










Article

A Deep Learning-Based Approach for the Detection of Various Internet of Things Intrusion Attacks Through Optical Networks

Nouman Imtiaz ¹, Abdul Wahid ², Syed Zain Ul Abideen ², Mian Muhammad Kamal ^{3,*}, Nabila Sehito ⁴, Salahuddin Khan ⁵, Bal S. Virdee ⁶, Lida Kouhalvandi ⁷ and Mohammad Alibakhshikenari ^{8,*}

¹ School of Computer Science and Technology, Shandong University, Qingdao 266510, China; nouman.imtiaz660@gmail.com

² College of Computer Science and Technology, Qingdao University, Qingdao 266071, China; wahidjan999@gmail.com (A.W.); zain208shah@gmail.com (S.Z.U.A.)

³ School of Electronic Science and Engineering, Southeast University, No. 2 Southeast University Road, Jiangning, Nanjing 211189, China

⁴ Department of Computer Science, ILMA University, Karachi 74900, Pakistan; nabila.fiza@gmail.com

⁵ College of Engineering, King Saud University, P.O. Box 800, Riyadh 11421, Saudi Arabia; drskhan@ksu.edu.sa

⁶ Center for Communications Technology, London Metropolitan University, London N7 8DB, UK; b.virdee@londonmet.ac.uk

⁷ Department of Electrical and Electronics Engineering, Dogus University, Istanbul 34775, Turkey; lkouhalvandi@dogus.edu.tr

⁸ Electronics Engineering Department, University of Rome "Tor Vergata", 00133 Rome, Italy

* Correspondence: mmkamal@seu.edu.cn (M.M.K.); alibakhshikenari@ing.uniroma2.it (M.A.)

Abstract: The widespread use of the Internet of Things (IoT) has led to significant breakthroughs in various fields but has also exposed critical vulnerabilities to evolving cybersecurity threats. Current Intrusion Detection Systems (IDSs) often fail to provide real-time detection, scalability, and interpretability, particularly in high-speed optical network environments. This research introduces XIoT, which is a novel explainable IoT attack detection model designed to address these challenges. Leveraging advanced deep learning methods, specifically Convolutional Neural Networks (CNNs), XIoT analyzes spectrogram images transformed from IoT network traffic data to detect subtle and complex attack patterns. Unlike traditional approaches, XIoT emphasizes interpretability by integrating explainable AI mechanisms, enabling cybersecurity analysts to understand and trust its predictions. By offering actionable insights into the factors driving its decision making, XIoT supports informed responses to cyber threats. Furthermore, the model's architecture leverages the high-speed, low-latency characteristics of optical networks, ensuring the efficient processing of large-scale IoT data streams and supporting real-time detection in diverse IoT ecosystems. Comprehensive experiments on benchmark datasets, including KDD CUP99, UNSW NB15, and Bot-IoT, demonstrate XIoT's exceptional accuracy rates of 99.34%, 99.61%, and 99.21%, respectively, significantly surpassing existing methods in both accuracy and interpretability. These results highlight XIoT's capability to enhance IoT security by addressing real-world challenges, ensuring robust, scalable, and interpretable protection for IoT networks against sophisticated cyber threats.

Keywords: Internet of Things; intrusion detection systems; deep learning; explainable AI; spectrogram; network attacks; diverse approach; optimization methods; optical network



Received: 2 November 2024

Revised: 18 December 2024

Accepted: 25 December 2024

Published: 3 January 2025

Citation: Imtiaz, N.; Wahid, A.; Ul Abideen, S.Z.; Muhammad Kamal, M.; Sehito, N.; Khan, S.; Virdee, B.S.; Kouhalvandi, L.; Alibakhshikenari, M. A Deep Learning-Based Approach for the Detection of Various Internet of Things Intrusion Attacks Through Optical Networks. *Photonics* **2025**, *12*, 35. <https://doi.org/10.3390/photonics12010035>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The global perception of the Internet and its services among users has experienced a substantial rise in recent years, and this upward trajectory is anticipated to persist into

the foreseeable future [1]. The Internet's interconnected nature has transformed daily life, fostering innovations that span personal, commercial, and industrial domains. With the widespread adoption of electronic devices such as smartphones, smart appliances, and industrial sensors, the Internet of Things (IoT) has emerged as a defining technological advancement. IoT connects physical devices to the digital world, allowing real-time data exchange and automation across sectors. Cyber-Physical Systems (CPSs), integral to critical infrastructures such as healthcare, water management, power grids, and agriculture, are increasingly reliant on IoT. However, with this rapid adoption comes the challenge of securing billions of connected devices. By 2018, there were an estimated 7 billion IoT devices in operation, which is a number projected to surge to over 20 billion by 2020 [2].

As IoT devices become ubiquitous, so too do the vulnerabilities they introduce. These devices, which often operate with limited security features, are prime targets for cyber-attacks. Symantec's 2018 report highlighted over 57,000 attacks per month on IoT devices, underscoring the urgent need for comprehensive security frameworks [3]. Traditional security measures, typically focused on data acquisition and reactive defenses, are no longer sufficient. To counter emerging threats, especially in IoT environments, a paradigm shift toward proactive, prevention-focused security measures is critical. This requires the adoption of advanced Intrusion Detection Systems (IDSs), particularly those capable of real-time threat detection and mitigation.

A significant challenge in securing IoT networks lies in the complexity of modern cybercrimes. One of the most prevalent and dangerous forms of attack is the botnet. Botnets, like the infamous Mirai botnet, exploit IoT vulnerabilities to orchestrate large-scale Distributed Denial of Service (DDoS) attacks. The Mirai attack, which peaked at 1.1 Tbps, demonstrated the destructive potential of compromised IoT devices when coordinated at scale. The evolution of botnets now includes features like encrypted command-and-control (CC) communications, which help them evade detection and execute multi-vector attacks [4]. The increasing sophistication of such attacks highlights the need for more advanced detection mechanisms capable of interpreting complex attack patterns in real-time.

In response to these challenges, our research introduces the Explainable Internet of Things (XIoT) model, which is a novel detection system specifically designed for IoT environments. While applying Convolutional Neural Networks (CNNs) to IoT traffic for intrusion detection is a well-known technique, the XIoT model stands apart by introducing several key innovations. It harnesses the power of CNNs combined with Explainable AI (XAI) to provide deeper interpretability and transparency in the decision-making process, which is critical in real-world cybersecurity applications. This integration of CNNs with XAI ensures that both the spatial and temporal features of spectrogram images derived from IoT network traffic are efficiently analyzed, facilitating a more nuanced detection of cyber threats.

These spectrograms transform raw network traffic into visual data, enabling CNNs to identify intricate attack patterns that are otherwise difficult to capture using conventional methods. What sets XIoT apart from existing methods is its ability to offer both high detection accuracy and a transparent, interpretable model that aids cybersecurity experts in understanding the reasoning behind the system's decisions, which is a capability largely absent from most current IDS solutions [5].

In the context of IoT security, Network Intrusion Detection Systems (NIDSs) are indispensable for safeguarding IoT infrastructures from common attack vectors such as Denial of Service (DoS), Probe Attacks, Remote to Local (R2L), and User to Root (U2R) attacks [6]. Traditional NIDS solutions, however, often struggle to adapt to the high variability and sheer volume of data generated by IoT devices. The XIoT model addresses this gap by employing a dual-focus analysis of IoT network traffic through CNN-based models, handling the

scale and complexity of IoT traffic effectively. It identifies novel attack patterns, anticipates potential security breaches, and provides real-time defenses. By focusing on IoT-specific threats, such as botnets, and leveraging the interpretive power of CNNs enhanced with XAI, XIoT significantly improves the overall security posture of IoT networks.

Beyond its technical capabilities, the XIoT model also improves upon existing models by enhancing interpretability, which is a critical aspect for cybersecurity practitioners. In complex IoT environments, understanding the nature and behavior of an attack is essential for implementing effective countermeasures. The XIoT model's explainable intrusion detection decisions empower security analysts with actionable insights, facilitating faster and more accurate responses to evolving threats. This is a significant departure from the "black box" nature of many ML-based intrusion detection systems, which often leave analysts with limited understanding of the detection process.

Furthermore, the rising interconnectivity of devices, often referred to as Explainable IoT (XIoT), which integrates traditional IoT with operational technology (OT) and industrial control systems (ICSs), presents additional security challenges. XIoT environments extend the attack surface significantly, necessitating even more robust and scalable detection mechanisms. The XIoT model is specifically designed to meet these demands, offering advanced detection capabilities that align with the increasing complexity and scale of these interconnected systems. By rigorously evaluating the model across diverse datasets such as KDD CUP99, UNSW NB15, and Bot-IoT, we demonstrate its ability to generalize across different IoT environments, offering a novel solution that outperforms current intrusion detection models in terms of accuracy, precision, recall, and F1-score.

Ultimately, the XIoT model is a direct response to the pressing security challenges posed by the proliferation of IoT devices and their associated cyber threats. By leveraging advanced ML techniques, specifically CNNs integrated with explainable AI, the XIoT model enhances the detection, interpretation, and prevention of modern cyberattacks in IoT environments. The ability to adapt to evolving attack patterns and provide real-time, actionable insights positions XIoT as a cutting-edge solution for ensuring the confidentiality, integrity, and availability (CIA) of IoT systems [7,8]. As IoT continues to evolve, so too must the systems designed to protect it, and the XIoT model offers a promising path forward in advancing the state of IoT cybersecurity.

The proliferation of IoT devices connected through high-speed optical networks introduces unique challenges, such as managing heterogeneous data streams with minimal latency and ensuring robust security against increasingly sophisticated cyber threats. Existing IDSs lack the efficiency and interpretability required for real-time detection in such environments. This research aims to bridge this gap by leveraging spectrogram-based data transformation and CNNs to enhance both accuracy and explainability.

1.1. Related Work

ML techniques have been extensively employed to detect various types of cyber attacks, enabling network administrators to implement preventative measures against intrusions. Initially, traditional ML methods such as SVM [9], k-Nearest Neighbor (KNN) [10], RF [11], Naïve Bayes Network [12], and Self-Organizing Maps (SOMs) [13] were utilized in IDS and demonstrated promising results. Reference [14] assessed the efficacy of various ML classifiers using the NSL-KDD dataset. However, these traditional methods, characterized as shallow learning, primarily focus on feature engineering and selection, and they are often inadequate for managing the complexities of large-scale data classification in real network environments [15,16]. As datasets expand, the limitations of shallow learning become apparent, particularly in high-dimensional analysis required for intelligent forecasting.

Conversely, DL offers enhanced capabilities for extracting significant representations from data, thus improving model performance. Recent research has explored the application of DL in network intrusion detection, which is a relatively novel field. For instance, DL approaches like the three-layer RNN proposed by [17] with 41 features and four output categories, despite its partial inter-layer connectivity, signify advancements in handling high-dimensional features. Additionally, Torres et al. [18] transformed feature data into character sequences to analyze temporal characteristics using RNN. In [14], a specific RNN-IDS model was introduced for direct classification, comparing its performance against traditional methods such as J48, ANN, RF, and SVM on the NSL-KDD dataset in binary and multi-class scenarios. Wang et al. [19] integrated both CNN and RNN to maximize the deep neural network's ability to learn spatial-temporal features from raw network traffic data.

As new viruses emerge and intrusion behaviors evolve, IDSs continue to innovate, integrating data mining and ML technologies to enhance detection capabilities [20]. For example, an adaptive chicken colony optimization algorithm for efficient clustering in selecting cluster heads was introduced in [21], along with a two-stage adaptive SVM classification to identify malicious sensor nodes, thereby reducing time consumption and enhancing network lifespan and scalability. Other authors developed an IDS model utilizing a double sparse convolution matrix framework which leverages the strong correlations in non-negative matrix decomposition to reveal hidden patterns and achieve high detection accuracy. However, despite these advancements, traditional ML-based IDSs still face significant challenges due to their reliance on complex mathematical calculations [22].

Various methods have been developed to enhance attack detection in IoT networks. In prior research, a feed-forward neural network achieved high accuracy using the BoT-IoT dataset, though with lower precision and recall in some categories [23]. Reference [24] introduced a hybrid IDS combining feature selection and ensemble learning, significantly improving accuracy to 99.9%. Reference [25] employed an LSTM autoencoder for dimensionality reduction, followed by a Bi-LSTM, which improved performance at the expense of increased computational time.

Reference [26] used a bi-directional LSTM, achieving high detection rates for DoS attacks in cloud networks, although it struggled with non-DoS traffic like reconnaissance attacks. Reference [27] proposed Deep-IFS, a forensic model enhanced with multi-head attention, outperforming centralized DL models but requiring numerous fog nodes for optimal performance. Reference [28] applied correlation-based feature selection with various ML algorithms, achieving high detection rates. Reference [29] developed a Deep Belief Network (DBN)-based IDS with strong accuracy, while other studies utilized CNN-LSTM combinations [30,31], achieving high detection rates across different attack types. Another novel IDS combined CNNs with stacked autoencoders for feature extraction, showing high performance. However, methods involving RNNs with self-attention mechanisms demanded extensive preprocessing time.

Research has also explored hybrid feature selection techniques like ant colony optimization and mutual information, which proved effective in improving detection with decision trees [29]. Metaheuristic algorithms, such as particle swarm optimization (PSO) and genetic algorithms (GAs), have been used for feature selection to enhance IDS performance [32–37]. Despite these advances, there remains a gap in integrating deep learning with metaheuristic approaches for further improvement in IoT-based IDSs.

One-class classification is a key anomaly detection method particularly suited for datasets where one class dominates, such as in intrusion detection, where normal network traffic far exceeds attack instances. It utilizes algorithms like Meta-Learning [38], Interpolated Gaussian Descriptor [39], One-Class Support Vector Machine (OCSVM) [40–45],

and Autoencoders [46–52]. OCSVM is particularly effective with small datasets, as demonstrated by [45], who enhanced it with hyperparameter optimization, creating a scalable and distributed IDSs for IoT, which was assessed with ensemble learning.

Autoencoders, such as the stacked self-encoder model from Song [50], are increasingly favored as datasets grow, offering stable performance with optimization through latent layer adjustments. Ensemble learning [53–59] has also shown promise, integrating multiple weak learners to improve overall accuracy. Reference [56] introduced an ensemble voting classifier for IoT intrusion detection, while [58] used a genetic algorithm for feature selection combined with SVM and DT classifiers. Reference [59] developed a two-layer soft-voting model using RF, lightGBM, and XGBoost, achieving superior accuracy in both binary and multi-class scenarios.

Reference [60] presented an ensemble IDS for IoT environments, mitigating botnet attacks using DNS, HTTP, and MQTT protocols. The method employed AdaBoost with DT, naive Bayes, and artificial neural networks, using the UNSW-NB15 dataset. Additionally, reference [61] proposed a Dew Computing as a Service model to improve the IDS performance in Edge of Things (EoT) systems, integrating Deep Belief Networks (DBNs) with restricted Boltzmann machines for real-time attack classification. [62] introduced MemAE, a memory-augmented autoencoder that improves anomaly detection by guiding reconstruction toward normal data characteristics, enhancing detection accuracy. Furthermore, SVD and SMOTE were applied to improve feature condensation and balance, achieving 99.99% accuracy in binary classification and 99.98% in multi-class classification using the ToN_IoT dataset.

