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

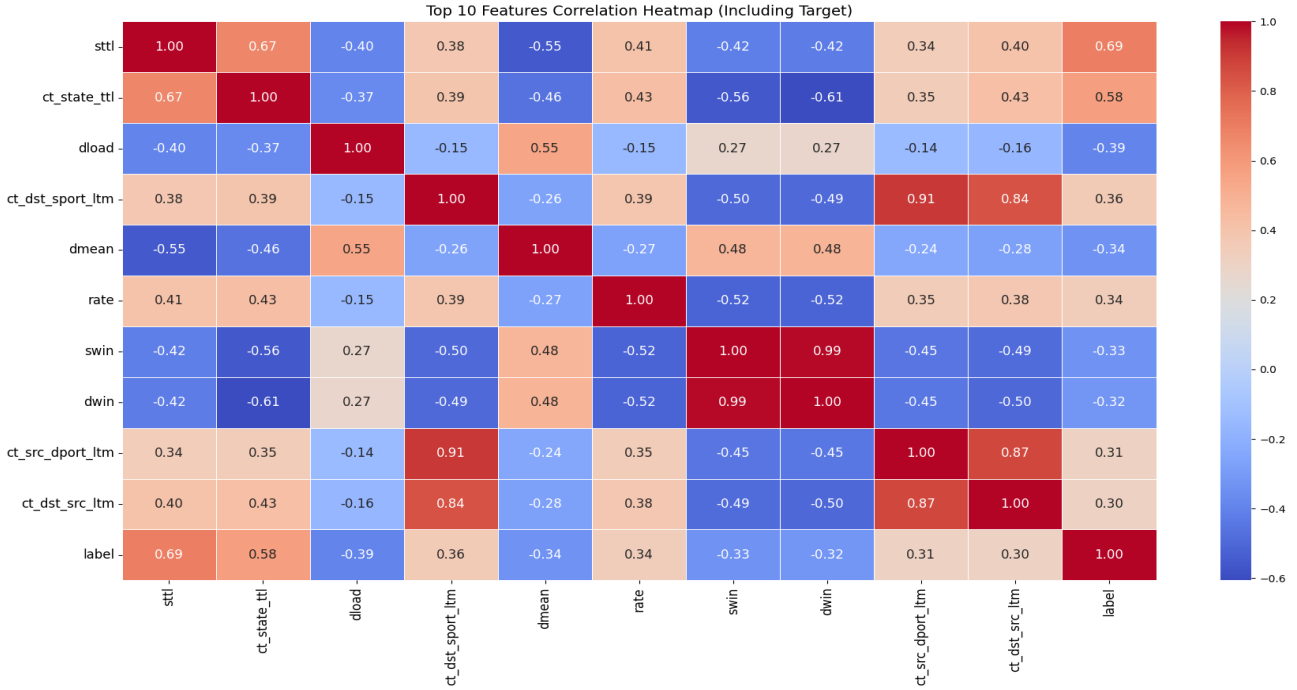
	* Web-based	
CICIDS-2017	* DoS * DDoS * Brute force * XSS	* SQL injection * Infiltration * Port scan * Botnet
UNSW-NB15	* Fuzzers * Analysis * Backdoors * DoS * Exploits	* Generic * Reconnaissance * Shellcode * Worms
IoMT – 2024	* ARP spoofing * Ping Sweep * Recon VulScan * OS Scan * Port Scan	* MQTT 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.

