



ARTICLE

Robust Network Security: A Deep Learning Approach to Intrusion Detection in IoT

Ammar Odeh* and Anas Abu Taleb

Department of Computer Science, Princess Sumaya University of Technology, Amman, 1196, Jordan

*Corresponding Author: Ammar Odeh. Email: a.odeh@psut.edu.jo

Received: 03 September 2024 Accepted: 05 November 2024 Published: 19 December 2024

ABSTRACT

The proliferation of Internet of Things (IoT) technology has exponentially increased the number of devices interconnected over networks, thereby escalating the potential vectors for cybersecurity threats. In response, this study rigorously applies and evaluates deep learning models—namely Convolutional Neural Networks (CNN), Autoencoders, and Long Short-Term Memory (LSTM) networks—to engineer an advanced Intrusion Detection System (IDS) specifically designed for IoT environments. Utilizing the comprehensive UNSW-NB15 dataset, which encompasses 49 distinct features representing varied network traffic characteristics, our methodology focused on meticulous data preprocessing including cleaning, normalization, and strategic feature selection to enhance model performance. A robust comparative analysis highlights the CNN model's outstanding performance, achieving an accuracy of 99.89%, precision of 99.90%, recall of 99.88%, and an F1 score of 99.89% in binary classification tasks, outperforming other evaluated models significantly. These results not only confirm the superior detection capabilities of CNNs in distinguishing between benign and malicious network activities but also illustrate the model's effectiveness in multiclass classification tasks, addressing various attack vectors prevalent in IoT setups. The empirical findings from this research demonstrate deep learning's transformative potential in fortifying network security infrastructures against sophisticated cyber threats, providing a scalable, high-performance solution that enhances security measures across increasingly complex IoT ecosystems. This study's outcomes are critical for security practitioners and researchers focusing on the next generation of cyber defense mechanisms, offering a data-driven foundation for future advancements in IoT security strategies.

KEYWORDS

Intrusion detection system (IDS); Internet of Things (IoT); convolutional neural network (CNN); long short-term memory (LSTM); autoencoder; network security; deep learning; data preprocessing; feature selection; cyber threats

Abbreviations

IDS	Intrusion Detection System
IoT	Internet of Things
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory



1 Introduction

The Internet of Things (IoT) is transforming the modern world by enabling a vast array of devices to connect and communicate seamlessly over the internet. Its applications span from smart homes and wearable devices to large-scale industrial automation, creating a networked ecosystem that is rapidly expanding. This growth has introduced new levels of convenience, efficiency, and automation across various sectors, fundamentally changing how we interact with technology and manage everyday tasks. The ability of IoT to streamline processes and optimize resource use makes it a driving force behind the digital transformation of industries, offering unparalleled opportunities for innovation and improved quality of life [1,2]. However, this interconnected ecosystem brings new security challenges. As the number of IoT devices grows, the risk of cyber-attacks rises, making Network Intrusion Detection Systems (NIDS) a crucial element for ensuring IoT security [3,4].

IoT devices are appealing targets for cybercriminals because they frequently lack robust security measures. This vulnerability can lead to unauthorized access, data breaches, and malware infections [5]. A strong NIDS is essential in this setting, as it monitors network traffic and analyzes data to perceive and reply to malicious activities and signs of intrusion [6].

The first step in establishing an effective NIDS for IoT is data collection. This involves deploying sensors across the network to gather real-time data from various devices, gateways, and network traffic [7,8]. Logs from these devices and network equipment provide valuable insights into network events and activities. Analyzing this data is critical for identifying potential security threats [9,10].

Feature extraction is a fundamental process in NIDS. Analyzing network traffic, one can identify patterns and extract relevant features such as packet size, communication frequency, and device interaction [11]. Behavioral analysis also plays a vital role in this process. Monitoring device behavior makes it possible to detect anomalies that deviate from established activity profiles [12].

Detection techniques in NIDS for IoT can be categorized into signature-based and anomaly-based detection [13]. Signature-based detection uses a database of recognized attack designs, effective for known threats but ineffective against new ones. In contrast, anomaly-based detection identifies deviations from normal behavior, allowing it to detect previously unknown attacks [14].

Machine learning algorithms are increasingly being working to enhance the detection capabilities of NIDS. These algorithms analyze patterns in network traffic and device behavior, allowing for more accurate and timely predictions of potential intrusions. Techniques such as clustering, classification, and deep learning can significantly improve the effectiveness of NIDS [15,16].

Once a potential threat is detected, the NIDS must generate alerts to notify administrators for further investigation [17]. Automated responses can be triggered in more advanced systems to mitigate the danger. These responses may include blocking malicious traffic, quarantining affected devices, or initiating predefined security protocols [18].

The implementation of NIDS for IoT networks comes with its own set of challenges. IoT devices often need more processing power and memory, restricting the complexity of algorithms that can be deployed [19]. Additionally, the diverse range of communication protocols used in IoT networks complicates the development of a universal NIDS solution. IoT devices' high volume of data can also overwhelm traditional NIDS solutions, necessitating efficient data processing and analysis techniques. Privacy concerns must also be addressed, as collecting and analyzing data from IoT devices can raise user privacy and protection issues [20].

Despite these challenges, implementing NIDS for IoT is crucial in securing modern digital environments. Organizations can protect their IoT networks from a wide range of security threats

by defining clear security policies, deploying sensors, analyzing behavior, and employing advanced detection algorithms. Regular updates and maintenance of the NIDS are essential to adapt to evolving threats and ensure optimal performance.

With the expanding IoT landscape, robust Network Intrusion Detection Systems are crucial for securing IoT networks and connected devices. By using advanced detection techniques and addressing IoT-specific challenges, organizations can strengthen their defenses and protect their digital assets from cyber threats.