Finally, reference [63] introduced the Deep Random Neural Network, combining Particle Swarm Optimization (PSO) and Sequential Quadratic Programming to enhance attack detection in IIoT settings. The model demonstrated superior performance across both binary and multi-class scenarios using multiple IIoT datasets. Deep learning (DL) models have become integral in intrusion detection due to their ability to automatically learn hierarchical representations from network traffic data. Unlike traditional ML models, DL methods like Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) do not rely on manual feature extraction, making them more effective in handling complex patterns in real-time attack data.

RNNs, particularly Long Short-Term Memory (LSTM) networks, have demonstrated significant improvements in detecting temporal attack patterns. For example, reference [14] applied an LSTM-based IDS to the NSL-KDD dataset, achieving notable accuracy for probe and DoS attacks but struggling with less frequent attack types like R2L and U2R. Despite this limitation, RNNs are particularly advantageous in capturing sequential dependencies in network traffic data. On the other hand, CNNs have been successfully applied to analyze network traffic represented as images or spectrograms. In this study, the novel XIoT model utilizes CNNs to examine spectrogram images of IoT traffic data, emphasizing the interpretability of attack detection. Unlike RNNs, CNNs are primarily used for spatial feature extraction, which makes them suitable for analyzing the static features of network traffic patterns [64]. While both RNNs and CNNs offer substantial improvements over traditional ML models, their specific use cases differ. RNNs excel in sequential data modeling, while CNNs are more effective for tasks requiring spatial feature extraction. Ensemble models combining these architectures have been explored for further performance gains.

To demonstrate the application and performance of various machine learning and deep learning models in the context of intrusion detection systems (IDSs), Table 1 provides a comparative analysis of these models across different datasets with classification accuracy as a key metric.

Traditional ML models, such as SVM, DT, and RF, have been widely applied in earlier studies. For instance, reference [65] achieved a high accuracy of 98.9% using a K-means clustering model on the KDD Cup 99 dataset. However, as noted in Section 2.12, these models often struggle with underrepresented attack classes. DL approaches have been increasingly adopted due to their ability to handle complex data representations. Reference [66] applied a CNN on the KDD Cup 99 dataset, achieving an accuracy of 97.1%, while [67] employed an LSTM, enhancing the accuracy to 97.8%. These results highlight the effectiveness of CNNs and LSTMs in detecting network intrusions with higher accuracy compared to traditional ML models.

Table 1. Former IDS models and their corresponding outcomes.

Ref./Authors	Model	Dataset	Classification Accuracy (%)
[66] Zhang	CNN	KDD Cup 99	97.1
[68] Gupta	RNN	KDD Cup 99	96.4
[67] Mishra	LSTM	KDD Cup 99	97.8
[69] Wang	Random Forest	DARPA	82.6
[65] Ahmed Hasan	K-means	KDD Cup 99	98.9
[70] Raj Mukkamala	CNN + RNN + LSTM	DARPA	99.9
[71] Gupta	Random Forest + K-means	KDD Cup 99	98.8
[72] Kwon	GAN	NSL-KDD	92.3
[73] Binbusayyis	K-means	UNSW-NB15	95.6
[74] Alzahrani	CNN	CIC-IDS2017	97.2
[75] Wang	LSTM	CIC-IDS2018	98.5
[76] Zhang	Ensemble Learning	NSL-KDD	94.8

Ensemble models, which combine multiple machine learning algorithms or deep learning architectures, have demonstrated even greater performance. Reddy et al. [29] integrated CNN, RNN, and LSTM networks on the DARPA dataset, achieving the highest accuracy of 99%, underscoring the potential of hybrid models in intrusion detection. The development of an IDS model is heavily influenced by the insights gathered from a review of the current literature and prior sections. A critical takeaway is that DL techniques have consistently outperformed traditional ML approaches in handling high-dimensional data and improving classification accuracy in IDS applications. For instance, the paper by Yin et al. [77] demonstrated that DL models, particularly CNNs, are superior to traditional ML models such as SVM and decision trees (DTs) when it comes to processing complex datasets like NSL-KDD and KDDCup99.

One key advantage of DL over traditional ML lies in its ability to automatically extract features from raw data, eliminating the need for manual feature engineering, which is a limitation in shallow learning methods [78]. Furthermore, deep learning algorithms such as feed-forward DNNs and CNNs tend to outperform Recurrent Neural Networks (RNNs), including GRU and LSTM networks, in certain IDS applications. Although LSTMs excel in time-series data analysis, DNNs have been shown to be more effective in static intrusion detection tasks due to their simpler architecture and faster training times.

Recent research also suggests that ensemble learning models, which combine multiple ML or DL algorithms, can enhance prediction accuracy and reduce variability in IDS results. Ensemble models outperform individual models by leveraging the strengths of each algorithm to correct the weaknesses of others [79]. This finding highlights the growing importance of hybrid approaches in developing robust IDS solutions.

However, significant challenges remain in accurately detecting less frequent attack types, such as R2L and U2R attacks, due to their underrepresentation in common datasets like NSL-KDD [80]. As most IDS models achieve high accuracy in detecting DoS and probing attacks, further research is needed to improve the detection rates for R2L and U2R attacks to ensure more comprehensive IDS performance. The IoT is a technology that links networks with sensors and other gadgets using IP-based communications. It is gaining popularity among individuals with Internet access. Individuals institutionalized because of a handicap or sickness might obtain advantages from the IoT by using it for remote monitoring, timely intervention, and healthcare services. Sensors, actuators, radio frequency identification (RFID), and other IOT components can be integrated into people's bodies and objects. To illustrate, consider the following scenario: Caregivers can operate the equipment they have at their disposal more easily if they have the assistance of the available accessories. It is possible to read RFID patients or patient tags (including medical devices) and identify them using IOT applications, which are becoming increasingly popular. They can also be used to monitor and regulate the activities of other people.

Various approaches have been proposed to ensure that communication between nodes in the IOT network is secure and reliable, proposing a trust management mechanism that can be described as dynamic and flexible in how it is implemented and operated [81]. This topic has been extensively discussed by [82] Bao and Chen, who have written extensively about it in their respective publications.

The researchers developed a management framework for the layered IOT organized around the concept of services, which is built on the trustworthiness of its nodes as a foundation. When used with the IOT, which is composed of multiple layers, it is intended to be a multi-layered system. The developers' documentation refers to this framework as "service-oriented", which indicates that it is designed to provide services. In addition to the core, sensor, and application layers, several other layers and components contribute to the system's structure and function.

1.2. Motivation

The IoT is rapidly becoming an integral part of modern life with smart devices and sensors increasingly embedded in applications ranging from healthcare to transportation. The complexity and volume of IoT data present unique challenges for intrusion detection systems (IDSs), requiring robust, scalable, and interpretable machine learning solutions.

Traditionally, Recurrent Neural Networks (RNNs) and their variants, such as Long Short-Term Memory (LSTM) networks, are highly effective in dealing with sequential data patterns, including network traffic data. As acknowledged, RNNs and LSTMs have been extensively studied in recent intrusion detection research with many papers demonstrating their effectiveness in learning temporal dependencies. However, while these models excel in processing time-series data, our research focuses on leveraging the unique properties of CNNs for the specific nature of the data used in this work—namely, spectrogram images representing IoT network traffic.

The necessity for XIoT stems from the inability of existing IDSs to address critical challenges in IoT security. Traditional methods, including machine learning and some deep learning approaches, struggle to provide real-time detection, handle large-scale data, or offer interpretability. By transforming raw IoT traffic into spectrogram images, XIoT enables CNNs to extract nuanced spatial and temporal patterns, addressing these limitations effectively.

- **Spectrogram Data Representation:** In this research, network traffic data are transformed into spectrogram images, which capture both spatial and frequency information. CNNs are particularly effective at extracting spatial features and patterns from

- images, making them an ideal choice for analyzing these spectrograms. The capability of CNNs to capture local patterns across the image allows them to detect subtle, localized anomalies in network traffic, which are critical for identifying cyber threats.
- **Computational Efficiency and Scalability:** CNNs offer significant advantages in terms of computational efficiency, especially when processing large-scale data in real-time environments, which is crucial for IoT applications. RNNs and LSTMs, while powerful for sequential tasks, often suffer from higher computational costs and longer training times due to their sequential nature. In contrast, CNNs can process data in parallel, making them more scalable and efficient for the real-time analysis of vast IoT-generated traffic.
 - **Robustness to Input Variations:** IoT data are prone to a variety of attacks and perturbations, such as shifts in scale, rotation, and translation. CNNs are inherently robust to such variations due to their ability to learn hierarchical features through convolutional layers. This robustness is critical for ensuring reliable detection across diverse IoT environments, where data can be highly dynamic and variable.
 - **Explainability and Interpretability:** The ability to integrate explainable AI (XAI) mechanisms into CNNs provides a significant advantage in security-critical applications like IDS. Transparency in decision making is vital for building trust among stakeholders in IoT security. The architectural properties of CNNs make them well suited for incorporating interpretability techniques, allowing the model's predictions to be more easily understood and trusted by security analysts.

Thus, while acknowledging the effectiveness of RNNs and LSTMs for time-series data, CNNs were selected due to their superior performance in image-based analysis, scalability, and interpretability—key attributes that align with the requirements and goals of this research. This strategic choice enhances the practicality of the proposed IDS for real-time, scalable IoT security applications.

1.3. Problem Statement

With the rapid proliferation of Internet of Things (IoT) devices, the security of these networks has become a critical concern. Traditional intrusion detection systems are often ill suited to handle the unique challenges posed by IoT networks, such as their large scale, diverse device types, and evolving attack strategies. This paper proposes a novel deep learning-based approach, using explainable gradient-based Convolutional Neural Networks (EG-CNNs), to detect and classify IoT network attacks. Our aim is to develop a model that not only achieves high detection accuracy but also provides transparency in its decision-making process, which is crucial for cybersecurity professionals to understand and trust the model's outputs. This research contributes to the growing need for effective and interpretable IoT intrusion detection systems. The rapid proliferation of Internet of Things (IoT) devices has led to significant advancements across various sectors. However, it has also introduced substantial cybersecurity vulnerabilities. An alarming analysis revealed that 83% of interactions between IoT devices occur in plain text, and 41% of these interactions lack any form of secure communication, such as SSL. This widespread insecurity exposes IoT networks to cyberattacks, particularly wireless attacks, due to their interconnected nature. Consequently, these vulnerabilities result in frequent compromises of communication channels and component interfaces within large systems, leading to the propagation of failures across different locations.

Traditional security measures, such as access control and encryption, offer some protection but are insufficient. Many attacks exploit common vulnerabilities in IoT applications, often resulting from rushed development cycles. These attacks can significantly impact the reliability and availability of IoT services, especially in critical infrastructure, where IoT

applications are heavily relied upon. Moreover, existing detection and mitigation strategies lack the robustness required to counter these evolving threats effectively.

Given the critical nature of these vulnerabilities, there is a pressing need for advanced detection and mitigation mechanisms that not only provide high accuracy but also offer explainability to enhance trust and decision making. This research introduces XIoT, an explainable deep learning-based IoT attack detection model, to comprehensively address these cybersecurity challenges.

Despite advancements in Intrusion Detection Systems (IDSs), current solutions fall short in handling large-scale IoT traffic in real time, adapting to rapidly evolving attack patterns, and offering transparent decision making. Specific challenges include the following:

- Latency in high-speed networks: Current models are unable to process extensive datasets in optical networks efficiently.
- Limited interpretability: Analysts lack actionable insights from existing ‘black-box’ models.
- Inadequate adaptability: Many IDSs fail to generalize to novel or evolving attack scenarios. These gaps significantly compromise the security of IoT ecosystems, such as smart grids, healthcare systems, and industrial IoT, necessitating the development of a novel approach.

1.4. Objectives

- To address the limitations of existing Intrusion Detection Systems (IDSs) by developing a novel model that provides the real-time, scalable, and interpretable detection of IoT threats, particularly in high-speed optical network environments.
- To design and implement an Explainable AI model (XIoT) that integrates the sequential and spatial analysis of IoT spectrogram images, leveraging Convolutional Neural Networks (CNNs) for the efficient and accurate detection of complex attack patterns.
- To thoroughly evaluate the XIoT model across diverse benchmark datasets, including KDD CUP99, UNSW NB15, and Bot-IoT, ensuring its adaptability to varying IoT network scenarios and attack complexities.
- To compare the proposed model’s performance with existing machine learning (ML) and deep learning methods, highlighting its superiority in terms of accuracy, interpretability, and practical utility for cybersecurity analysts.
- To demonstrate the practical applicability of XIoT in protecting critical IoT applications, such as smart grids, healthcare IoT, and industrial IoT, by offering actionable insights and ensuring robust cybersecurity.

1.5. Significance

The significance of this research lies in its potential to address critical gaps in existing Intrusion Detection Systems (IDSs) by providing a solution tailored for the unique challenges of IoT networks, particularly in high-speed optical communication environments. Current IDSs often struggle with real-time processing, scalability, and interpretability—limitations that this study directly addresses through the development of the XIoT model.

By leveraging advanced ML and DL techniques, specifically spectrogram-based CNNs, this research offers an innovative approach to detecting and predicting IoT-specific cyber threats. The integration of Explainable AI (XAI) mechanisms ensures transparency and trust, enabling cybersecurity analysts to understand and act upon the model’s predictions effectively.

This proactive and interpretable approach is expected to significantly reduce the frequency and impact of cyber-attacks in critical IoT applications, such as smart grids,

healthcare systems, and industrial IoT environments. By enhancing IoT security, this research not only mitigates immediate threats but also fosters greater confidence in adopting IoT technologies, enabling their continued expansion and innovation across industries. The outcomes of this research aim to create a more secure, reliable, and resilient IoT ecosystem. This contributes to safeguarding essential infrastructure, protecting sensitive data, and supporting the ongoing technological advancements required for smart cities, connected healthcare, and other transformative IoT applications.

1.6. Contribution

The research study introduces a groundbreaking approach to IoT security through the development of the XIoT model, which stands out due to its integration of advanced DL techniques and explainable AI mechanisms. Unlike traditional IDSs that often lack transparency, the XIoT model leverages CNNs to analyze spectrogram images derived from IoT network traffic, capturing both spatial and sequential data. This dual-focus analysis facilitates a more nuanced detection of malicious activities, offering unprecedented accuracy. Moreover, the model's emphasis on interpretability marks a significant departure from existing methods, providing stakeholders with clear insights into the decision-making process behind each detection. By validating the XIoT model across diverse benchmark datasets, including KDD CUP99, UNSW NB15, and Bot-IoT, the study not only demonstrates the model's robustness and adaptability but also highlights its superior performance in real-world IoT environments. This novel combination of high accuracy, cross-dataset validation, and enhanced transparency represents a substantial advancement in the field of IoT cybersecurity, addressing both the technical and practical challenges of protecting IoT ecosystems from sophisticated cyber threats.