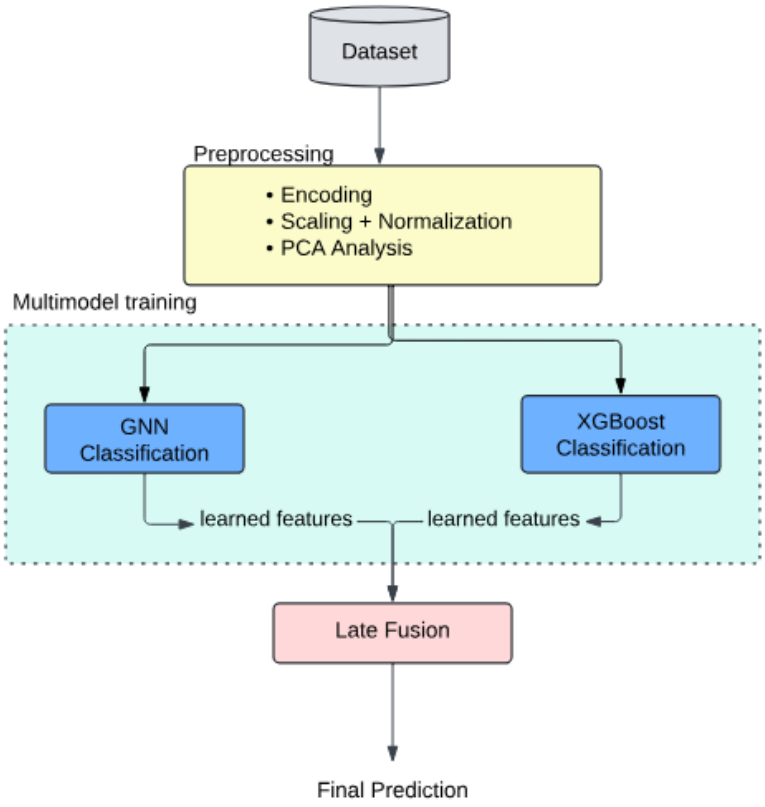


Figure 2. Architectural Framework for Proposed Model

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.

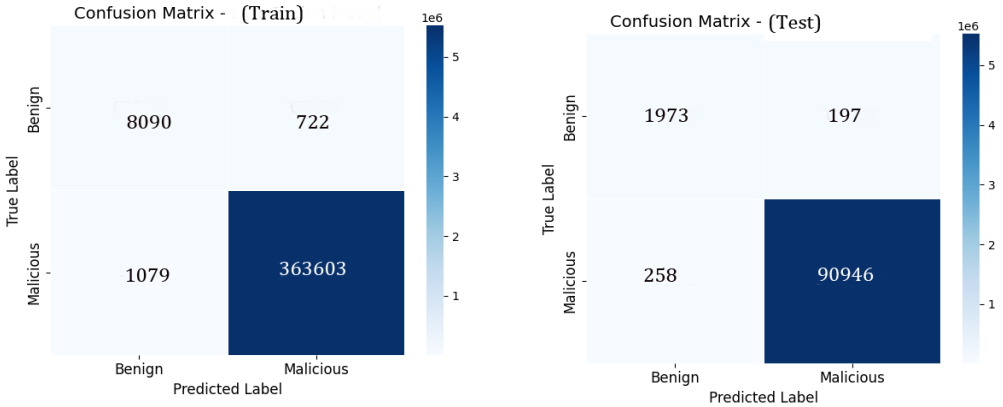


Figure 3. Training and Testing Confusion Matrix for CICIoT-2023.

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.

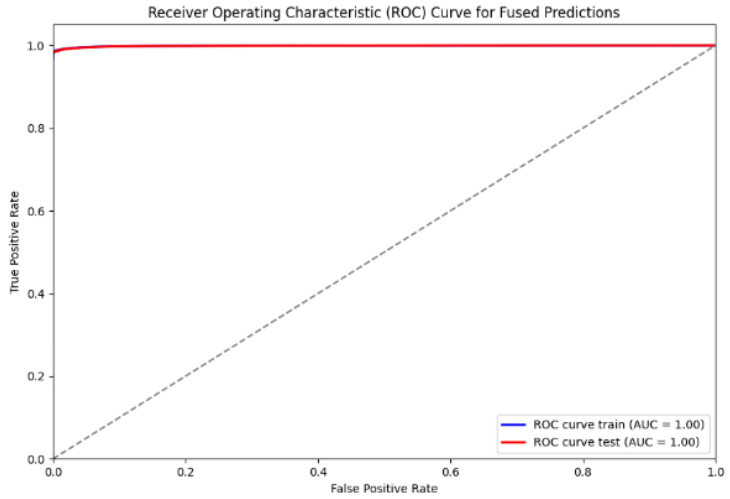


Figure 4. The ROC Curve for CICIoT-2023 Dataset Training Model

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.