Traditional Intrusion Detection Systems (IDS) struggle to secure complex IoT networks due to their static, rule-based nature and the challenges of processing large data volumes from numerous IoT devices in real-time. Our research goals to address these limitations by applying advanced deep learning models like Convolutional Neural Networks (CNN), Autoencoders, and Long Short-Term Memory (LSTM) networks. The goal is to develop an adaptive and accurate IDS capable of anticipating and mitigating emerging threats in modern IoT environments.

The motivation for this research stems from the escalating threat landscape in IoT networks, where traditional security measures have consistently fallen short in protecting against dynamic and increasingly sophisticated cyber attacks. Recent high-profile breaches have underscored the vulnerability of IoT devices, many of which lack robust built-in security features, making them prime targets for cybercriminals. This susceptibility is compounded by the diversity and volume of IoT traffic, which traditional IDS tools are ill-equipped to monitor effectively due to their reliance on predefined rubrics and signatures that often fail to distinguish novel or developing threats. Inspired by the potential of deep learning to transcend these limitations, this study is driven by the aim to harness the power of algorithms such as CNNs, Autoencoders, and LSTMs. These models offer promising capabilities in pattern recognition, anomaly detection, and automated learning from vast amounts of unstructured data—features that are critical in developing an IDS that is not only reactive but also proactive in its defense mechanisms. Furthermore, the academic and industrial urgency to fortify IoT infrastructures provides a compelling impetus for exploring innovative approaches that can adapt quickly to the changing tactics of cyber adversaries, thereby contributing pointedly to the field of cybersecurity.

This paper introduces several novel contributions to the field of network security within IoT environments, distinguishing it from existing methodologies:

1. **Integration of Multiple Deep Learning Models:** Unlike traditional approaches that often utilize a single model, this study uniquely combines Convolutional Neural Networks (CNNs), Autoencoders, and Long Short-Term Memory (LSTM) networks in a cohesive framework. This integration leverages the strengths of each model to increase the detection correctness and trustworthiness of the IDS, providing a comprehensive solution to diverse attack vectors.
2. **Dynamic Threat Detection Capabilities:** The proposed IDS dynamically adapts to new and evolving threats without the need for frequent manual updates. This is accomplished through the continuous learning capabilities intrinsic in deep learning models, which allow the system to automatically update its threat detection criteria based on newly encountered data patterns.
3. **Real-time Processing of IoT Traffic:** By optimizing the deep learning models for high efficiency, the system is designed to procedure huge volumes of data in actual time. This capability is critical in IoT contexts, where delay in threat detection can lead to significant security breaches and damages.
4. **Extensive Validation on Modern Datasets:** The study employs the UNSW-NB15 dataset, one of the most recent and comprehensive datasets available, which includes a mix of modern usual

activities and synthetic contemporary attack actions. This choice ensures that the findings are relevant and applicable to current security challenges in IoT networks.

5. **Focus on Model Generalizability and Scalability:** Special attention is given to developing models that are not only effective in detecting known types of attacks but also capable of scaling to accommodate the rapid growth of IoT devices and traffic. This scalability is crucial for ensuring long-term applicability of the IDS across various sectors and device ecosystems.

This research demonstrates that deploying deep learning models like CNNs, Autoencoders, and LSTMs for Intrusion Detection Systems (IDS) in IoT networks offers significant advantages for industry. These models provide accurate detection of various cyber threats, making them ideal for sectors like healthcare and finance where data security is critical. The CNN model, in particular, excels in accuracy and precision, enhancing network security. Additionally, the scalability of these models allows them to adapt as IoT networks grow, minimizing the need for manual intervention. By applying feature selection and preprocessing techniques, the study reduces false positives, improving efficiency, maintaining trust, and reducing costs related to security management. This approach results in a scalable, cost-effective, and adaptable IDS solution for modern industrial needs.

2 Related Works

Mayuranathan et al. [21] proposed a Best Features Intrusion Detection System (IDS) using a Restricted Boltzmann Machine (RBM) to detect DDoS attacks in a cloud environment. It aimed to recognize the most related features for detection, enhancing accuracy with the RBM model. The results showed that the RBM effectively reduced false positives and improved detection rates, offering a reliable solution for securing cloud-based systems against DDoS threats.

SaiSindhuTheja et al. [22] developed an efficient feature selection method using a metaheuristic algorithm combined with a Recurrent Neural Network (RNN) for DoS attack detection in cloud computing. It highlights how feature selection reduces computational complexity and enhances IDS performance. The metaheuristic algorithm optimizes feature selection, while the RNN accurately detects DoS attacks, achieving high detection accuracy and robustness against different attack patterns.

Qais et al. [23] introduced the Transient Search Optimization (TSO), a new meta-heuristic optimization algorithm. The TSO algorithm was applied to optimize the parameters of IDS models, improving their efficiency and accuracy. Their research highlighted the algorithm's ability to find optimal solutions quickly and effectively, making it appropriate for real-time intrusion finding applications. The TSO algorithm outperformed other optimization techniques regarding convergence speed and solution quality.

Kfoury et al. [24] presented a Self-Organizing Map (SOM) Intrusion Detection System for detecting attacks on the RPL protocol in IoT environments. The SOM-based IDS identified various RPL attacks, such as sinkhole and blackhole attacks, by clustering similar patterns and recognizing anomalies. The approach demonstrated high correctness and low false-positive rates in detecting intrusions.

Waheed et al. [25] integrated machine learning and blockchain to enhance IoT security and privacy. It evaluated machine learning algorithms for intrusion detection and used blockchain for secure data sharing and verification. The approach addressed key threats, highlighting machine learning's role in threat detection and blockchain's role in data integrity. The results showed notable improvements in securing IoT systems against common vulnerabilities.

Li et al. [26] compared feature selection and feature extraction techniques to optimize IoT intrusion detection systems using machine learning. It aimed to determine the best approach for improving detection accuracy while minimizing computational costs. By evaluating various machine learning algorithms and feature engineering methods, the research found that feature selection was more efficient and effective than feature extraction. The findings offer valuable insights for designing optimal IDS solutions in IoT environments.