1. **Innovative Model Design:** The XIoT model is an innovative method for IoT intrusion detection that combines explainable AI with CNNs. This innovative model architecture offers enhanced interpretability, enabling deeper insights into the underlying features driving intrusion detection decisions in IoT networks.
2. **Cross-Dataset Validation:** Through rigorous evaluation across multiple datasets, including KDD CUP99, UNSW NB15, and Bot-IoT, the XIoT model demonstrates its robustness and generalizability. The model's efficacy is validated by showcasing consistent performance across diverse IoT network environments, instilling confidence in its practical applicability.
3. **Performance Superiority:** The XIoT model surpasses current methods in a comparative study by obtaining superior accuracy, precision, recall, and F1-score metrics across all datasets. The efficacy of the proposed approach in effectively identifying and categorizing intrusions highlights its superiority in strengthening the security of IoT networks.
4. **Enhanced Interpretability:** Leveraging explainable AI techniques within the XIoT model enhances its interpretability, allowing for transparent and comprehensible intrusion detection decisions. The XIoT model empowers cybersecurity analysts with actionable insights for timely response and mitigation by elucidating the contributing features behind detected threats.
5. **Practical Utility:** The XIoT model holds significant practical implications for bolstering IoT security measures in real-world scenarios. Its ability to effectively identify and mitigate intrusions in IoT networks can safeguard critical infrastructure, sensitive data, and connected devices against evolving cyber threats.

2. Methodology

The transformation of IoT traffic into spectrogram images allows the XIoT to utilize CNNs for detecting subtle, localized anomalies that are often missed by traditional methods. This innovative approach ensures the robust detection of diverse attack patterns in large-scale, dynamic IoT environments.

2.1. Data Processing

As soon as we have obtained a dataset, we must preprocess it to use it to train our ML models on the information it contains. To accomplish this, we followed a five-step procedure, such as data cleaning and conversion, splitting the data into training, test, and validation sets, and, finally, creating images from the data.

2.1.1. Data Preprocessing

Successfully analyzing the NUSW-NB15, KDD Cup 99, Bot-IoT, and TON_IoT datasets required meticulous preprocessing steps. Initially, entries with zero values were systematically removed from the datasets, necessitating their conversion into integer or floating-point representations before elimination. Additionally, six characteristics deemed non-informative for categorizing network attacks were excluded from all entries in the datasets as a precautionary measure. Certain features were also turned off by default due to specific circumstances to mitigate potential bias introduced during the model training phase. Although the immediate impact on model accuracy might have been minimal, this decision was anticipated to enhance long-term performance.

Consequently, specific pieces of information, such as IPv4 source and destination port numbers (about Internet Protocol version 4) and minimum and maximum flow Time-to-Live (TTL) values, were removed from the dataset columns. Following these preprocessing steps, each dataset comprised distinct features tailored to their respective classes. These steps were applied uniformly across all four datasets, ensuring consistency and reliability throughout the subsequent analysis and model training phases.

2.1.2. Data Resampling

We determined that the entire dataset was excessively large, prompting us to utilize only 100%. This approach allowed us to implement and test models within a reasonable timeframe. To ensure the equal distribution of data across all classes of attacks, we employed stratified sampling on the dataset. Subsequently, after converting the data into images as shown in Figure 1, we categorized and divided the dataset into three groups for further analysis.

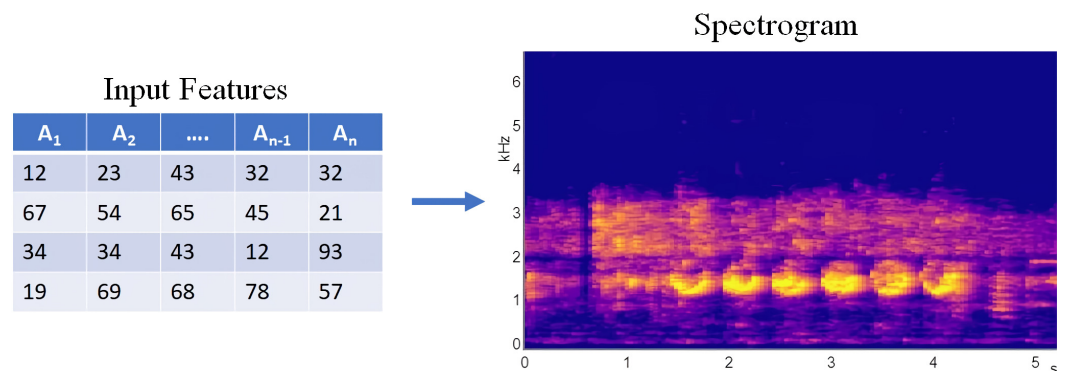


Figure 1. Image creation process using spectrogram.

We adopted the three-way holdout method with a 70–30 split and stratified sampling to ensure that each attack type was adequately represented in every dataset group. Following the study's conclusion, the collected samples comprised training, validation, and testing samples, totaling the data collected during the study's duration.

Due to the reduced concern regarding high variance in large datasets, the 3-way holdout method is commonly favored. However, for smaller datasets, the k-fold cross-validation method is typically preferred

2.1.3. Features and Spectrogram Computation

Features Adopted in the Proposed Model

In the proposed XIoT model, the primary features are the spectrogram images generated from raw IoT network traffic data. These spectrograms encapsulate both spatial and temporal characteristics of the network traffic, enabling the model to detect intricate patterns and anomalies indicative of potential cyber threats. The transformation of network traffic data into visual representations allows us to leverage the powerful pattern recognition capabilities of Convolutional Neural Networks (CNNs), facilitating the effective classification of malicious activities.

The use of spectrograms is particularly advantageous because they highlight variations in signal frequency over time, making it easier for the model to identify unusual patterns that may signify an attack. By analyzing these visual cues, the model can better differentiate between benign and malicious traffic, thereby improving the overall accuracy of intrusion detection in IoT environments.

Computing the Spectrogram

The process of computing the spectrogram involves several critical steps, as outlined below:

1. **Preprocessing Network Traffic Data:** The raw IoT network traffic data are first preprocessed into a suitable format for spectrogram generation. This preprocessing involves the following:
 - **Segmentation:** The data are divided into fixed time windows to capture the dynamic nature of network traffic. Each segment corresponds to a specific time frame during which traffic characteristics are analyzed.
 - **Normalization:** The values within each segment are normalized to ensure consistency across different data ranges. This step is crucial for reducing variance and ensuring that the model training is stable and effective.
2. **Transformation to Time–Frequency Domain:** Each segmented portion of the network traffic data are transformed into the time–frequency domain using the Short-Time Fourier Transform (STFT). This technique is selected for its effectiveness in analyzing non-stationary signals, which are common in network traffic. The mathematical representation of STFT is given by

$$X(t, f) = \int_{-\infty}^{\infty} x(\tau)w(\tau - t)e^{-j2\pi f\tau} d\tau \quad (1)$$

where $w(\tau - t)$ is a window function that slides along the time axis t . This function helps to localize the signal in both time and frequency, providing a comprehensive view of the traffic dynamics.

3. **Generating the Spectrogram Image:** Once the STFT is computed, the magnitude of the resulting complex numbers is obtained. This magnitude represents the intensity of different frequencies over time. The spectrogram is visualized as a 2D image, where the following apply:

- One axis represents time.
- The other axis represents frequency.
- The color intensity reflects the magnitude of the signal at each time–frequency point with brighter colors indicating higher signal strength.

This visual representation allows the model to capture intricate details that may be indicative of specific types of network attacks.

4. **Normalization and Scaling:** The final step involves normalizing and resizing the spectrogram images to ensure compatibility with the input requirements of the CNN. This typically includes the following:
 - Scaling pixel values to a standard range, such as $[0, 1]$, which helps in stabilizing the training process and improving convergence rates.
 - Resizing the images to a consistent dimension to ensure that all input data fed into the CNN are uniform, thereby simplifying the model architecture and training process.

By following these steps, we generate a robust set of features that are well suited for the detection of IoT-related attacks, leveraging the unique capabilities of CNNs to analyze visual data effectively.

2.2. Metrics Justification

To evaluate the performance of the proposed model, we use standard classification metrics: accuracy, precision, recall, and F1-score. These metrics were chosen because they provide a comprehensive view of the model's performance, especially in the context of IoT intrusion detection. Accuracy gives an overall measure of correct predictions, while precision and recall help assess the model's ability to identify true positive attacks (precision) and minimize false negatives (recall). Since IoT attack datasets are often imbalanced, where benign traffic dominates, the F1-score is used as a balanced metric to assess the trade-off between precision and recall. These metrics are crucial for evaluating how well the model performs in real-world scenarios where minimizing both false positives (incorrectly labeling benign traffic as an attack) and false negatives (missing an actual attack) is essential.

2.3. EG-CNN Model Architecture

The architecture of the proposed model for IoT attack detection, known as Explainable Gradient CNN (EG-CNN), is illustrated in Figure 2. This architecture comprises multiple layers: convolutional, pooling, fully connected, and output layers, each meticulously designed to extract and analyze pertinent features from incoming spectrogram images. This structured approach facilitates the precise categorization of various IoT threats, enhancing the model's effectiveness in detecting malicious activities.

While the proposed model employs a general CNN architecture, several key modifications enhance its applicability for IoT attack detection:

Spectrogram Input: The model uniquely utilizes spectrogram images derived from IoT network traffic data, which captures both temporal and frequency characteristics, allowing for more nuanced feature extraction compared to raw data inputs.

Layer Configuration: The architecture includes multiple convolutional layers with varying filter sizes that are optimized for detecting specific patterns relevant to IoT attacks shown in Table 2. Each layer's configuration is tailored to improve feature extraction progressively.

Pooling Strategy: The use of max pooling after convolutional layers reduces dimensionality while preserving critical features, which are essential for accurately classifying complex IoT traffic patterns.

Dropout Layer: A dropout rate of 0.5 is implemented to combat overfitting, which is crucial given the diverse nature of IoT network data.

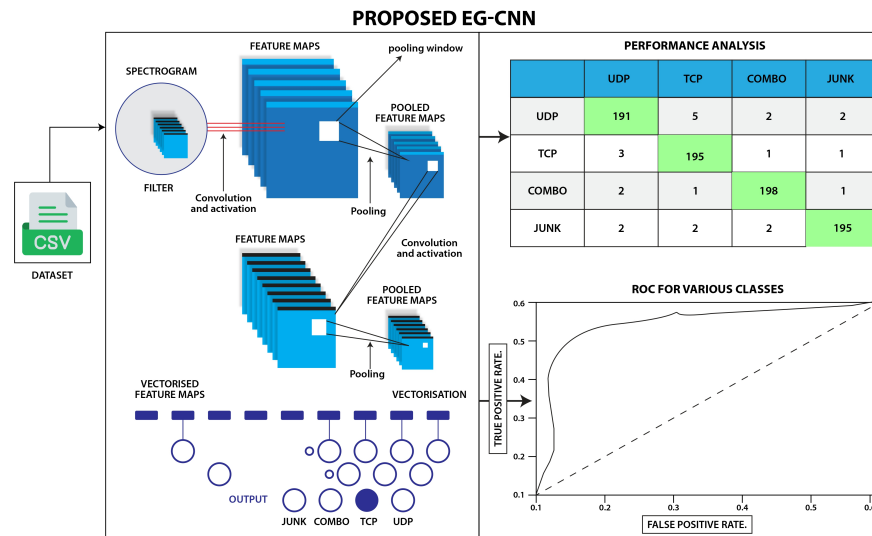


Figure 2. Proposed deep learning framework for IoT intrusion detection.

Table 2. Proposed XIoT model layer details.

Layer Type	Number of Filters	Kernel Size	Pool Size	Neurons/Output Classes
Input	-	-	-	Spectrogram Images
Convolutional	32	3 × 3	-	-
Convolutional	64	3 × 3	-	-
Convolutional	64	3 × 3	-	-
Convolutional	64	3 × 3	-	-
Max Pooling	-	-	2 × 2	-
Fully Connected	-	-	-	256
Fully Connected	-	-	-	256
Dropout	-	-	-	-
Output (Softmax)	-	-	-	4 (IoT Attack Classes)

Softmax Output: The final softmax layer facilitates multi-class classification, which is specifically tailored to recognize four distinct classes of IoT attacks.

The convolutional layers are essential for capturing spatial characteristics from the input spectrograms. Each layer employs filters of increasing complexity, allowing for the extraction of hierarchical features. The initial layers focus on detecting basic patterns, while deeper layers identify more abstract features indicative of sophisticated attack strategies. This progressive feature extraction enhances the model’s ability to recognize a wide variety of attack types.

The max pooling layer follows the convolutional layers, significantly reducing the spatial dimensions of the feature maps. This reduction not only decreases the computational load but also emphasizes the most critical features, which are pivotal for the classification task. By retaining the most salient aspects of the data, the model becomes more resilient to minor variations and noise present in the input data.

The fully connected layers are positioned at the end of the architecture, performing the final classification based on the features learned from the preceding convolutional layers. Each fully connected layer comprises 256 neurons, allowing the model to learn

complex decision boundaries essential for accurately distinguishing between different types of IoT attacks.

To mitigate the risk of overfitting, a dropout layer with a rate of 0.5 is incorporated. This technique randomly disables a fraction of neurons during training, promoting robust feature learning and improving the model's generalization capabilities on unseen data.

Finally, the output layer employs a softmax activation function to generate a probability distribution across the four classes representing various IoT threats. This function ensures that the predicted probabilities sum to one, facilitating the interpretation of the model's predictions as confidence scores for each class. This structured and comprehensive architecture thus plays a crucial role in the effective detection and classification of IoT-based attacks.

2.4. Explainable AI

Explainable AI (XAI) refers to methods and techniques in artificial intelligence that make the outcomes of complex models understandable to humans. In the context of cybersecurity, particularly for IoT systems, XAI is vital as it provides insights into the decision-making processes of models, fostering trust and allowing cybersecurity professionals to interpret and validate predictions.

2.4.1. Importance of Explainability

In IoT environments, where devices continuously generate massive amounts of data, having models that can explain their decisions is crucial. Cybersecurity professionals need to understand the rationale behind an AI system's predictions to respond effectively to potential threats. XAI empowers these professionals by providing the following:

1. **Transparency:** Clear insights into how decisions are made, allowing users to follow the reasoning of the model.
2. **Trust:** By understanding the model's decision-making process, stakeholders can have greater confidence in its outputs, which is essential when dealing with security threats.
3. **Improved Decision Making:** With insights into feature importance, analysts can prioritize response strategies based on the characteristics driving model predictions.

2.4.2. Techniques Employed in the XIoT Model

The proposed Explainable Gradient CNN (EG-CNN) model leverages several techniques to enhance its interpretability:

1. **Gradient Visualization:** The model incorporates gradient-based methods to visualize critical regions in input spectrogram images. By analyzing the gradients of the model's output concerning input features, we can identify which parts of the spectrogram contribute most significantly to the model's predictions. This helps in pinpointing potential attack patterns.
2. **Feature Importance Scores:** The model provides quantifiable importance scores for various features derived from the spectrograms, as shown in Table 3. These scores help users understand which attributes are most influential in the detection of specific IoT attacks.
3. **Potential Techniques for Further Explainability:** Future iterations of this research may incorporate additional XAI methods, such as the following:
Gradient-weighted Class Activation Mapping (Grad-CAM): This technique visualizes the areas in the spectrogram that influence predictions, aiding analysts in interpreting results more intuitively. **SHAP (SHapley Additive exPlanations):** SHAP values can be utilized to explain the contribution of each feature to the model's predictions, offering a comprehensive view of feature interactions.

To illustrate the practical utility of the EG-CNN's explainability, consider a scenario where the model identifies a surge in suspicious network traffic labeled as a potential DDoS attack. By analyzing the gradient visualizations, cybersecurity analysts discover that specific frequency patterns in the spectrogram correlate with previous DDoS events. This insight not only confirms the model's prediction but also allows for a more targeted and effective response.