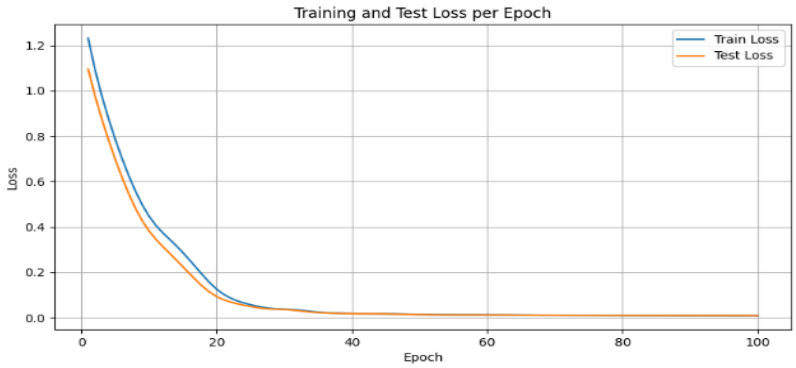


Figure 5. Training and testing loss

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.

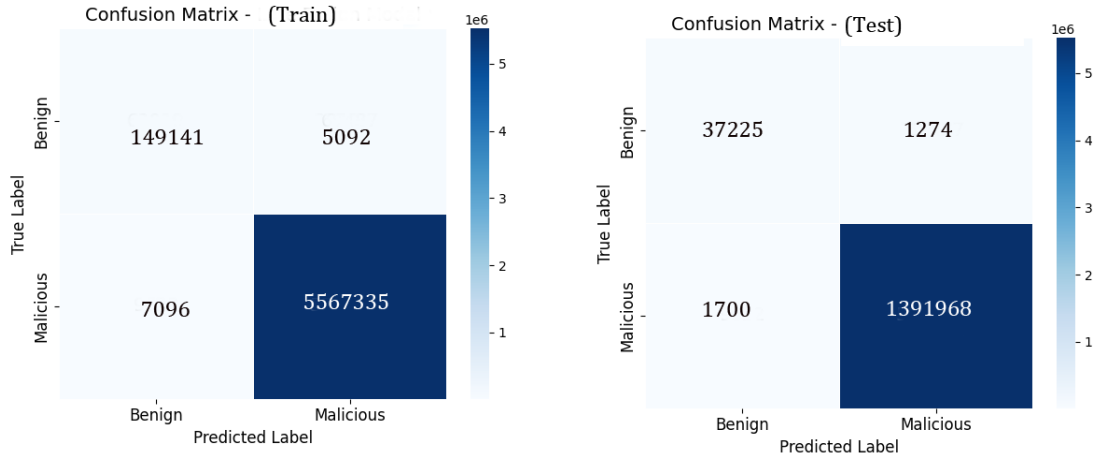


Figure 6. Training and testing the confusion matrix for the IoMT dataset.

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.