Altulaihan et al. [13] developed an Anomaly Detection Intrusion Detection System (IDS) for identifying DoS attacks in IoT networks using machine learning algorithms. It aimed to create a lightweight and efficient IDS capable of real-time detection. By employing machine learning techniques, the model effectively identified anomalies associated with DoS attacks. The approach achieved high detection accuracy with minimal computational demands, making it ideal for use in resource-limited IoT devices.

Khanday et al. [27] proposed a new data preprocessing model for lightweight IoT intrusion detection, aimed at boosting IDS performance by minimizing noise and irrelevant features in sensory data. Using feature normalization, selection, and transformation, the model prepared data for machine learning algorithms. The results demonstrated improved IDS accuracy and speed, making it ideal for real-time IoT applications.

The following [Table 1](#) delivers a inclusive summary of related works in the intrusion detection systems (IDS) for IoT networks. Each entry outlines the approach used by different researchers, highlighting the specific methodologies and techniques employed. The table also presents the advantages and disadvantages associated with each method, offering a balanced perspective on their efficacy and limitations. This comparative analysis facilitates a deeper understanding of the current state of IDS technologies, enabling researchers and practitioners to categorize the advantages and disadvantages of various approaches and make informed decisions about their application in securing IoT environments.

Table 1: Related work comparison

Reference	Approach	Advantages	Disadvantages
[21]	Utilized an RBM model to detect DDoS attacks in cloud environments	Reduces false positives, improves detection rates for DDoS attacks	Limited to DDoS detection, may not generalize well to other types of attacks
[22]	Metaheuristic Algorithm-Based Feature Selection and RNN for DoS attack detection	High detection accuracy, robust against various attack patterns	High computational complexity due to RNN
[23]	Transient Search Optimization (TSO) for optimizing IDS models	Quickly finds optimal solutions suitable for real-time applications	Requires careful parameter tuning for best performance
[24]	Self-Organizing Map (SOM) IDS for RPL protocol attacks	High accuracy and low false-positive rates for detecting RPL attacks	Limited to specific types of attacks (RPL)

(Continued)

Table 1 (continued)

Reference	Approach	Advantages	Disadvantages
[25]	Machine learning and blockchain for enhancing IoT security and privacy	Improved security and privacy, addresses common vulnerabilities and attack vectors	Integration complexity between machine learning and blockchain
[26]	Comparison of feature selection and feature extraction techniques for IoT IDS	Feature selection is efficient and effective, enhances detection accuracy	Feature extraction methods may be computationally expensive
[13]	Anomaly Detection IDS to identify DoS attacks in IoT networks	High detection rates, suitable for resource-constrained IoT devices	Focused on DoS attacks may require adaptation for other types of attacks
[27]	Data preprocessing model for efficient intrusion detection in lightweight sensory IoT systems	Enhances IDS performance, reduces noise and irrelevant features	Preprocessing may introduce additional computational overhead

Recent studies, such as the one conducted by Ali et al. [28], study emphasizes the usefulness of machine learning techniques in detecting DDoS attacks on low-power IoT devices. It shows that customized machine learning models can maintain high detection accuracy while operating efficiently within the limitations of low-power environments. This research aligns with our focus on developing scalable and efficient intrusion detection systems that are adaptable to both high-power and resource-constrained environments. The findings support the viability of advanced machine learning methods in enhancing cybersecurity measures across diverse technological landscapes.

This research significantly advances the field of intrusion detection in IoT networks by integrating innovative deep learning models that surpass the capabilities of conventional systems, which are predominantly rule-based and static in nature. Unlike existing models that often fail to keep pace with the rapid evolution of cyber threats, the deep learning approaches explored in this paper—Convolutional Neural Networks (CNNs), Autoencoders, and Long Short-Term Memory (LSTM) networks—offer dynamic adaptability and superior performance in real-time threat detection and classification. By employing a composite model strategy, this work not only enhances detection rates but also reduces false positives, a critical issue in traditional IDS that often leads to unnecessary alerts and security fatigue.

Furthermore, the justification for this research is rooted deeply in the pressing need for advanced detection systems capable of handling the unique complexities of IoT environments. Current literature and existing technologies exhibit significant gaps in addressing multi-vector attacks and in processing the enormous volumes of data generated by IoT devices without compromising system responsiveness. The novel integration of multiple deep learning techniques proposed in this study addresses these gaps by providing a scalable, efficient, and highly accurate IDS capable of adapting to new threats autonomously. This approach not only enriches the academic discourse with validated, cutting-edge

methodologies but also serves a practical need within the cybersecurity industry, where the demand for robust, adaptable, and scalable security solutions is ever-increasing.

3 Dataset and Proposed Model

In this section, we will provide a comprehensive discussion on the dataset utilized for our research and detail the workflow of the proposed system. We will delve into the characteristics and sources of the dataset, followed by an in-depth explanation of each step in our system's workflow, from data preprocessing to final implementation and analysis.

3.1 Dataset

The UNSW-NB15 dataset [29] is a inclusive dataset designed for evaluating the performance of network intrusion detection systems. It was created using the IXIA PerfectStorm tool to generate a mix of real modern normal events and synthetic modern attack actions. The dataset is widely used in research to develop and benchmark intrusion detection algorithms.

For the experimentation in this study, we utilized the UNSW-NB15 dataset, a widely recognized dataset designed specifically for network intrusion detection research. This dataset is considered secondary data as it was initially developed by the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS) and has been used in various research projects globally to benchmark network intrusion detection systems.

The UNSW-NB15 dataset comprises a mix of real modern normal activities and synthetic modern attack activities, which simulate a diverse range of intrusion scenarios that are typical in IoT networks. The dataset contains approximately 2.5 million records, each consisting of 49 features that include basic information such as source and destination Internet Protocols (IPs), ports, timestamps, and traffic type (e.g., Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Internet Control Message Protocol (ICMP)), as well as content features derived from packet payloads, such as the number of bytes and packets. These features are crucial for training models to classify and predict network traffic behaviors accurately.