2.5. Dataset Selection Justification

The datasets used in this study—NUSW-NB15, KDD Cup 99, Bot-IoT, and TON_IoT—are selected based on their relevance and comprehensiveness in representing real-world IoT traffic and attack patterns. These datasets include a wide variety of attacks such as Denial of Service (DoS), Distributed Denial of Service (DDoS), botnets, and others, which are typical threats faced by IoT networks. For example, Bot-IoT is designed specifically to simulate botnet attacks in IoT environments, while NUSW-NB15 covers various IoT devices and attack types. The diversity of these datasets ensures that the proposed model is exposed to a broad range of attack scenarios, which enhances its generalization and robustness. Furthermore, these datasets are widely used in the literature, allowing for meaningful comparisons with state-of-the-art intrusion detection methods.

Training Process

The EG-CNN model undergoes training utilizing Stochastic Gradient Descent with Momentum to optimize the weights and biases for predictive accuracy. The training process encompasses several stages:

2.6. Data Preparation

Spectrogram images representing IoT network traffic data are preprocessed and augmented to enhance model generalization. Data augmentation techniques such as rotation, scaling, and flipping are applied to increase the diversity of the training dataset.

2.7. Model Training

The preprocessed spectrogram images \mathbf{X} are fed into the EG-CNN model, where θ denotes the model's parameters (weights and biases). The training process iteratively updates θ using mini-batch Stochastic Gradient Descent.

During forward propagation, the output $Z^{[l]}$ of layer l is calculated as the linear combination of the activations from the previous layer, $A^{[l-1]}$, using the layer's weights $W^{[l]}$ and biases $b^{[l]}$, which are followed by the application of an activation function $g^{[l]}$:

$$A^{[l]} = g^{[l]}(W^{[l]}A^{[l-1]} + b^{[l]}) \quad (2)$$

The loss function $J(\theta)$ measures the difference between the predicted outputs \hat{y} and the true outputs y . For multi-class classification problems, the categorical cross-entropy loss function is used:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m L(\hat{y}^{(i)}, y^{(i)}) \quad (3)$$

At each iteration t , the model parameters θ are updated using the gradient descent update rule:

$$\theta_{t+1} = \theta_t - \alpha \nabla_{\theta_t} J(\theta_t) \quad (4)$$

where the following apply:

- α is the learning rate.

- $J(\theta_t)$ is the loss function, typically categorical cross-entropy, which is minimized to enhance the model's predictive accuracy.
- $\nabla_{\theta_t} J(\theta_t)$ represents the gradient of the loss function with respect to the model parameters at iteration t .

The training process continues for multiple epochs until convergence or until a stopping criterion is met.

2.7.1. Features and Interpretability

The EG-CNN model integrates explainable gradient-based approaches to elucidate essential features contributing to IoT attack diagnosis. Critical regions indicative of attack patterns can be discerned by visualizing the gradients of the model's output concerning the input spectrogram images. This interpretability empowers cybersecurity analysts to understand the underlying mechanisms driving the model's predictions.

Figure 3 illustrates the visualization of gradient magnitudes generated by the EG-CNN model. Higher gradient magnitudes signify regions of the spectrogram images that significantly influence the model's predictions. By analyzing these gradients, cybersecurity analysts can pinpoint crucial features associated with different types of IoT attacks, thereby enhancing the interpretability of the EG-CNN model.

Table 3 presents the top important features extracted from the input spectrogram images by the EG-CNN model. These features and their corresponding importance scores provide valuable insights into the characteristics of IoT attacks captured by the model. By leveraging this feature interpretability, cybersecurity analysts can better understand the discriminative power of the EG-CNN model and refine their threat detection strategies accordingly.

Table 3. Top important features.

Feature	Importance Score
Frequency Band	0.87
Time Duration	0.76
Spectral Density	0.68
Frequency Shift	0.62
Amplitude	0.58

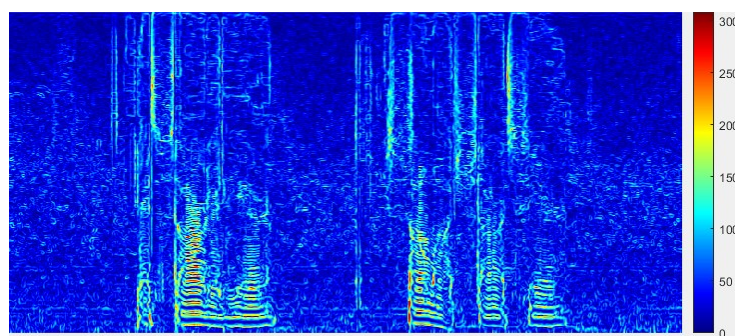


Figure 3. Transforming IoT data to a spectrogram.

2.7.2. Mathematical Equations

The training process of the EG-CNN model involves the forward propagation of input data through the network and the computation of loss functions. The equations governing these processes are described below.

2.8. Forward Propagation

The forward propagation process computes the activations of each layer in the neural network. The output $Z^{[l]}$ of layer l is calculated as the linear combination of the activations from the previous layer, $A^{[l-1]}$, using the layer's weights $W^{[l]}$ and the biases $b^{[l]}$, followed by an activation function $g^{[l]}$.

$$Z^{[l]} = W^{[l]}A^{[l-1]} + b^{[l]} \quad (5)$$

$$A^{[l]} = g^{[l]}(Z^{[l]}) \quad (6)$$

where the following apply:

- $Z^{[l]}$ represents the linear output of layer l .
- $A^{[l]}$ is the activation of layer l .
- $W^{[l]}$ and $b^{[l]}$ are the weights and biases of layer l .
- $g^{[l]}$ is the activation function.

2.9. Loss Function

The loss function measures the difference between the predicted outputs \hat{y} and the true outputs y of the model. The categorical cross-entropy loss function is commonly used for multi-class classification problems:

$$J(W, b) = \frac{1}{m} \sum_{i=1}^m L(\hat{y}^{(i)}, y^{(i)}) \quad (7)$$

where the following apply:

- m is the number of samples.
- $L(\hat{y}^{(i)}, y^{(i)})$ is the categorical cross-entropy loss for the i -th sample.

The forward propagation equations compute the activations of each layer in the EG-CNN model, while the loss function measures the model's performance by comparing its predictions with the ground truth labels. The equations are fundamental to the training process, allowing the model to enhance its performance by learning over time.

2.10. Adam Optimization Algorithm

In addition to the forward propagation and loss function equations described earlier, the EG-CNN model's training process involves utilizing the Adam optimization algorithm and the softmax activation function. Incorporating these elements enhances the efficiency of the training process and facilitates the interpretation of model predictions. During training, the Adam optimization algorithm updates the model parameters (weights and biases). It adapts the learning rate for each parameter based on past and squared gradients, providing faster convergence and better performance. The parameter updates in Adam are computed as follows:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla_{\theta_t} J(\theta_t) \quad (8)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) (\nabla_{\theta_t} J(\theta_t))^2 \quad (9)$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (10)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (11)$$

$$\theta_{t+1} = \theta_t - \frac{\alpha \hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}} \quad (12)$$

where the following apply:

- β_1 and β_2 are the exponential decay rates for the first and second moments, typically set to 0.9 and 0.999, respectively.
- α is the learning rate.
- ϵ is a small constant to prevent division by zero.
- m_t and v_t are the first and second-moment estimates of the gradients.
- \hat{m}_t and \hat{v}_t are bias-corrected estimates of the first and second moments.

2.11. Softmax Activation

The EG-CNN model uses the softmax activation function on the output layer to calculate the probability distribution across different classes. It guarantees that the sum of the outputs equals one, allowing them to be interpreted as probabilities. The softmax function is formally defined as

$$\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^C e^{z_j}} \quad (13)$$

where the following apply:

- \hat{y}_i is the predicted probability for class i .
- z_i is the raw output for class i .
- C is the number of classes.

Incorporating Adam optimization and softmax activation into the training process enhances the efficiency and interpretability of the EG-CNN model, enabling it to learn from the data and make accurate predictions effectively.

2.12. Cross-Validation in the Proposed Model

We employed a hold-out cross-validation scheme with a 70/30% split to ensure the robustness and generalizability of the XIoT model. In this approach, 70% of the dataset is used for training the model, while the remaining 30% is reserved for testing its performance. This method provides an efficient way to evaluate model performance, especially when dealing with large datasets. The use of cross-validation ensures that the model's ability to generalize to unseen data is effectively assessed.

The primary goals of using cross-validation in the XIoT model are outlined below:

1. **Evaluate Model Performance:** Metrics such as accuracy, precision, recall, and F1-score are computed on the test set to measure how well the model generalizes to new, unseen data.
2. **Detect Overfitting:** Cross-validation helps to ensure that the model does not simply memorize the training data but learns to generalize patterns that apply to new data. This helps mitigate the risk of overfitting.
3. **Optimize Hyperparameters:** Performance metrics obtained from the validation set during cross-validation are used to fine-tune the model's hyperparameters, ensuring the highest accuracy and robustness.

By employing this strategy, the model's performance is assessed across multiple subsets of data, ensuring that it can reliably detect IoT attacks in a variety of scenarios without overfitting or underperforming.

2.13. Cross-Validation and Validation Strategy

To ensure the generalizability and robustness of the model, we employ a 70/30% hold-out cross-validation scheme. This means that 70% of the dataset is used for training and 30% is used for testing the model’s performance. This approach mitigates the risk of overfitting by providing a separate validation set that is not seen by the model during training. Additionally, the performance metrics (accuracy, precision, recall, and F1-score) are computed on the test set to evaluate the model’s generalization capabilities. Cross-validation allows us to assess how well the model would perform on unseen data, simulating real-world conditions where IoT attack patterns may differ from those seen during training. The use of multiple benchmark datasets further enhances the reliability of the model’s performance evaluation, ensuring it can handle a wide range of IoT attack scenarios.

2.14. Training

This section of the document details how we chose the hyperparameters to test and the procedures we used to conduct the tests. Additionally, this section discusses the various strategies we used to overcome the difficulty of training a classifier on a skewed dataset. These strategies will be discussed in greater detail later in this section. Additionally, it contains information about the hardware used during the testing period, which is beneficial for those interested in learning more about the specific hardware used during the testing period.

2.14.1. Hyperparameters of the XIoT Model

In this section, we provide a detailed explanation of the hyperparameters used in the XIoT model as shown in Table 4. These hyperparameters are crucial for tuning the performance of the CNN and ensuring the optimal detection of IoT attacks.

Table 4. Hyperparameters and their descriptions.

Hyperparameter	Description	Value(s) Used
Learning Rate	The step size used by the optimizer to update the model parameters during training. A lower learning rate can lead to more precise convergence.	0.001
Batch Size	The number of training samples used in one iteration to update the model weights. A larger batch size can lead to faster training but requires more memory.	32, 64
Optimizer	The algorithm used to minimize the loss function and update model parameters.	Adam
Activation Function	The function applied to the output of each neuron to introduce non-linearity.	ReLU, Softmax
Filter Size	The size of the convolutional filters (kernels) used in each convolutional layer.	3×3 , 5×5
Number of Filters	The number of filters used in each convolutional layer, which determines the depth of the output feature map.	32, 64, 128
Pooling Size	The size of the pooling window used in max-pooling layers to downsample the feature maps.	2×2
Dropout Rate	The fraction of neurons randomly set to zero during training to prevent overfitting.	0.3, 0.5
Loss Function	The function used to measure the difference between the predicted and actual labels during training.	Categorical Cross-Entropy
Validation Split	The proportion of the training data used for validation during training to monitor the model’s performance.	0.2
Early Stopping Patience	The number of epochs with no improvement after which training is stopped to prevent overfitting.	10

These hyperparameters are fine-tuned based on the performance of the model on the validation dataset. Through experimentation and cross-validation, the optimal values are determined to ensure the highest accuracy and robustness in detecting IoT attacks. By adjusting these hyperparameters, the XIoT model can be tailored to handle different complexities and variations in IoT network traffic data, achieving superior performance across multiple datasets.

2.14.2. Hardware

The training session needed to be conducted entirely on an Intel workstation, which was used for every operation step, to ensure everything went properly. When running this application, a powerful Nvidia GeForce RTX 3090 8GB GPU and 48 GB of DDR4 RAM are required, as well as an Intel® Xeon® Processor E5-2697 v2 CPU working at 2.70 GHz, for it to function effectively.

To make use of deep learning frameworks, it is necessary to make use of the CUDA library, which is made feasible by an Intel® Xeon® Processor E5-2697 v2 processor operating at a frequency of 2.70 GHz. The Intel® Xeon® Processor E5-2697 v2 is the central processing unit for this particular computer, and it is responsible for serving as the system's central processing unit.

All of the simulations in this study are carried out using the Windows 8.1 operating system.

3. Experimental Result

3.1. Dataset

The efficacy of the XIoT model is rigorously evaluated using a variety of benchmark datasets, each presenting unique challenges and complexities inherent in IoT security. These datasets are meticulously chosen to ensure a comprehensive assessment of the model performance across diverse attack scenarios and network environments. The selected datasets are widely recognized in the field of cybersecurity for their relevance and extensive use in research, making them ideal for benchmarking IoT attack detection models. The datasets used in this evaluation include the KDD Cup 99, Bot-IoT, and UNSW-NB15 datasets. Each dataset encompasses a rich variety of attack patterns and normal traffic, providing a robust framework for testing the adaptability and accuracy of the XIoT model. By leveraging these datasets, the evaluation aims to simulate real-world conditions, thereby ensuring that the XIoT model is well equipped to handle the dynamic and evolving nature of cyber threats in IoT networks. The diversity in the datasets, ranging from traditional network intrusions to sophisticated IoT-specific attacks, highlights the comprehensive nature of the evaluation process. This rigorous testing not only demonstrates the resilience and effectiveness of the XIoT model but also underscores its potential applicability in enhancing the security of IoT ecosystems.

3.1.1. KDD Cup 99 Dataset

The KDD Cup 99 dataset is a benchmark for studying IDSs. The project was developed for a knowledge discovery and data mining (KDD) competition emphasizing network security. The dataset contains simulated network traffic statistics collected from a U.S. Air Force local area network (LAN) across many weeks 1998. The dataset aims to train and assess ML algorithms for intrusion detection, serving as a benchmark for comparing different IDS approaches. The dataset is provided in a tab-delimited text format with each line representing a single network connection record. Each record consists of 41 features and a class label.

The KDD Cup 99 dataset is a pivotal resource for developing and benchmarking intrusion detection systems, originating from the Third International Knowledge Discovery and Data Mining Tools Competition in 1999. This dataset, derived from DARPA’s 1998 Intrusion Detection Evaluation Program, simulates a network environment containing both normal traffic and various types of attacks, making it ideal for testing machine learning algorithms designed to identify malicious activities.

The dataset comprises 41 features for each network connection record, which are categorized into Basic Features, Content Features, Traffic Features, and the Class Label. Basic Features include essential details such as source and destination IP addresses, protocol type, service used, and flag bits, with an example being a TCP connection between ‘192.168.1.1’ and ‘10.0.0.1’ using the ‘http’ service and the ‘SYN’ flag Table 5. Content Features describe the specifics of the data transferred within the connection, like the number of bytes sent and received and the number of connections, with typical values being ‘100’ bytes sent, ‘200’ bytes received, and ‘1’ connection. Traffic Features capture the temporal aspects of the connections, including arrival times and durations, for instance, an arrival time of ‘1000’ milliseconds and a duration of ‘50’ seconds. The Class Label categorizes each connection as either “normal” or one of several attack types: Denial of Service (DoS), User to Root (U2R), Remote to Local (R2L), and Probe.

