *2022 IEEE International Conference on Networking, Sensing and Control (ICNSC)*, 2022, pp. 1–5.

[90] J. Debayle, N. Hatami, and Y. Gavet, "Classification of time-series images using deep convolutional neural net works," 04 2018, p. 23.

[91] X. Yuan, D. Tanksley, P. Jiao, L. Li, G. Chen, and D. Wunsch, "Encoding Time-Series Ground Motions as Images for Convolutional Neural Networks-Based Seismic Damage Evaluation," *Frontiers in Built Environment*, vol. 7, p. 52, apr 2021.

[92] K. S. Kiangala and Z. Wang, "An effective predictive maintenance framework for conveyor motors using dual time-series imaging and convolutional neural network in an industry 4.0 environment," *IEEE Access*, vol. 8, pp. 121 033–121 049, 2020.

[93] M. Ashraf, F. Anowar, J. H. Setu, A. I. Chowdhury, E. Ahmed, A. Islam, and A. Al-Mamun, "A survey on dimensionality reduction techniques for time-series data," *IEEE Access*, vol. 11, pp. 42 909–42 923, 2023.

[94] H. Jiang, L. Liu, and C. Lian, "Multi-modal fusion transformer for multivariate time series classification," in *2022 14th International Conference on Advanced Computational Intelligence (ICACI)*, 2022, pp. 284–288.

[95] H. Nizam, S. Zafar, Z. Lv, F. Wang, and X. Hu, "Real-time deep anomaly detection framework for multivariate time-series data in industrial iot," *IEEE Sensors Journal*, vol. 22, no. 23, pp. 22 836–22 849, 2022.

[96] I. Ali, H. Bayomi, and K. Wassif, "Dimensionality reduction for images of iot using machine learning," 2023. [Online]. Available: https://doi.org/10.21203/rs.3.rs-2666777/v1

Author(s) Contributions

This article was prepared by multiple authors. The contributions of each author are as follows:

Duygu Altunkaya contributed to the conception and design of the study, data analysis and interpretation, writing the technical sections of the paper, critically reviewed its content, and literature review. Feyza Yıldırım Okay contributed to the conception and design of the study, writing the technical sections of the paper, critically reviewed its content, and the literature review. Suat Özdemir contributed to the overall coordination, the conception and design of the study, critically reviewed the manuscript's content and the literature review.

Conflict of Interest Notice

The authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical Approval and Informed Consent

It is declared that during the preparation process of this study, scientific and ethical principles were followed, and all the studies benefited from are stated in the bibliography.

Availability of data and material

Not applicable

Artificial Intelligence Statement

ChatGPT was used solely for grammar and spelling corrections during the preparation of the article. The scientific content, analysis, and interpretation were produced entirely by the authors.

Plagiarism Statement

This article has been scanned by iThenticate ™.