The dataset is split into two sets: a training set, which includes 1.75 million records, and a testing set with 750,000 records. This division allows for comprehensive training and rigorous validation of the deep learning models employed in this study, ensuring that the models are tested against unseen data to evaluate their generalization capabilities effectively.

Number of Features

The UNSW-NB15 dataset involves of 49 features that capture various network traffic characteristics. These features include a mix of elementary features, content features, time-based features, connection-based features, and supplementary generated features. The features can be broadly categorized as follows:

1. **Basic Features:** encompasses attributes such as source IP, destination IP, source port, destination port, and protocol.
2. **Content Features:** Contains information extracted from the payload of the packets, such as the number of bytes, packets, and the connection content.
3. **Time-Based Features:** Encompasses features related to the timing of the connections, such as duration and inter-arrival times.
4. **Connection-Based Features:** Captures the behavior of connections over time, including characteristics like the number of contacts to the same host.

5. **Additional Generated Features:** These are features generated using algorithms and heuristics to provide more insight into the traffic behavior.

3.2 *Types of Labels*

The dataset includes two primary types of labels:

1. **Binary Label (Label):** This label indicates whether the traffic is benign (0) or malicious (1). It is used for binary classification tasks where the objective is to distinguish between normal and attack traffic.
2. **Multiclass Label (Attack Category):** This label provides more granularity by categorizing malicious traffic into nine different types of attacks. These attack categories include:
 - o **Fuzzers:** Attacks aimed at uncovering security weaknesses by flooding the system with large volumes of random data.
 - o **Analysis:** Attacks involving various forms of data analysis to discover vulnerabilities.
 - o **Backdoor:** Attacks where an attacker gains unauthorized access to a system through a hidden method.
 - o **DoS:** Denial of Service attacks aimed at making a system unavailable to its intended users.
 - o **Exploits:** Attacks that exploit weaknesses in software or hardware.
 - o **Generic:** Attacks not specific to any particular protocol or software.
 - o **Reconnaissance:** Attacks focused on gathering information about a target system.
 - o **Shellcode:** Attacks involving shellcode injection to be executed by the system.
 - o **Worms:** Malware that replicates itself and propagates through networks.

3.3 *Significance of the Dataset*

The UNSW-NB15 dataset is significant for several reasons:

1. **Realistic Traffic:** The dataset combines real-world network traffic with synthetic attack patterns, providing a realistic and challenging environment for testing intrusion detection systems.
2. **Diverse Attack Types:** With nine different attack categories, the dataset allows researchers to develop and evaluate systems capable of detecting various attacks, from traditional DoS attacks to more sophisticated exploits and reconnaissance operations.
3. **Comprehensive Feature Set:** The 49 features cover a broad spectrum of network traffic characteristics, enabling the development of detailed and accurate detection models. These features allow both traditional machine learning and advanced deep learning techniques to be applied effectively.
4. **Benchmarking:** The UNSW-NB15 dataset is widely used in the research community, making it a standard benchmark for comparing the performance of different intrusion detection algorithms. This common ground facilitates the development of better and more robust detection systems.

The UNSW-NB15 dataset is a crucial resource for investigators and consultants in network security, providing a realistic and comprehensive benchmark for developing and evaluating intrusion detection systems.

4 Proposed Model

Fig. 1 illustrates a comprehensive workflow for developing and evaluating an Intrusion Detection System (IDS) using the UNSW-NB15 dataset. The process is divided into several key stages: Data Preprocessing, Data Optimization, Data Split, Model Training with Deep Learning Models, and Performance Evaluation. Each stage is crucial for building an effective IDS.

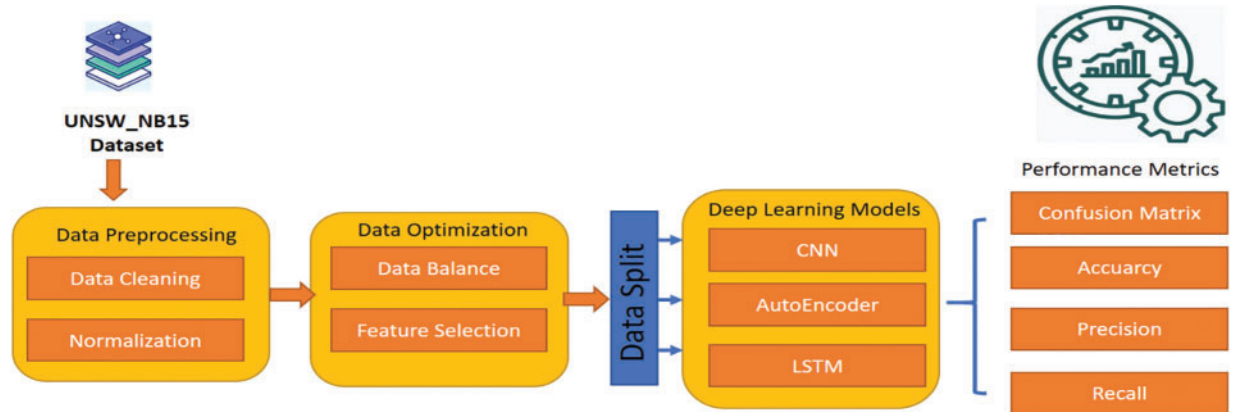


Figure 1: Workflow of the proposed system

4.1 UNSW-NB15 Dataset

The workflow begins with the UNSW-NB15 dataset, a well-known network intrusion detection dataset. It contains a mix of normal network traffic and various types of attack traffic, providing a rich dataset for training and evaluating IDS models.