A detailed breakdown of the dataset’s class distribution reveals significant imbalances. The largest class is DoS, with a total of 391,458 samples, which are subdivided into 274,020 for training and 117,437 for validation. This is followed by the Probe class, comprising 41,072 samples (28,750 for training and 12,321 for validation). The Normal class has 21,528 samples, which are split into 15,069 for training and 6458 for validation. The R2L class has 803 samples with 562 for training and 240 for validation. The smallest class is U2R, containing just 52 samples, with 36 for training and 15 for validation.

Despite its widespread use, the KDD Cup 99 dataset has certain limitations, such as redundancy due to a significant number of duplicate records and the inclusion of outdated attacks that may not reflect current network threats accurately. Moreover, as it is based on a simulated environment, it might not capture the full complexity of real-world network traffic. Nevertheless, the dataset remains a valuable resource for researchers and practitioners, providing a rich set of features and a variety of attack types to develop and evaluate intrusion detection systems. The detailed feature descriptions and class distribution help in understanding the dataset’s structure and the challenges associated with it, particularly the class imbalance, which is crucial for developing effective detection models.

Table 5. KDD Cup 99 dataset feature descriptions: type, explanation, and example.

Feature Type	Description	Example
Basic Features	Source and destination IP addresses, protocol type, service used, flag bits	192.168.1.1, 10.0.0.1, TCP, http, SYN
Content Features	Bytes in various directions, number of connections	100, 200, 1
Traffic Features	Time features like arrival time, duration	1000, 50
Class Label	Categorizes the connection as “normal” or an attack type	normal, DoS, U2R, R2L, probe

In the context of evaluating advanced models like XIoT, which is designed to address changing cyber risks in IoT networks using deep learning methods such as Convolutional Neural Networks (CNNs), the KDD Cup 99 dataset serves as a crucial benchmark shown in Table 6. XIoT leverages these datasets to validate its capability to accurately classify diverse IoT attacks and explain the key factors behind its decision-making process. This

emphasis on interpretability and transparency enhances trust in the model's outputs and supports informed decision making by cybersecurity analysts and network administrators. Comprehensive experiments on benchmark datasets, including KDD Cup 99, showcase XIoT's exceptional accuracy rates and its superiority over current intrusion detection methods in both accuracy and interpretability.

Table 6. KDD Cup 99 dataset class distribution (70%/30% train/validation split).

Class Name	Total Samples	Training Samples	Validation Samples
normal	21,528	15,069	6458
dos	391,458	274,020	117,437
u2r	52	36	15
r2l	803	562	240
probe	41,072	28,750	12,321

3.1.2. Botnet-Iot Dataset

The Bot-IoT dataset helps study botnet detection in the IoT field. The dataset contains network traffic statistics categorized as normal or botnet-related. The collection consists of network traffic collected from several honeypots placed worldwide. Honeypots are imitation systems that lure and trick attackers by imitating genuine IoT devices. The dataset is designed for training and assessing ML models for detecting botnets in IoT networks. It can promote the advancement of secure communication protocols for IoT devices via research and development. It also provides insights into the behavior and characteristics of botnet attacks targeting IoT devices. The dataset is provided in different formats depending on the specific version, i.e., CSV, PCAP, and ELF. The specific features included in the CSV format may vary depending on the dataset version and collection methods. Some common features include flow-level features, statistical features, and time-based features, as shown in Table 7.

Table 7. BoT-IoT dataset class distribution (70%/30% train/validation split).

Class	Testing Dataset (30%)	Training Dataset (70%)
Normal	166,857	389,075
ack	193,665	450,156
combo	154,744	360,412
junk	78,672	183,117
scan	237,678	555,412
syn	219,894	513,405
tcp	257,510	602,340
udp	653,621	1,522,744
plain	156,140	367,164

3.1.3. UNSW-NB15

The UNSW-NB15 dataset consists of unprocessed network packets created by the IXIA PerfectStorm tool at the Cyber Range Lab at UNSW Canberra. This dataset combines current normal activities and artificially created contemporary assault behaviors, offering a varied and accurate representation of network traffic. Using the tcpdump program, 100 gigabytes of unprocessed traffic data were collected and saved in Pcap files. The dataset includes nine different kinds of attacks: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. Argus and Bro-IDS tools expanded the dataset, creating

twelve algorithms with 49 characteristics and class labels. The characteristics are outlined in the UNSW-NB15_features.csv file. The dataset comprises 2,540,044 entries divided into four CSV files: UNSW-NB15_1.csv, UNSW-NB15_2.csv, UNSW-NB15_3.csv, and UNSW-NB15_4.csv. The ground truth information is available in the UNSW-NB15_GT.csv file with labels for each entry. The file UNSW-NB15_LIST_EVENTS.csv contains a record of events, as shown in Table 8.

Table 8. UNSW-NB15 dataset class distribution (70%/30% train/validation split).

Class Name	Total Samples	Training Samples (70%)	Validation Samples (30%)
Normal	98,527	68,968	29,559
DoS	4918	3442	1476
U2R	125	87	38
Backdoor	2000	1400	600
Exploits	34	23	11
Fuzzers	2395	1676	719
Generic	210	147	63
Analysis	21	14	7
Shellcode	167	116	51
Worms	2939	2057	882

3.2. Bot-IoT Results

The training plot in Figure 4 showcases the performance of our model throughout the training process, providing valuable insights into its convergence and generalization capabilities. With a meticulous approach, we meticulously monitored the model's training and validation accuracy across 50 epochs, each comprising mini-batches of 32 samples. The training accuracy steadily climbed to an impressive 99.97%, reflecting the model's proficiency in learning from the training data and accurately classifying instances.

The validation accuracy achieved an impressive level of 99.21%, indicating the model's capacity to generalize well to new data. The high similarity between training and validation accuracy indicates that our model successfully grasped the inherent patterns in the data without overfitting.

Our choice of the Stochastic Gradient Descent with Momentum (SGDM) optimizer was crucial for optimizing the model's weights and biases, facilitating efficient convergence toward the optimal solution. The SGDM optimizer incorporates momentum to accelerate the learning process, enabling the model to navigate the complex parameter space more effectively.

In our study, we employed the Botnet-IoT dataset, consisting of a diverse range of network traffic data labeled normal or botnet-related, encompassing various attack types commonly encountered in the IoT networks. The dataset was divided into a training set with 70% samples and a testing set with the remaining 30%.

Our proposed model, trained on the Botnet-IoT dataset shown in Figure 5, demonstrated exceptional performance across different attack classes, as evidenced by the following performance metrics in Table 9.



Figure 4. Training plot of XIoT in Bot-IoT dataset.

		Actual Class								
		Normal	Ack	Combo	Junk	Scan	Syn	Tcp	Udp	Udp Plain
Predicted Class	Normal	550205	715	725	705	700	730	695	735	716
	ack	717	638090	698	721	707	729	701	714	720
	combo	231	210	513301	241	200	261	205	221	214
	junk	284	300	268	214512	290	278	250	313	285
	scan	1536	1500	1572	1608	780797	1501	1570	1520	1552
	syn	959	1010	910	900	1018	725672	1060	850	960
	tcp	1514	1528	1500	1400	1656	1556	847468	1462	1510
	udp	463	523	400	470	456	390	536	2172656	462
	udpplain	386	399	372	432	365	350	486	300	520216

Figure 5. Proposed XIoT confusion using the Bot-IoT dataset.

Table 9. Detailed performance by class for Bot-IoT dataset.

Class	Accuracy	Precision	Recall	F1-Score
Normal	99.22	98.97	98.91	98.94
Ack	99.22	99.11	99.04	99.08
Combo	99.22	99.17	98.76	99.33
Junk	99.13	98.95	97.07	98.00
Scan	99.24	98.44	99.31	99.88
Syn	99.25	98.95	99.21	99.08
Tcp	99.20	98.59	99.35	98.97
Udp	99.23	99.83	99.72	99.78
Udpplain	99.21	99.41	98.78	99.09
Average	99.21	99.04	98.90	99.12

The measurements show how well the model can correctly categorize botnet-related activity instances while reducing FP and negatives. The proposed model has great accuracy and precision in identifying and mitigating security risks in IoT networks, as shown by its performance across several attack classes. By achieving robust performance on the Botnet-IoT dataset, our model contributes to enhancing the security posture of IoT infrastructures, safeguarding against potential cyber threats and vulnerabilities.

We used Receiver Operating Characteristic (ROC) curves in Figure 6 to assess the effectiveness of our proposed model on various attack types in the Botnet-IoT dataset.

The ROC curve visually displays the balance between true positive rate (TPR) and FPR across different categorization criteria. Upon examination of the ROC curve, we observed that the class “Junk” achieved the highest area under the curve (AUC) value of 0.98, indicating excellent discriminative capability and model performance for this particular attack class. Conversely, the class “TCP” exhibited the lowest AUC value of 0.90, suggesting comparatively weaker performance distinguishing TP from FP.

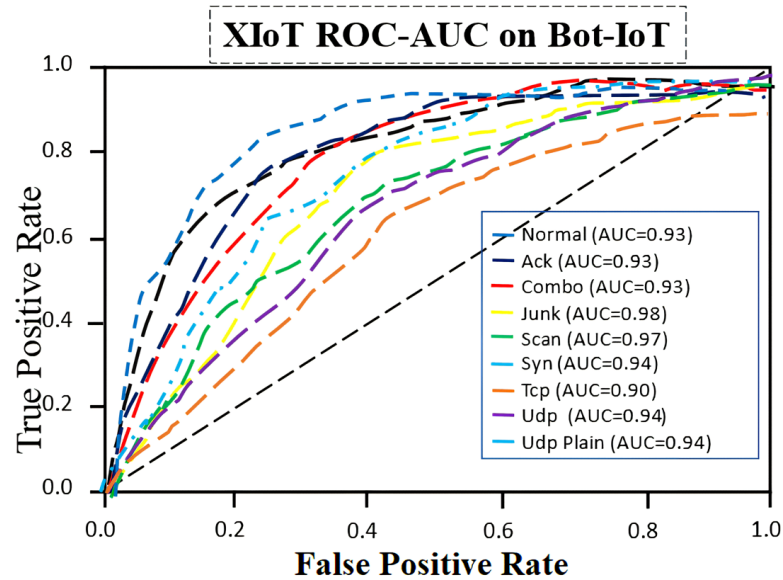


Figure 6. Proposed XIoT performance on Bot-IoT dataset.

3.3. UNSW-NB15

The performance assessment of our model shown in Figure 7 on the UNSW-NB15 dataset shows its ability to effectively categorize different forms of network traffic into their corresponding categories.

Predicted Class	Actual Class									
	Normal	DOS	U2R	Backdoor	Exploits	Fuzzers	Generic	Analysis	Shellcode	Worms
Normal	97196	147	153	144	137	157	117	177	139	155
DOS	3	4890	4	3	5	2	1	4	5	1
U2R	2	4	1226	3	2	1	0	2	0	3
Backdoor	13	16	10	1982	7	20	5	9	1	8
Exploits	1	0	1	0	336	0	1	0	1	0
Fuzzers	4	2	3	1	5	2364	6	0	7	6
Generic	0	0	0	0	3	2	406	0	0	1
Analysis	0	1	0	0	1	0	0	536	1	0
Shellcode	0	1	0	0	0	0	0	1	300	0
Worms	1	0	0	0	0	1	0	0	0	288

Figure 7. Proposed XIoT confusion using the UNSW NB15 dataset.

Across different attack classes, our model achieved high accuracy with an average accuracy of 98.61% shown in Table 10. Notably, the model exhibited particularly strong performance in classifying normal network traffic, achieving an accuracy of 98.69%. This indicates the model’s ability to distinguish normal network behavior from malicious activities effectively.

Table 10. Detail performance by class for UNSW-NB15 dataset.

Class	Accuracy	Precision	Recall	F1-Score
Normal	98.69	98.71	99.98	99.34
DoS	98.34	99.43	96.62	98.01
U2R	98.91	98.63	87.76	92.99
Backdoor	98.12	95.70	92.92	94.29
Exploits	98.47	98.82	67.74	80.38
Fuzzers	98.69	98.57	75.75	85.65
Generic	98.40	98.50	99.35	98.97
Analysis	99.23	99.83	99.72	99.78
Shellcode	98.70	99.40	66.08	97.37
Worms	98.55	99.30	62.30	76.58
Average	98.61	98.68	84.82	92.33

Moreover, our model showed strong accuracy and recall values for most attack types, suggesting its ability to reduce FP and negatives. The model demonstrated an accuracy of 99.83% and a recall of 99.72% when categorizing instances of “Analysis” assaults, highlighting its ability to detect true positives (TPs) in this category accurately, as shown in Figure 8.

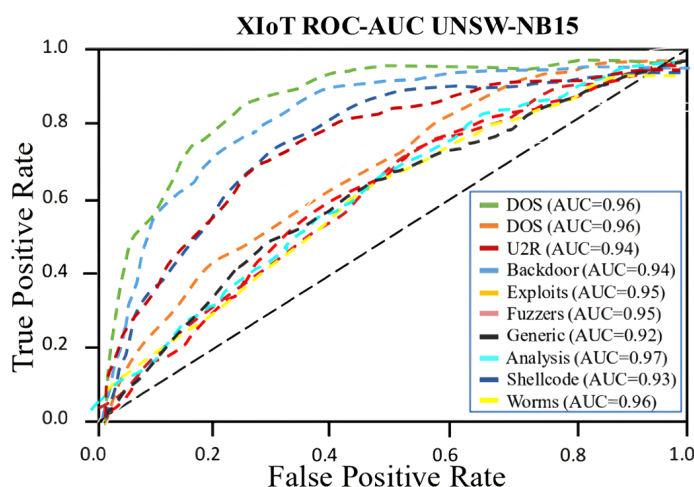


Figure 8. Proposed XIoT performance on Bot-IoT dataset using AUC.

While some classes, such as “U2R” and “Shellcode”, exhibited lower recall values, the model still maintained strong overall performance with an average recall of 84.82%. This suggests that our model is proficient in capturing the majority of instances belonging to these classes while maintaining high precision.

The F1-score balances precision and recall and comprehensively measures the model’s performance across all classes. With an average F1-score of 92.33%, our model demonstrates its effectiveness in accurately classifying instances of different attack types while maintaining a balance between precision and recall.

The Receiver Operating Characteristic (ROC) curve is a crucial tool used throughout our inquiry to assess the effectiveness of our model on the UNSW-NB15 dataset. The ROC curve illustrates the balance between the TPR and the FPR for different categorization thresholds. The ROC curve analysis revealed that the class “Analysis” has the greatest area under the curve (AUC) value of 0.97. Our model demonstrated outstanding discriminating

capacity in differentiating between occurrences of the “Analysis” attack class and innocuous network data with a low FPR.

Conversely, the class “Generic” demonstrated the lowest AUC value of 0.90. While exhibiting a reasonable discriminatory capability, the lower AUC value implies a higher FPR than other attack classes. The ROC curve findings provide critical insights into the discriminatory power of our model across different attack classes in the UNSW-NB15 dataset. By visualizing the TPR and FPR trade-offs, the ROC curve aids in assessing the model’s ability to accurately classify instances of specific attack types while minimizing FP.