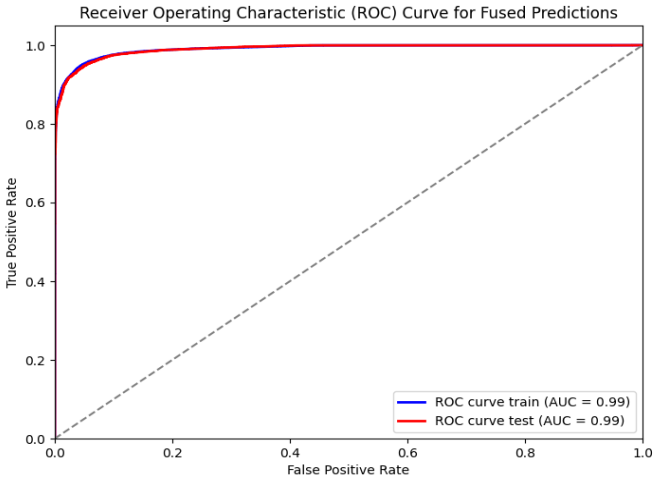


Figure 7. The ROC curve for the IoMT-2024 dataset training model

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.

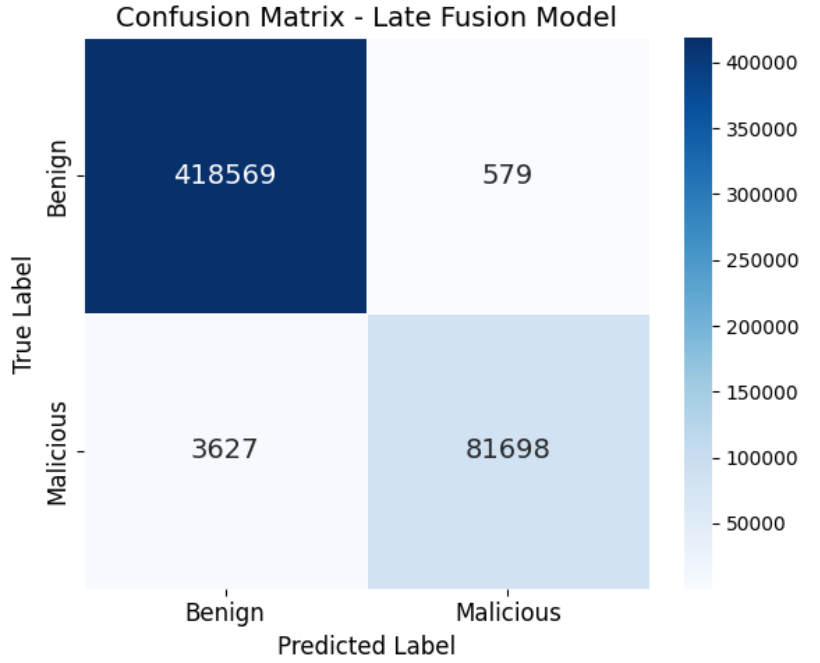


Figure 8. Confusion matrix for the CICIDS-2017 dataset.

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.

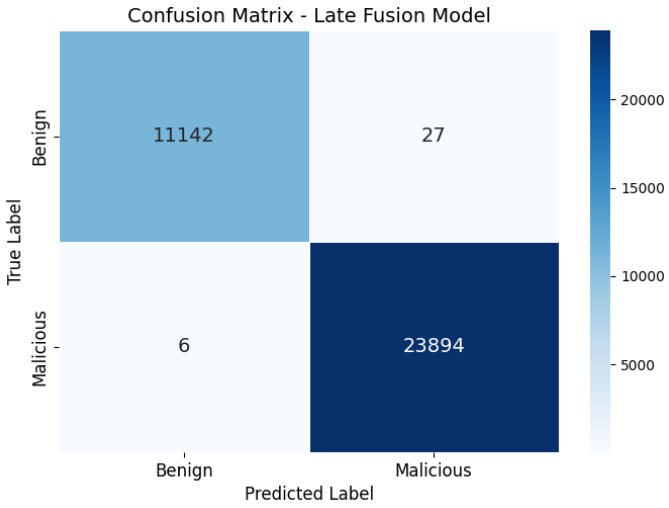


Figure 9. Training and testing confusion matrix for UNSW-NB15 dataset

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.