4.2 Data Preprocessing

The first major step involves Data Preprocessing, essential for cleaning and preparing the data for analysis. This step includes:

- **Data Cleaning:** This process eliminates corrupted or irrelevant data from the dataset, ensuring that the data used for training the models is clean and error-free. This step is crucial for maintaining the models' performance and accuracy.
- **Normalization:** Normalization scales the dataset's features to a standard range [0, 1]. This is done using the `MinMaxScaler` from the `sklearn.preprocessing` module. This step helps accelerate the training procedure and improves the performance of the models by ensuring that all features contribute equally to the learning process.

4.3 Data Optimization

After preprocessing, the data undergoes optimization to enhance the quality and effectiveness of the training process. This step includes:

- **Data Balance:** Ensuring that the dataset is balanced regarding class distribution is crucial for training an unbiased model. The balancing algorithm used in the provided code is a stratified sampling technique with replacement. This algorithm is designed to create a balanced dataset by ensuring that each class within the target variable, in this case, `attack_cat`, is represented equally in the final dataset. The process begins by segregating the dataset into smaller subsets based on

the unique classes found in the target variable. For each class, the algorithm randomly selects a specified number of samples, which is defined by `n_samples_per_class`. The sampling is done with replacement, meaning that the same data point can be selected more than once, allowing the algorithm to reach the desired sample size even if the original class distribution is imbalanced or contains fewer instances than required. Once the algorithm completes the sampling process for each class, these balanced subsets are then combined to form a new dataset. This method is particularly effective *in situ* where the original dataset is highly imbalanced, as it creates a more uniform distribution of classes, thereby enhancing the model's ability to learn from each category equally and improving overall model performance.

- **Feature Selection:** An integral part of our methodology was the feature selection process, which aimed to identify the most predictive features of the UNSW-NB15 dataset for effective intrusion detection. The feature selection was guided by a combination of statistical techniques and domain knowledge to ensure that the final model was both efficient and accurate.
 1. **Statistical Filtering:** Initially, we applied statistical filters to eliminate features with low modification and high correlation. Features with low variance do not contribute significantly to the model's ability to separate between classes, while highly correlated features provide redundant information, which can lead to overfitting. For example, features such as 'source IP' and 'destination IP' were excluded because they do not typically vary much within the context of controlled experimental traffic and hence offer limited predictive power.
 2. **Information Gain and Mutual Information:** We then utilized Information Gain and Mutual Information criteria to evaluate and rank the remaining features based on their individual contributions to the prediction of the target variable. These measures helped us identify which features were most informative about the presence of intrusions, leading to the retention of features like 'packet size', 'number of packets', and specific flag types that are indicative of network anomalies.
 3. **Domain Expertise:** In conjunction with these statistical methods, domain expertise played a crucial role in the feature selection process. Certain features that are known within the cybersecurity community to be indicative of intrusion activities, such as unusual TCP/UDP port numbers and specific protocol behaviors, were specifically included. This approach ensured that the model was attuned to both empirical evidence and theoretical knowledge of network intrusion tactics.
 4. **Iterative Testing:** This process involved conducting multiple training cycles with various feature combinations and evaluating performance metrics like accuracy, precision, recall, and F1 score to identify the optimal feature set.

This thorough feature selection process not only simplified the model by reducing the dimensionality of the input data but also improved its ability to generalize to new, unseen data, leading to enhanced overall detection performance.

4.4 Data Split

Fig. 2 illustrates a 5-fold cross-validation approach for splitting the dataset into training, validation, and test sets. This method ensures that the model is trained, validated, and tested on different subsets of the data, providing a comprehensive evaluation of its performance.



Figure 2: Illustration of the K-fold cross-validation scheme with an 80–20 training-testing split and 20% of each fold used for validation to optimize model parameters before final evaluation

The training set, comprising 80% of the data, is subjected to 5-fold cross-validation, where it is divided into five subsets. Each subset is used once for validation while the other four are used for training. This process helps fine-tune the model's parameters based on the validation results from each fold, ensuring better generalization to unseen data.

4.5 Deep Learning Models

The next stage involves training various deep-learning models on the training data. The Fig. 1 highlights three types of models:

- **Convolutional Neural Networks (CNN):** CNNs are particularly effective for analyzing data with spatial hierarchies, making them suitable for image-like data representations. They are used here to detect network traffic data patterns that signify intrusion.
- **Autoencoder:** An Autoencoder is an unsupervised learning model for anomaly detection. It learns to compress data and then reconstruct it. Intrusions are detected by identifying significant deviations between the original and reconstructed data.
- **LSTM (Long Short-Term Memory):** LSTM networks are a type of recurrent neural network (RNN) capable of learning long-term dependencies. They are particularly effective for time-series data, making them suitable for detecting temporal patterns in network traffic that may indicate intrusions.

4.6 Performance Metrics

The final stage involves evaluating the performance of the trained models using various performance metrics. These metrics include:

- **Confusion Matrix:** Fig. 3 illustrates the performance of a classification model by comparing actual vs. predicted classifications, offering insights into true positives, false positives, and false negatives.

$$\text{False Postive Rate (FPR)} = \frac{FP}{FP + TN} \quad (1)$$

$$F - \text{Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \quad (2)$$

- **Accuracy:** The proportion of correctly classified instances among the total cases. It gives an overall measure of the model's performance.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

- **Precision:** The ratio of true positive predictions to the total predicted positives. It indicates the accuracy of the positive predictions.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4)$$

- **Recall:** The ratio of true positive predictions to the actual positives. It measures the model's ability to detect all relevant instances.

$$\text{True Postive Rate (Recall)} = \frac{TP}{TP + FN} \quad (5)$$

This workflow provides a structured approach to developing and evaluating an Intrusion Detection System using the UNSW-NB15 dataset. Each step, from data preprocessing to performance evaluation, is essential to ensure that the IDS is effective and efficient. By following this workflow, researchers and practitioners can develop robust IDS models that accurately detect network intrusions.

		Actual	
		Positive	Negative
Predicted	Positive	True Positiive	False Positive
	Negative	False Negative	True Negative

Figure 3: Confusion matrix

5 Experiment Results and Discussion

In this section, we present a series of experimental results to evaluate the performance of the three proposed deep-learning models.