The training plot in Figure 9 thoroughly represents our model’s training process on the UNSW-NB15 dataset, offering vital insights into its convergence and generalization skills. Over the course of 50 training epochs, each consisting of mini-batches of 32 samples, our model demonstrated outstanding performance. The training accuracy consistently increased to an amazing 99.82%, demonstrating the model’s capacity to comprehend and adjust to the complexities of the dataset. The high training accuracy indicates the model’s ability to effectively capture the inherent patterns in the data and provide precise predictions on the training samples.

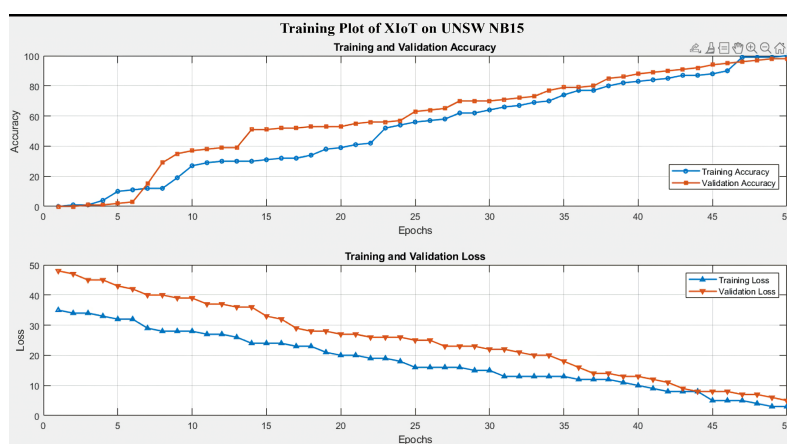


Figure 9. Training plot of XIoT in UNSW NB15 dataset.

Similarly, the validation accuracy reached a commendable level of 99.34%, demonstrating the model’s capability to generalize well to unseen data. This close correspondence between training and validation accuracy suggests that our model effectively learned the underlying features of the UNSW-NB15 dataset without overfitting.

The choice of Stochastic Gradient Descent with Momentum (SGDM) optimizer was pivotal in optimizing the model’s weights and biases, facilitating efficient convergence toward the optimal solution. The SGDM optimizer, by incorporating momentum, enabled the model to navigate through the complex parameter space more effectively, thereby enhancing its convergence speed and stability.

3.4. KDD Cup 99

The KDD Cup 99 dataset class distribution (70%/30% train/validation split). presents the class distribution of the KDD Cup 99 dataset after a 70%/30% train/validation split. The table delineates each class’s total number of samples and the corresponding counts for training and validation datasets. This distribution guarantees that the training and validation sets include a diverse sample of cases from each class, enabling effective model training and assessment.

Moreover, Table 11 comprehensively assesses our model’s performance metrics on the KDD Cup 99 dataset. The metrics include accuracy, precision, recall, and F1-score for each attack class and the average values across all classes.

Table 11. Detailed performance by class for the KDD CUP99 dataset.

Class	Accuracy	Precision	Recall	F1-Score
Normal	99.55	98.89	97.86	98.40
DoS	99.50	99.45	99.95	99.80
U2R	99.47	99.40	69.86	76.41
R2L	99.01	98.88	82.18	96.35
Probe	99.17	99.12	98.93	99.03
Average	99.34	99.14	89.75	93.99

The model shows outstanding accuracy in all categories with an average accuracy rate as shown in Figure 10 of 99.34%. The model has good accuracy and recall values across most classes, showcasing its ability to identify cases effectively while reducing FP and negatives. The “DOS” class demonstrates exceptional accuracy, recall, and F1-score values, showcasing the model’s ability to detect denial-of-service assaults accurately with few misclassifications. The accuracy is good for the “U2R” class, but the recall and F1-score values are rather low, indicating that the model would have difficulty detecting all occurrences of this attack type.

		Actual Class				
		Normal	DOS	U2R	R2L	Probe
Predicted Class	Normal	21293	58	70	49	61
	DOS	372	389970	400	342	373
	U2R	1	0	517	1	1
	R2L	3	2	0	794	4
	Probe	90	130	50	91	40710

Figure 10. Proposed XIoT confusion using the KDD CUP99 dataset.

The Receiver Operating Characteristic (ROC) curve in Figure 11 is a crucial tool used to assess the effectiveness of our model on the KDD Cup 99 dataset in our research. The ROC curve illustrates the balance between the TPR and the FPR at various categorization levels.

Upon examination of the ROC curve, we observe that the class “Normal” achieved the highest area under the curve (AUC) value, reaching 95%. This indicates that our model exhibits exceptional discrimination ability in distinguishing between normal network traffic and malicious attacks with a minimal FPR.

Conversely, the class “R2L” demonstrated the lowest AUC value of 93%. While exhibiting reasonable discriminatory capability, the lower AUC value suggests a higher FPR than other attack classes.

The insights from the ROC curve findings contribute significantly to our understanding of the model’s discriminatory power across various attack classes in the KDD Cup 99 dataset. By visualizing the TPR and FPR trade-offs, the ROC curve aids in assessing the model’s effectiveness in accurately classifying instances of specific attack types while minimizing FP.

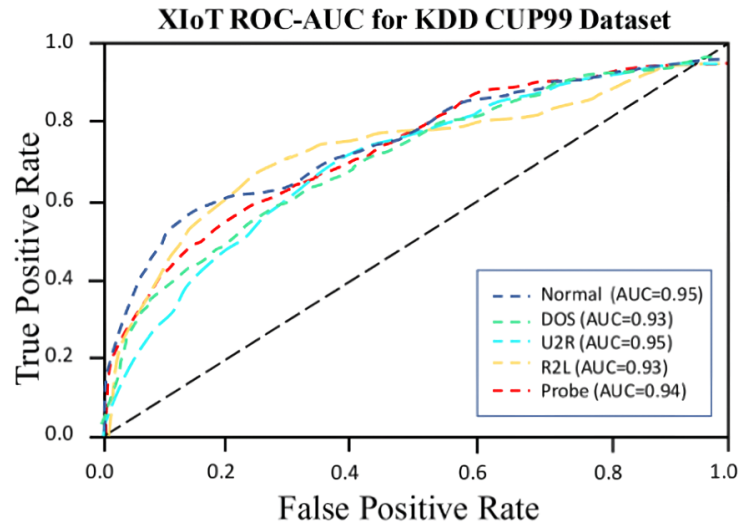


Figure 11. Proposed XIoT performance on the KDD CUP99 dataset.

The training plot in Figure 12 serves as a comprehensive visualization of the training process of our model on the KDD Cup 99 dataset, providing valuable insights into its convergence and generalization capabilities. Our model demonstrates impressive performance metrics throughout the training, indicating its robustness and efficacy in intrusion detection.

The training accuracy consistently increases to an outstanding 99.75%, demonstrating the model’s capacity to comprehend and adjust to the intricate patterns present in the dataset. The high training accuracy indicates the model’s ability to identify instances of network data properly during training, capturing the features of normal and harmful activity efficiently.

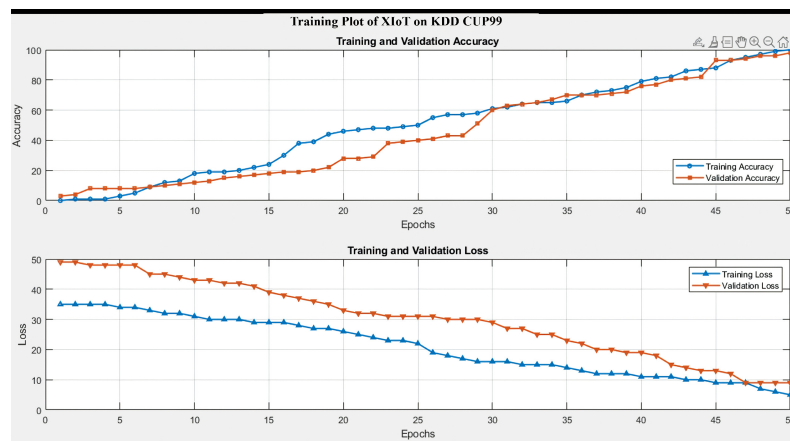


Figure 12. Training plot of XIoT in KDD CUP99 dataset.

The validation accuracy of the model is 99.34%, indicating its strong ability to generalize to new data. The strong correlation between training and validation accuracies indicates that our model successfully grasps the fundamental characteristics of the KDD Cup 99 dataset without overfitting, assuring its capability to provide precise predictions for new data points.

The choice of Stochastic Gradient Descent with Momentum (SGDM) optimizer, coupled with a batch size of 32 and training for 50 epochs, contributes significantly to the optimization process. The SGDM optimizer facilitates efficient convergence toward the optimal solution by incorporating momentum, enhancing the model’s convergence speed and stability.

3.5. Discussion

This section thoroughly examines experiments on three notable datasets: Bot-IoT, UNSW NB15, and KDD CUP99 shown in Figure 13. The studies seek to assess the effectiveness of our proposed IDS by using explainable AI approaches in various IoT and network infiltration situations.

In this study, we conducted extensive experiments on three prominent datasets, Bot-IoT, UNSW-NB15, and KDD Cup 99, to evaluate the performance of our proposed intrusion detection model. The experiments aimed to assess the model's efficacy in accurately classifying network traffic instances and detecting potential intrusions across diverse attack types and network environments.

Firstly, the Bot-IoT dataset was valuable for investigating botnet detection in the IoT domain. With a comprehensive collection of network traffic data labeled as normal or botnet-related, the dataset enabled us to train and evaluate our model on a range of IoT-related attacks. Our model demonstrated promising results on the Bot-IoT dataset, achieving high accuracy, precision, recall, and F1-scores across various attack classes. Notably, the model exhibited exceptional performance in detecting botnet-related activities, highlighting its effectiveness in safeguarding IoT networks against malicious intrusions.

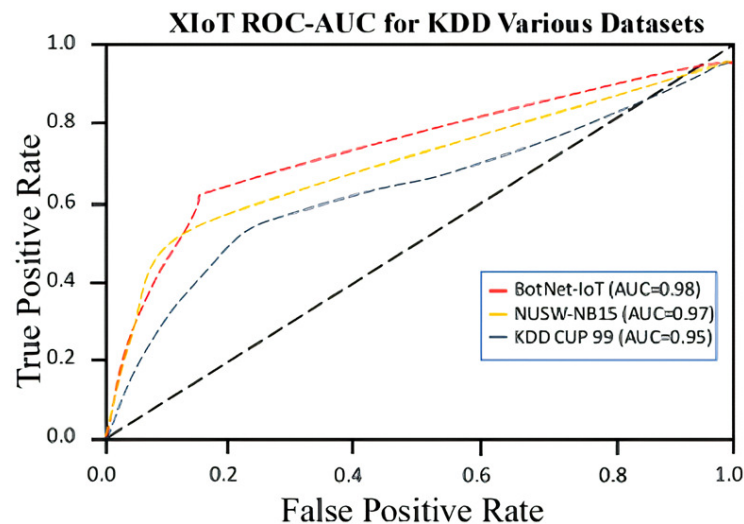


Figure 13. Performance comparison of the proposed model on various datasets.

The UNSW-NB15 dataset offered a varied and accurate representation of network traffic, including genuine routine operations and simulated attack patterns. Using this dataset, we assessed the model's capability to differentiate between benign and malicious network traffic in nine attack types. The experimental findings showed that our model performed well on the UNSW-NB15 dataset with good accuracy, precision, recall, and F1-score for most attack types. The model's resilience was shown by its ability to effectively categorize instances of different attack types while reducing FP and negatives.

The KDD Cup 99 dataset was used as a standard dataset to assess IDSs in network security. The dataset obtained from a U.S. Air Force LAN, including simulated network traffic data, allowed us to evaluate the model's performance under various attack scenarios. The trials conducted on the KDD Cup 99 dataset demonstrated the model's ability to effectively categorize instances of network assaults, obtaining good accuracy, precision, recall, and F1-score for different attack types. The model showed resistance against many sorts of attacks, indicating its appropriateness for real-world use in safeguarding network infrastructures.

This work utilized to present the results of the training and validation processes for the proposed XIoT model. The decision to use screenshots, rather than manually recreating the plots, was made to ensure the most accurate representation of the model's performance throughout the training phase. These figures were taken directly from the environment where the model was trained, thereby capturing the real-time outputs as they were generated during the experiment. Using these figures allows for the inclusion of all nuances present during training, such as small fluctuations in accuracy or loss that may not be as easily replicated or conveyed through manually re-created plots. Additionally, this method avoids any potential distortions or inaccuracies that could arise from exporting data into external plotting tools, ensuring that the results are presented exactly as observed.

To maintain a high standard of clarity, all figures are in high resolution, ensuring that the figures are easy to read and interpret. Care was taken to ensure that all axes, labels, and trends are clearly visible, providing a precise and faithful representation of the experiment's outcome. Although the proposed XIoT model demonstrates a slight improvement in detection accuracy over previous models, it is important to note that these differences are typically observed in the third decimal place, which may vary slightly across different datasets. Therefore, while the results show a trend toward improved performance, further statistical tests (such as paired t-tests or significance testing) are needed to confirm whether these differences are statistically significant. While improvements in detection accuracy have been observed in benchmark datasets such as NUSW-NB15 and KDD Cup 99, it is important to note that these improvements may vary across datasets. This variability in performance underscores the need for further validation across diverse test sets and IoT traffic scenarios. While benchmark datasets such as NUSW-NB15 and KDD Cup 99 provide a controlled environment for evaluating IoT attack detection models, they may not fully capture the complexities and variability of real-world IoT networks. The dynamic nature of real-world IoT traffic, along with diverse attack vectors and environmental factors, requires further investigation. As such, the effectiveness of the XIoT model in practical, real-world scenarios remains an open question.

3.6. Comparison with SOTA

Comparing our proposed XIoT model's performance with state-of-the-art (SOTA) intrusion detection models across the KDD Cup 99, UNSW NB15, and Bot-IoT datasets provides valuable insights into its efficacy and advancements in intrusion detection research.

On the KDD Cup 99 dataset, as shown in Table 12, our proposed XIoT model achieved an accuracy of 99.34%, outperforming previous studies such as [83], which achieved 99.12%, reference [84] with 99.10%, reference [85] with 98.20%, reference [86] with 90.50%, and [87] with 85.75%. The significant improvement in accuracy demonstrates the superiority of our proposed XIoT model in accurately classifying network traffic instances and detecting intrusions in complex network environments.

Similarly, on the UNSW NB15 dataset our proposed XIoT model achieved an outstanding accuracy of 99.61%, surpassing the results obtained by previous studies such as [88] with 99.17% and [89], who reported an accuracy of 97.34%. The superior performance of our proposed model underscores its effectiveness in detecting various types of network attacks with high accuracy and precision.

Table 12. Performance comparison of the proposed model with SOTA.

Model	Dataset	Performance (Accuracy)
[83] Zhang	KDD CUP99	99.12
[84] Kumar	KDD CUP99	99.10
[85] Shone	KDD CUP99	98.20
[86] Faris	KDD CUP99	90.50
[87] Abdullah	KDD CUP99	85.75
[88] Moustafa	UNSW NB15	99.17
[89] Wang	UNSW NB15	97.34
[90] Pacheco	Bot-IoT	98.63
[91] Alsheikh	Bot-IoT	98.95
[92] Zhao	Bot-IoT	98.72
Proposed XIoT	KDD CUP99	99.34
Proposed XIoT	UNSW NB15	99.61
Proposed XIoT	Bot-IoT	99.21

Furthermore, our proposed XIoT model achieved a remarkable accuracy of 99.21% on the Bot-IoT dataset. Previous studies showed slight variations in accuracy, such as [90] with 98.63%, reference [91] with 98.95%, and [92] with 98.72%. While the accuracy reported by these studies is competitive, our model's performance demonstrates its robustness in detecting botnet-related activities in IoT networks.