5.1 Feature Selection

Figs. 4 and 5 provided represent the top 10 features selected using two different feature selection techniques: Information Gain (IG) and Gain Ratio (GR). Each figure visualizes the most significant features that will likely enhance the performance of intrusion detection systems (IDS) when applied to the UNSW-NB15 dataset.

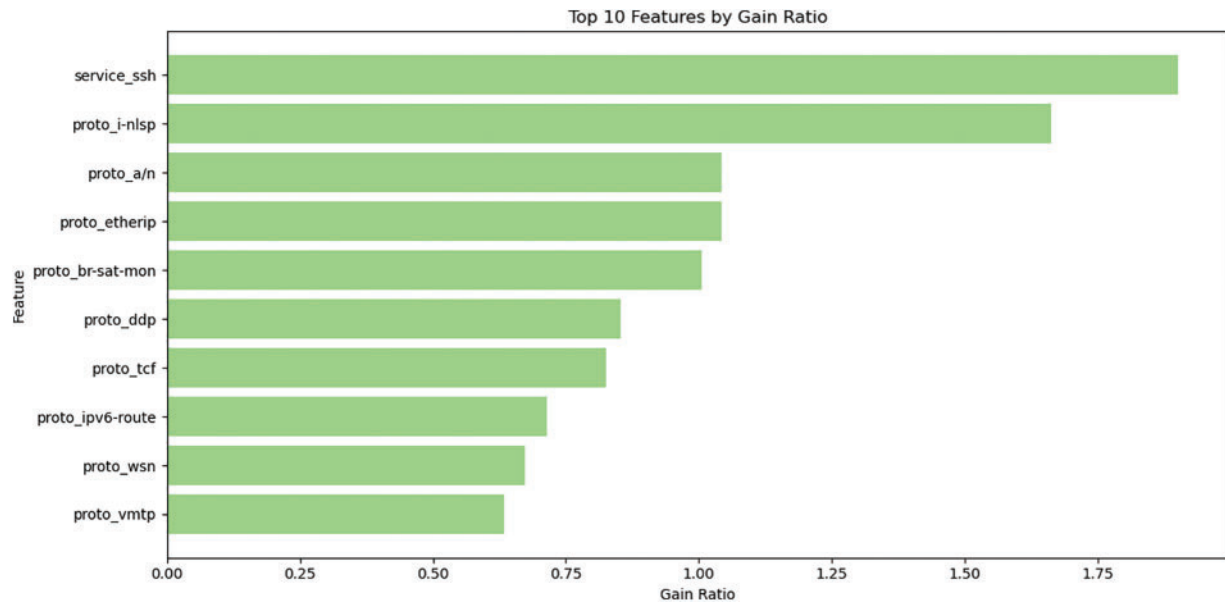


Figure 4: Top 10 features selected by Gain Ratio

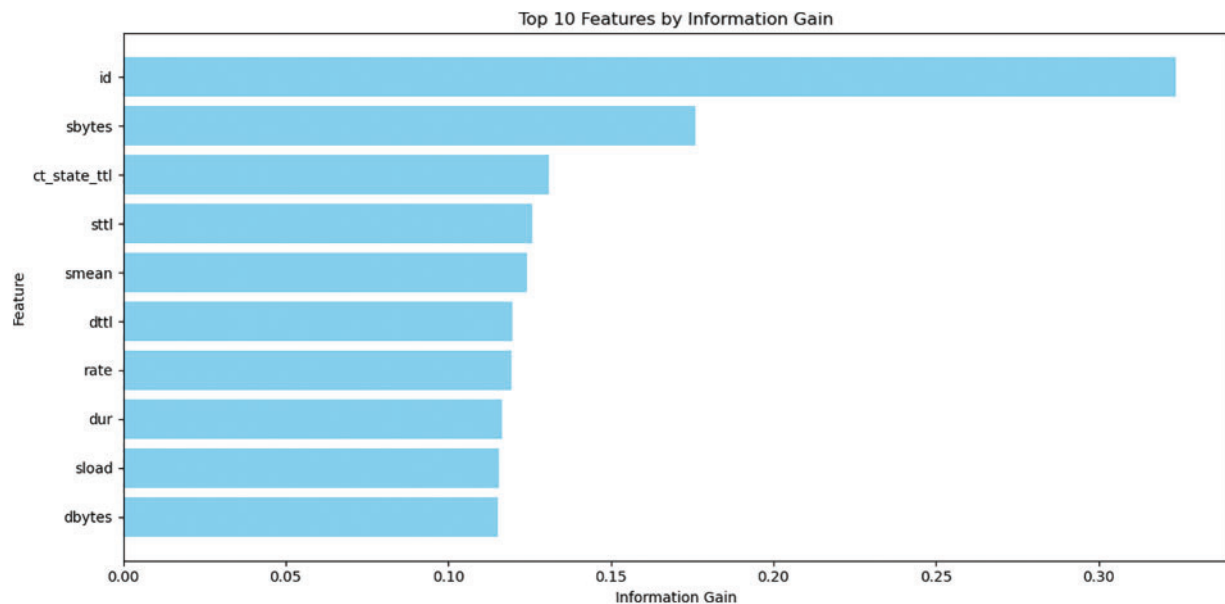


Figure 5: Top 10 features selected by Information Ratio

The features identified by both Information Gain and Gain Ratio are crucial for improving the performance of intrusion detection models. Using these top features, models can achieve higher accuracy, precision, and recall by focusing on the most relevant data attributes. By incorporating these features into the experiments, we can ensure that the models are trained on the most informative aspects of the dataset, leading to more robust and reliable intrusion detection systems.

The most correlated feature of Multi-Class Classification vs. Binary Classification.

Fig. 6 illustrates the top 10 features most correlated with the target variable “label” for binary classification in the UNSW-NB15 dataset. The height of each bar represents the correlation strength between each feature and the target variable, which differentiates between benign and malicious network traffic. Top Correlated Feature: sttl with a correlation of 0.50. Other Highly Correlated Features: swin (0.41), ct_dst_sport_ltm (0.39), id (0.39), dwin (0.37), ct_src_dport_ltm (0.34), rate (0.33), ct_state_ttl (0.32), ct_srv_dst (0.29), and ct_srv_src (0.29).

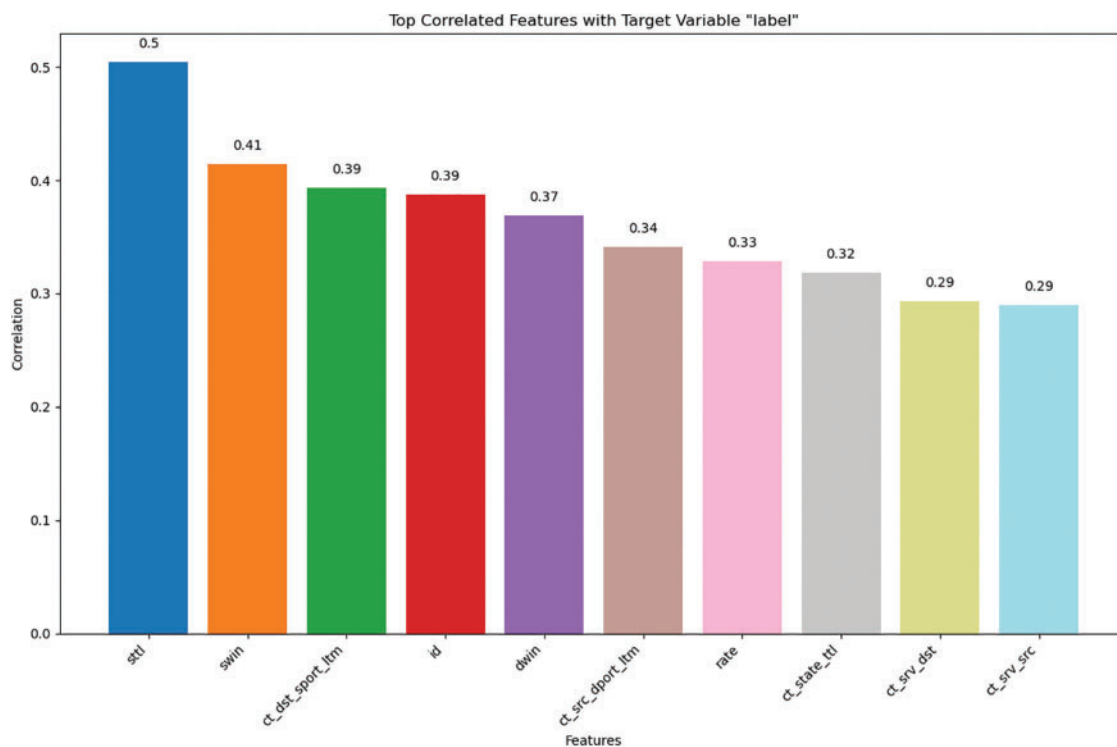


Figure 6: Top 10 correlated features with target variable label

These features are critical in determining the classification of network traffic as benign or malicious, thus playing a key role in the effectiveness of intrusion detection systems.

Fig. 7 illustrates the distribution of various attack categories in the original UNSW-NB15 dataset. It reveals a significant imbalance, with the ‘Normal’ category comprising 37,000 instances, dwarfing the counts of other categories like ‘Generic’ with 18,871 cases, ‘Exploits’ with 11,132 instances, and much smaller categories like ‘Worms’ with only 44 cases. This uneven distribution indicates a potential bias in the dataset, where the majority class (‘Normal’) vastly outnumbers the minority classes.

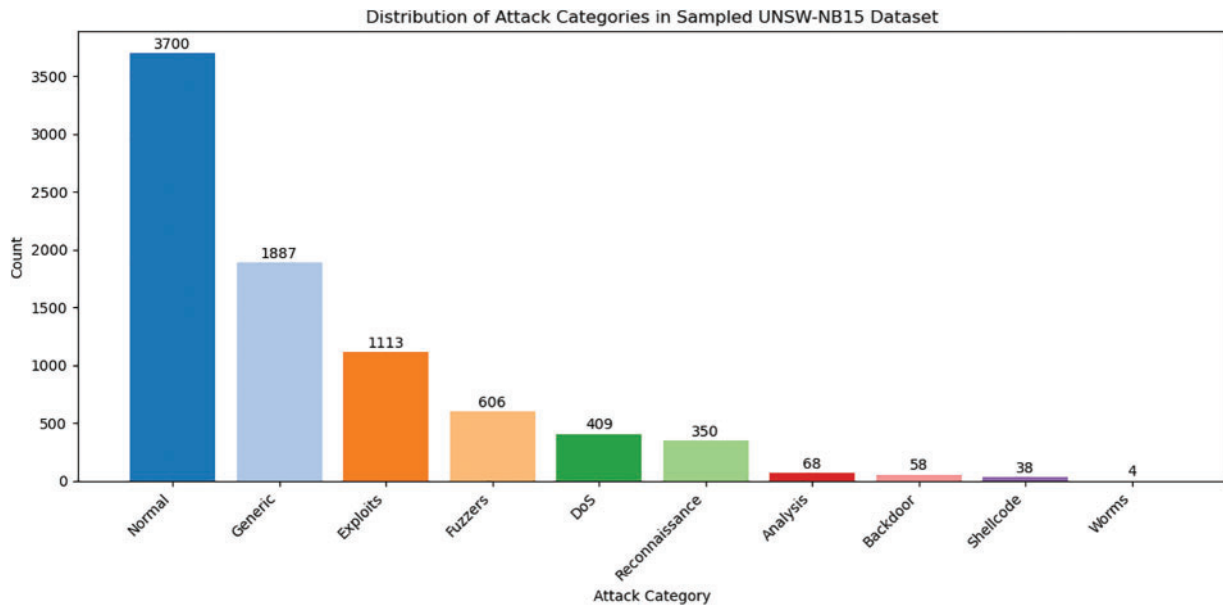


Figure 7: Distribution of attack categories in balanced UNSW-NB15 dataset

Fig. 8 illustrates the distribution of attack categories after balancing the UNSW-NB15 dataset, where each category, including ‘Normal’ and various attack types like ‘Reconnaissance,’ ‘DoS,’ and ‘Exploits,’ is represented by 500 instances. Balancing the dataset is crucial for training fair and accurate multiclass classification models. It prevents bias toward majority classes, allowing the model to better learn the features of all categories. This leads to improved accuracy, precision, recall, and overall robustness, enhancing the effectiveness of intrusion detection systems.