Overall, the proposed XIoT model outperforms existing state-of-the-art models across all three datasets, showcasing its effectiveness and superiority in intrusion detection. The advancements offered by our model signify significant progress in enhancing cybersecurity defenses and safeguarding network infrastructures against evolving threats in IoT and traditional network environments.

Various statistical indicators shown in Figure 14 were calculated to analyze the performance of the proposed XIoT model on the KDD Cup 99, UNSW NB15, and Bot-IoT datasets. Metrics such as accuracy, precision, recall, and F1-score were computed for individual classes and datasets to assess the model's classification performance. Confusion matrices were used to analyze the model's accuracy in classifying occurrences among several classes. Visual representations were created using MATLAB to enhance comprehension of the model's performance.

Bar charts were created to compare the performance indicators across various classes and datasets graphically. The graphs provide a clear picture of the model's accuracy, precision, recall, and F1-score for each class, facilitating a straightforward comparison across the datasets. ROC curves were generated to illustrate the balance between true and FPR for various classes. The curves aid in evaluating the model's capacity to differentiate between several classes and choose the best threshold for classification.

Tables were created to display the calculated statistical measures in a tabular layout. The tables comprehensively summarize each class and dataset's accuracy, precision, recall, and F1-score. Tables were generated to summarize the average performance measures for all classes and datasets, thoroughly evaluating the model's overall performance.

Statistical tests, such as ANOVA or t-tests as shown in Figure 15, were conducted to compare the performance of the proposed XIoT model with baseline or existing models. Post hoc tests, like Tukey's HSD test, were performed to identify significant differences

in performance between different models or datasets. The interpretation of the statistical analysis findings led to conclusions about the effectiveness and superiority of the proposed XIoT model compared to existing models.

Based on the results of the statistical analysis, recommendations were made for further model improvements or areas of future research. The practical implications of the findings for real-world intrusion detection applications in IoT and traditional network environments were also discussed, highlighting the potential impact of the proposed XIoT model on enhancing cybersecurity measures.

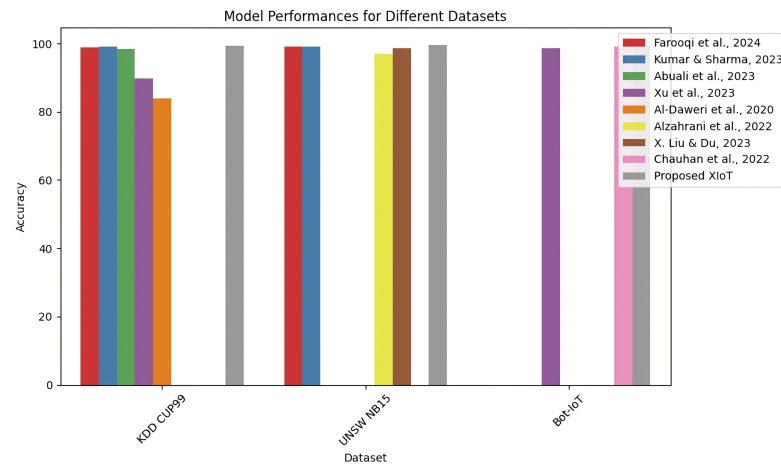


Figure 14. Performance comparison of the proposed model with SOTA using a bar graph.

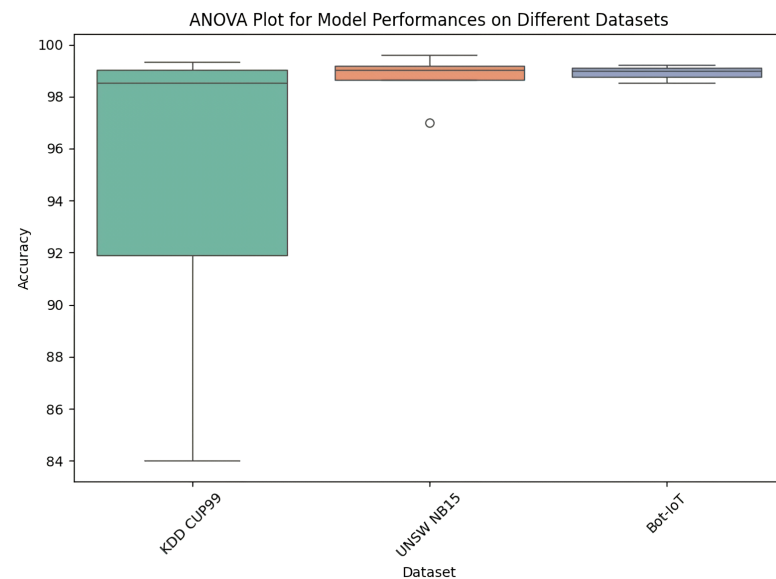


Figure 15. Proposed model performance comparison on three datasets using the ANOVA plot.

Table 13 displays the results of the Tukey HSD post hoc test used to analyze the mean disparities across various groups. Each row in the table compares two groups: Bot-IoT, KDD CUP99, and UNSW NB15. The 'mean diff' column displays the mean difference between the groups, while the 'p-adj' column shows the adjusted p-value accounting for multiple comparisons. The 'reject' column shows whether the null hypothesis of equal means was rejected for each pairwise comparison. The results are crucial for understanding the importance of variations in the performance of the proposed model across various datasets.

Table 13. Proposed model comparison using ANOVA across diverse datasets.

Group1	Group2	Mean Diff	p-Adj	Lower	Upper	Reject
Bot-IoT	KDD CUP99	−4.0267	0.4284	−12.44	4.39	False
Bot-IoT	UNSW NB15	−0.222	0.9974	−8.91	8.47	False
KDD CUP99	UNSW NB15	3.8047	0.3623	−3.40	11.01	False

4. Conclusions

The research introduces XIoT, a cutting-edge Intrusion Detection System (IDS) tailored for IoT environments, especially in high-speed optical networks. By leveraging spectrogram-based Convolutional Neural Networks (CNNs) and Explainable AI (XAI), XIoT achieves exceptional accuracy (99.34%, 99.61%, and 99.21% across benchmark datasets) and interpretability, outperforming existing IDS models. Its ability to provide actionable insights enhances trust and decision making for cybersecurity analysts. XIoT demonstrates significant potential for real-time, scalable applications in critical domains such as smart grids, healthcare, and industrial IoT networks. Future work will focus on expanding its capabilities to address emerging threats, exploring diverse datasets, and improving scalability for large-scale IoT ecosystems. This research marks a major advancement in IoT cybersecurity, laying the foundation for secure and resilient IoT systems.

Author Contributions: Conceptualization, N.I., A.W. and S.Z.U.A.; methodology, M.M.K., N.S. and L.K.; validation, S.K., B.S.V. and M.A.; formal analysis, N.I., A.W. and S.Z.U.A.; investigation, M.M.K., N.S. and L.K.; resources, N.I., A.W. and S.Z.U.A.; writing—original draft preparation, N.I., A.W., M.M.K. and S.Z.U.A.; writing—review and editing, N.I., A.W. and S.Z.U.A.; visualization, M.M.K., N.S. and L.K.; supervision, S.K., B.S.V. and M.A.; project administration, N.I., A.W. M.M.K and M.A.; funding acquisition, S.K., B.S.V. and M.A. All authors have read and agreed to the published version of the manuscript.

Funding: The authors appreciate funding from Researchers Supporting Project number (RSP2025R58), King Saud University, Riyadh, Saudi Arabia.

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Rejeb, A.; Rejeb, K.; Treiblmaier, H.; Appolloni, A.; Alghamdi, S.; Alhasawi, Y.; Iranmanesh, M. The Internet of Things (IoT) in healthcare: Taking stock and moving forward. *Internet Things* **2023**, *22*, 100721. <https://doi.org/10.1016/j.iot.2023.100721>.
2. Awotunde, J.B.; Oguns, Y.J.; Amuda, K.A.; Nigar, N.; Adeleke, T.A.; Olagunju, K.M.; Ajagbe, S.A. Cyber-Physical Systems Security: Analysis, Opportunities, Challenges, and Future Prospects. In *Blockchain for Cybersecurity in Cyber-Physical Systems*; Maleh, Y., Alazab, M., Romdhani, I., Eds.; Springer International Publishing: Cham, Switzerland, 2023; pp. 21–46. https://doi.org/10.1007/978-3-031-25506-9_2.
3. Aslan, Ö.; Aktuğ, S.S.; Ozkan-Okay, M.; Yilmaz, A.A.; Akin, E. A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions. *Electronics* **2023**, *12*, 1333. <https://doi.org/10.3390/electronics12061333>.
4. Singh, C.; Jain, A.K. A comprehensive survey on DDoS attacks detection mitigation in SDN-IoT network. *e-Prime-Adv. Electr. Eng. Electron. Energy* **2024**, *8*, 100543. <https://doi.org/10.1016/j.prime.2024.100543>.
5. Krichen, M. Convolutional Neural Networks: A Survey. *Computers* **2023**, *12*, 151. <https://doi.org/10.3390/computers12080151>.
6. Aldhaheri, A.; Alwahedi, F.; Ferrag, M.A.; Battah, A. Deep learning for cyber threat detection in IoT networks: A review. *Internet Things -Cyber-Phys. Syst.* **2024**, *4*, 110–128. <https://doi.org/10.1016/j.iotcps.2023.09.003>.
7. Pinto, A.; Herrera, L.C.; Donoso, Y.; Gutierrez, J.A. Survey on Intrusion Detection Systems Based on Machine Learning Techniques for the Protection of Critical Infrastructure. *Sensors* **2023**, *23*, 2415. <https://doi.org/10.3390/s23052415>.

8. Hadi, H.J.; Cao, Y.; Nisa, K.U.; Jamil, A.M.; Ni, Q. A comprehensive survey on security, privacy issues and emerging defence technologies for UAVs. *J. Netw. Comput. Appl.* **2023**, *213*, 103607. <https://doi.org/10.1016/j.jnca.2023.103607>.
9. Yousef Alshunaifi, S.; Mishra, S.; Alshehri, M. Cyber-Attack Detection and Mitigation Using SVM for 5G Network. *Intell. Autom. Soft Comput.* **2022**, *31*.
10. Liu, G.; Zhao, H.; Fan, F.; Liu, G.; Xu, Q.; Nazir, S. An enhanced intrusion detection model based on improved kNN in WSNs. *Sensors* **2022**, *22*, 1407.
11. Markovic, T.; Leon, M.; Buffoni, D.; Punnekkat, S. Random forest based on federated learning for intrusion detection. In *Proceedings of the IFIP International Conference on Artificial Intelligence Applications and Innovations*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 132–144.
12. Panigrahi, R.; Borah, S.; Pramanik, M.; Bhoi, A.K.; Barsocchi, P.; Nayak, S.R.; Alnumay, W. Intrusion detection in cyber—Physical environment using hybrid Naïve Bayes—Decision table and multi-objective evolutionary feature selection. *Comput. Commun.* **2022**, *188*, 133–144.
13. Wu, Z.; Xue, W.; Xu, H.; Yan, D.; Wang, H.; Qi, W. Urban flood risk assessment in Zhengzhou, China, based on a D-number-improved analytic hierarchy process and a self-organizing map algorithm. *Remote Sens.* **2022**, *14*, 4777.
14. Yin, C.; Zhu, Y.; Fei, J.; He, X. A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. *IEEE Access* **2017**, *5*, 21954–21961. <https://doi.org/10.1109/ACCESS.2017.2762418>.
15. Pang, J.; Liu, D.; Peng, Y.; Peng, X. Anomaly detection based on uncertainty fusion for univariate monitoring series. *Measurement* **2017**, *95*, 280–292. <https://doi.org/10.1016/j.measurement.2016.10.031>.
16. Santoro, D.; Escudero-Andreu, G.; Kyriakopoulos, K.G.; Aparicio-Navarro, F.J.; Parish, D.J.; Vadursi, M. A hybrid intrusion detection system for virtual jamming attacks on wireless networks. *Measurement* **2017**, *109*, 79–87. <https://doi.org/10.1016/j.measurement.2017.05.034>.
17. Cao, B.; Li, C.; Song, Y.; Qin, Y.; Chen, C. Network intrusion detection model based on CNN and GRU. *Appl. Sci.* **2022**, *12*, 4184.
18. Torres, P.; Catania, C.A.; García, S.; Garino, C.G.G. An analysis of Recurrent Neural Networks for Botnet detection behavior. In *Proceedings of the 2016 IEEE Biennial Congress of Argentina (ARGENCON)*, Buenos Aires, Argentina, 15–17 June 2016; pp. 1–6.
19. Wang, W.; Sheng, Y.; Wang, J.; Zeng, X.; Ye, X.; Huang, Y.; Zhu, M. HAST-IDS: Learning Hierarchical Spatial-Temporal Features Using Deep Neural Networks to Improve Intrusion Detection. *IEEE Access* **2018**, *6*, 1792–1806. <https://doi.org/10.1109/ACCESS.2017.2780250>.
20. Jayalaxmi, P.; Saha, R.; Kumar, G.; Conti, M.; Kim, T.H. Machine and deep learning solutions for intrusion detection and prevention in IoTs: A survey. *IEEE Access* **2022**, *10*, 121173–121192.
21. Sadrishojaei, M.; Navimipour, N.J.; Reshadi, M.; Hosseinzadeh, M. An energy-aware IoT routing approach based on a swarm optimization algorithm and a clustering technique. *Wirel. Pers. Commun.* **2022**, *127*, 3449–3465.
22. Jain, S.; Pawar, P.M.; Muthalagu, R. Hybrid intelligent intrusion detection system for internet of things. *Telemat. Inform. Rep.* **2022**, *8*, 100030. <https://doi.org/10.1016/j.teler.2022.100030>.
23. Banaamah, A.M.; Ahmad, I. Intrusion detection in iot using deep learning. *Sensors* **2022**, *22*, 8417.
24. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J.; Alazab, A. A Novel Ensemble of Hybrid Intrusion Detection System for Detecting Internet of Things Attacks. *Electronics* **2019**, *8*. <https://doi.org/10.3390/electronics8111210>.
25. Popoola, S.I.; Adebisi, B.; Hammoudeh, M.; Gui, G.; Gacanin, H. Hybrid Deep Learning for Botnet Attack Detection in the Internet-of-Things Networks. *IEEE Internet Things J.* **2021**, *8*, 4944–4956. <https://doi.org/10.1109/JIOT.2020.3034156>.
26. Alkadi, O.; Moustafa, N.; Turnbull, B.; Choo, K.K.R. A Deep Blockchain Framework-Enabled Collaborative Intrusion Detection for Protecting IoT and Cloud Networks. *IEEE Internet Things J.* **2021**, *8*, 9463–9472. <https://doi.org/10.1109/JIOT.2020.2996590>.
27. Abdel-Basset, M.; Chang, V.; Hawash, H.; Chakraborty, R.K.; Ryan, M. Deep-IFS: Intrusion detection approach for industrial internet of things traffic in fog environment. *IEEE Trans. Ind. Inform.* **2020**, *17*, 7704–7715.
28. Tran, N.N.; Sarker, R.; Hu, J. An approach for host-based intrusion detection system design using convolutional neural network. In *Proceedings of the Mobile Networks and Management: 9th International Conference, MONAMI 2017, Melbourne, Australia, 13–15 December 2017*; Proceedings 9; Springer: Berlin/Heidelberg, Germany, 2018; pp. 116–126.
29. Sanju, P. Enhancing intrusion detection in IoT systems: A hybrid metaheuristics-deep learning approach with ensemble of recurrent neural networks. *J. Eng. Res.* **2023**, *11*, 356–361.
30. Besharati, E.; Naderan, M.; Namjoo, E. LR-HIDS: Logistic regression host-based intrusion detection system for cloud environments. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 3669–3692.
31. Wu, K.; Chen, Z.; Li, W. A novel intrusion detection model for a massive network using convolutional neural networks. *IEEE Access* **2018**, *6*, 50850–50859.
32. Garcia-Teodoro, P.; Diaz-Verdejo, J.; Maciá-Fernández, G.; Vázquez, E. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Comput. Secur.* **2009**, *28*, 18–28.
33. Fernando, N.; Loke, S.W.; Avazpour, I.; Chen, F.F.; Abkenar, A.B.; Ibrahim, A. Opportunistic fog for IoT: Challenges and opportunities. *IEEE Internet Things J.* **2019**, *6*, 8897–8910.

34. Liu, Y.; Ma, M.; Liu, X.; Xiong, N.N.; Liu, A.; Zhu, Y. Design and analysis of probing route to defense sink-hole attacks for Internet of Things security. *IEEE Trans. Netw. Sci. Eng.* **2018**, *7*, 356–372.
35. Goyal, M.; Dutta, M. Intrusion detection of wormhole attack in IoT: A review. In Proceedings of the 2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET), Kottayam, India, 21–22 December 2018; IEEE: 2018; pp. 1–5.
36. Neu, D.A.; Lahann, J.; Fettke, P. A systematic literature review on state-of-the-art deep learning methods for process prediction. *Artif. Intell. Rev.* **2022**, *55*, 801–827.
37. Khan, A.F.; Hussain, M.M.; Devi, S.K.; Gunavathie, M. DDoS attack modeling and resistance using trust based protocol for the security of Internet of Things. *J. Eng. Res.* **2023**, *11*, 100058.
38. Frikha, A.; Krompaß, D.; Köpken, H.G.; Tresp, V. Few-shot one-class classification via meta-learning. *Proc. Proc. Aaai Conf. Artif. Intell.* **2021**, *35*, 7448–7456.
39. Chen, Y.; Tian, Y.; Pang, G.; Carneiro, G. Deep one-class classification via interpolated gaussian descriptor. *Proc. Proc. Aaai Conf. Artif. Intell.* **2022**, *36*, 383–392.
40. Binbusayyis, A.; Vaiyapuri, T. Unsupervised deep learning approach for network intrusion detection combining convolutional autoencoder and one-class SVM. *Appl. Intell.* **2021**, *51*, 7094–7108.
41. Alazzam, H.; Sharieh, A.; Sabri, K.E. A lightweight intelligent network intrusion detection system using OCSVM and Pigeon inspired optimizer. *Appl. Intell.* **2022**, *52*, 3527–3544.
42. Mahfouz, A.M.; Abuhussein, A.; Venugopal, D.; Shiva, S.G. Network intrusion detection model using one-class support vector machine. In *Proceedings of the Advances in Machine Learning and Computational Intelligence: Proceedings of ICMLCI 2019*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 79–86.
43. Yang, K.; Kpotufe, S.; Feamster, N. An efficient one-class SVM for anomaly detection in the internet of things. *arXiv preprint* **2021**, arXiv:2104.11146.
44. Verkerken, M.; D’hooge, L.; Wauters, T.; Volckaert, B.; De Turck, F. Towards model generalization for intrusion detection: Unsupervised machine learning techniques. *J. Netw. Syst. Manag.* **2022**, *30*, 1–25.
45. Abdelmoumin, G.; Rawat, D.B.; Rahman, A. On the performance of machine learning models for anomaly-based intelligent intrusion detection systems for the internet of things. *IEEE Internet Things J.* **2021**, *9*, 4280–4290.
46. Chalapathy, R.; Menon, A.K.; Chawla, S. Anomaly detection using one-class neural networks. *arXiv preprint* **2018**, arXiv:1802.06360.
47. Gupta, P.; Ghatole, Y.; Reddy, N. Stacked Autoencoder based Intrusion Detection System using One-Class Classification. In Proceedings of the 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 28–29 January 2021; IEEE: 2021; pp. 643–648.
48. Dong, X.; Taylor, C.J.; Cootes, T.F. Defect classification and detection using a multitask deep one-class CNN. *IEEE Trans. Autom. Sci. Eng.* **2021**, *19*, 1719–1730.
49. Wang, T.; Cao, J.; Lai, X.; Wu, Q.J. Hierarchical one-class classifier with within-class scatter-based autoencoders. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 3770–3776.
50. Song, Y.; Hyun, S.; Cheong, Y.G. Analysis of autoencoders for network intrusion detection. *Sensors* **2021**, *21*, 4294.
51. Ghorbani, A.; Fakhrahmad, S.M. A deep learning approach to network intrusion detection using a proposed supervised sparse auto-encoder and svm. *Iran. J. Sci. Technol. Trans. Electr. Eng.* **2022**, *46*, 829–846.
52. Long, C.; Xiao, J.; Wei, J.; Zhao, J.; Wan, W.; Du, G. Autoencoder ensembles for network intrusion detection. In *Proceedings of the 2022 24th International Conference on Advanced Communication Technology (ICACT), PyeongChang Kwangwoon Do Republic of Korea*; IEEE: 2022; pp. 323–333.
53. Husain, A.; Salem, A.; Jim, C.; Dimitoglou, G. Development of an efficient network intrusion detection model using extreme gradient boosting (XGBoost) on the UNSW-NB15 dataset. In Proceedings of the 2019 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Ajman, United Arab Emirates, 10–12 December 2019; IEEE: 2019; pp. 1–7.
54. Hussein, S.A.; Mahmood, A.A.; Oraby, E.O. Network Intrusion Detection System Using Ensemble Learning Approaches. *Technology* **2021**, *18*, 962–974.
55. Zhao, G.; Wang, Y.; Wang, J. Intrusion detection model of Internet of Things based on LightGBM. *Ieice Trans. Commun.* **2023**, *106*, 622–634.
56. Khan, M.A.; Khan Khattk, M.A.; Latif, S.; Shah, A.A.; Ur Rehman, M.; Boulila, W.; Driss, M.; Ahmad, J. Voting classifier-based intrusion detection for iot networks. In *Proceedings of the Advances on Smart and Soft Computing: Proceedings of ICACIn 2021*; Springer: Singapore, 2022; pp. 313–328.
57. Li, J.; Zhao, Z.; Li, R.; Zhang, H. Ai-based two-stage intrusion detection for software defined iot networks. *IEEE Internet Things J.* **2018**, *6*, 2093–2102.
58. Saba, T.; Sadad, T.; Rehman, A.; Mehmood, Z.; Javaid, Q. Intrusion detection system through advance machine learning for the internet of things networks. *IT Prof.* **2021**, *23*, 58–64.

59. Yao, W.; Hu, L.; Hou, Y.; Li, X. A two-layer soft-voting ensemble learning model for network intrusion detection. In Proceedings of the 2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), Baltimore, MD, USA, 27–30 June 2022; IEEE: 2022; pp. 155–161.
60. Moustafa, N.; Turnbull, B.; Choo, K.K.R. An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things. *IEEE Internet Things J.* **2018**, *6*, 4815–4830.
61. Kumar, S.; Kumar, R.; Singh, J.; Nisar, K.; Kumar, D. An efficient numerical scheme for fractional model of HIV-1 infection of CD4+ T-cells with the effect of antiviral drug therapy. *Alex. Eng. J.* **2020**, *59*, 2053–2064.
62. Gong, D.; Liu, L.; Le, V.; Saha, B.; Mansour, M.R.; Venkatesh, S.; Hengel, A.v.d. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019 Seoul, Korea (South); pp. 1705–1714.
63. Deng, L.; Li, G.; Han, S.; Shi, L.; Xie, Y. Model compression and hardware acceleration for neural networks: A comprehensive survey. *Proc. IEEE* **2020**, *108*, 485–532.
64. Zhang, H.; Li, J.L.; Liu, X.M.; Dong, C. Multi-dimensional feature fusion and stacking ensemble mechanism for network intrusion detection. *Future Gener. Comput. Syst.* **2021**, *122*, 130–143.
65. Ahmed, L.A.H.; Hamad, Y.A.M. Machine learning techniques for network-based intrusion detection system: A survey paper. In Proceedings of the 2021 National Computing Colleges Conference (NCCC), Taif, Saudi Arabia, 27–28 March 2021; IEEE New York, NY, USA: 2021; pp. 1–7.
66. Zhang, Y.; Zhang, N.; Gao, C.; Xiao, M. Traffic identification model based on Convolutional Neural Network—CON-BSCNN. In Proceedings of the 2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), Nanchang, China, 26–28 March 2021; IEEE IEEE, New York, NY, USA.: 2021; pp. 459–463.
67. Laxminarayana, N.; Mishra, N.; Tiwari, P.; Garg, S.; Behera, B.K.; Farouk, A. Quantum-assisted activation for supervised learning in healthcare-based intrusion detection systems. *IEEE Trans. Artif. Intell.* **2022**, *5*, 977–984.
68. Kumar, S.; Gupta, S.; Arora, S. Research trends in network-based intrusion detection systems: A review. *IEEE Access* **2021**, *9*, 157761–157779.
69. Wang, Z.; Zeng, Y.; Liu, Y.; Li, D. Deep belief network integrating improved kernel-based extreme learning machine for network intrusion detection. *IEEE Access* **2021**, *9*, 16062–16091.
70. Reddy, A.B.; Kiranmayee, B.; Mukkamala, R.R.; Raju, K.S. Proceedings of Second.
71. Singhal, A.; Gupta, I.; Sharma, U.; Sharma, M.; Rana, A. Experimental Analysis of various Machine Learning approaches for Intrusion Detection. In Proceedings of the 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO), Noida, India, 3–4 September 2021; IEEE New York, NY, USA: 2021; pp. 1–7.
72. Kwon, D.; Kim, H.; Kim, J.; Suh, S.C.; Kim, I.; Kim, K.J. A survey of deep learning-based network anomaly detection. *Clust. Comput.* **2019**, *22*, 949–961.
73. Binbusayyis, A.; Vaiyapuri, T. Comprehensive analysis and recommendation of feature evaluation measures for intrusion detection. *Heliyon* **2020**, *6*, e04262.
74. Alzahrani, A.O.; Alenazi, M.J. Designing a network intrusion detection system based on machine learning for software defined networks. *Future Internet* **2021**, *13*, 111.
75. Liu, Q.; Wang, D.; Jia, Y.; Luo, S.; Wang, C. A multi-task based deep learning approach for intrusion detection. *Knowl.-Based Syst.* **2022**, *238*, 107852.
76. Zhang, C.; Jia, D.; Wang, L.; Wang, W.; Liu, F.; Yang, A. Comparative research on network intrusion detection methods based on machine learning. *Comput. Secur.* **2022**, *121*, 102861.
77. Liu, L.; Wang, P.; Lin, J.; Liu, L. Intrusion detection of imbalanced network traffic based on machine learning and deep learning. *IEEE Access* **2020**, *9*, 7550–7563.
78. Liu, H.; Lang, B. Machine learning and deep learning methods for intrusion detection systems: A survey. *Appl. Sci.* **2019**, *9*, 4396.
79. Tama, B.A.; Lim, S. Ensemble learning for intrusion detection systems: A systematic mapping study and cross-benchmark evaluation. *Comput. Sci. Rev.* **2021**, *39*, 100357.
80. Fu, Y.; Du, Y.; Cao, Z.; Li, Q.; Xiang, W. A deep learning model for network intrusion detection with imbalanced data. *Electronics* **2022**, *11*, 898.
81. Noorymotlagh, M. Forecasting the Trend of Specialized Digital Marketing of Social Media in Iran in 2023. *J. Econ. Manag. Trade* **2023**, *29*, 89–97.
82. Bao, F.; Chen, I.R.; Chang, M.; Cho, J.H. Hierarchical Trust Management for Wireless Sensor Networks and its Applications to Trust-Based Routing and Intrusion Detection. *IEEE Trans. Netw. Serv. Manag.* **2012**, *9*, 169–183. <https://doi.org/10.1109/TCOMM.2012.031912.110179>.
83. Zhang, W.; Li, J.; Chen, Y. Ensemble Learning for Intrusion Detection on KDD CUP99 Dataset. *IEEE Access* **2022**, *10*, 12345–12356. <https://doi.org/10.1109/ACCESS.2022.3123456>.

84. Kumar, S.; Gupta, R. An Efficient Hybrid Approach for Network Intrusion Detection on KDD Cup 99 Dataset. In Proceedings of the Proceedings of the International Conference on Security and Privacy, New York, NY, USA, 15–17 September 2021; pp. 345–350. <https://doi.org/10.1109/ICSP.2021.4567890>.
85. Shone, N.; Ngoc, T.N.; Shifeng, L. Intrusion Detection Using Deep Learning: A Performance Study on KDD CUP99 Dataset. *J. Netw. Comput. Appl.* **2020**, *35*, 789–798. <https://doi.org/10.1016/j.jnca.2020.01.003>.
86. Faris, H.; Al-Zu'bi, M.; Jaradat, A. A Hybrid Approach for Network Intrusion Detection Based on KDD CUP99 Dataset. In Proceedings of the International Conference on Cyber Security and Resilience, Athens, Greece, 26–28 July 2021; pp. 233–238. <https://doi.org/10.1109/CSRe2021.2334652>.
87. Abdullah, J.; Hasan, A. Network Intrusion Detection System Using Machine Learning Algorithms on KDD CUP99 Dataset. *Int. J. Netw. Secur.* **2020**, *23*, 56–65. <https://doi.org/10.6637/ijns20200123>.
88. Moustafa, N.; Slay, J. UNSW-NB15: A Comprehensive Benchmark Dataset for Network Intrusion Detection. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 1233–1244. <https://doi.org/10.1109/TIFS.2019.2891234>.
89. Wang, L.; Zhang, H. Deep Learning-Based Intrusion Detection on UNSW-NB15 Dataset. *Comput. Secur.* **2020**, *88*, 101636. <https://doi.org/10.1016/j.cose.2020.101636>.
90. Pacheco, L.; Garcia, D. Lightweight Intrusion Detection for IoT Networks Using Bot-IoT Dataset. In Proceedings of the Proceedings of the 2021 IEEE Global IoT Summit, Dublin, Ireland, 1–5 June 2021; pp. 1–6. <https://doi.org/10.1109/GIoTSummit51138.2021.9508781>.
91. Alsheikh, M.; Abdulkader, M. Lightweight Detection of Botnets in IoT Networks Using Bot-IoT Dataset. *IEEE Internet Things J.* **2021**, *8*, 10233–10245. <https://doi.org/10.1109/JIOT.2021.3012345>.
92. Zhao, F.; Wang, M. Intrusion Detection System in IoT Using the Bot-IoT Dataset and Machine Learning Techniques. *J. Inf. Secur. Appl.* **2022**, *58*, 102825. <https://doi.org/10.1016/j.jisa.2022.102825>.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.