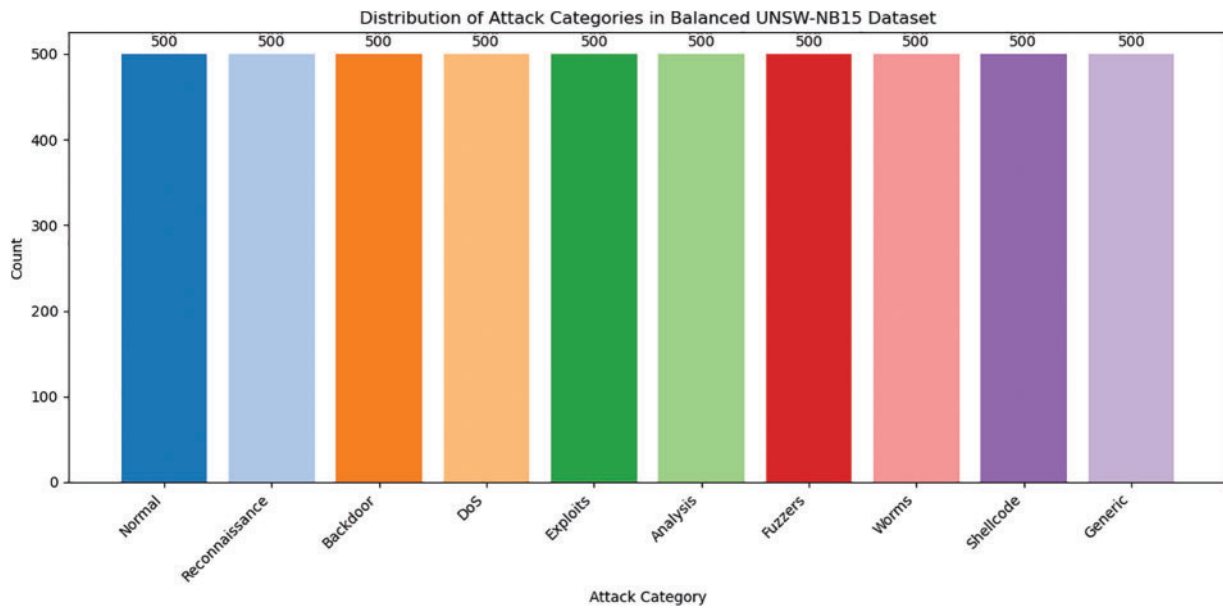


Figure 8: Attack category distribution in the UNSW-NB15 dataset

In this study, the reward function is defined to systematically incentivize the model towards optimal behaviors that align with our specific objectives. The function operates by assigning a numerical reward based on the accuracy and efficiency of the actions taken by the model within the environment. Parameters influencing the reward include the magnitude of the action's impact on the system's performance and the proximity of the action to the desired outcome. For instance, actions that directly lead to a decrease in system errors receive higher rewards, while less impactful actions receive proportionately smaller rewards. This method ensures that the model progressively learns to favor actions that contribute most significantly to achieving predefined goals. We have incorporated detailed mathematical representations and operational scenarios of the reward function into the methodology section to provide clear insights into how rewards are calculated and distributed, thereby enhancing the transparency and reproducibility of our research.

5.2 Proposed System Performance

Table 2 presents the performance metrics of three deep learning models—CNN, Autoencoder, and LSTM—used for a binary classification task in intrusion detection. The CNN model outperforms the others with an accuracy of 0.9989, precision of 0.9990, recall of 0.9988, and an F1 score of 0.9989. The LSTM model follows closely, achieving an accuracy of 0.9985, precision of 0.9986, recall of 0.9984, and an F1 score of 0.9985. The Autoencoder, while slightly behind, still shows strong results with an accuracy of 0.9978, precision of 0.9979, recall of 0.9977, and an F1 score of 0.9978. Overall, the CNN model proves to be the most effective for this task due to its superior performance metrics.

Table 2: Comparison of the proposed deep learning model binary classification

Model	Accuracy	Precision	Recall	F1 score
CNN	0.9989	0.999	0.9988	0.9989
Autoencoder	0.9978	0.9979	0.9977	0.9978
LSTM	0.9985	0.9986	0.9984	0.9985

Table 3 presents the cross-validation performance metrics for three deep learning models—CNN, Autoencoder, and LSTM—applied to a classification task. Across all folds, the CNN model consistently achieves the highest performance metrics with accuracy ranging from 0.9958 to 0.9961, precision from 0.9960 to 0.9963, recall from 0.9958 to 0.9961, and F1 score from 0.9959 to 0.9962. The LSTM model follows closely, showing strong performance with accuracy between 0.9956 and 0.9959, precision from 0.9958 to 0.9961, recall from 0.9956 to 0.9959, and F1 score from 0.9957 to 0.9960. While demonstrating solid performance, the Autoencoder has slightly lower metrics with accuracy from 0.9948 to 0.9952, precision from 0.9949 to 0.9953, recall from 0.9948 to 0.9952, and F1 score from 0.9948 to 0.9952. CNN is the best-performing model, consistently achieving the highest scores across all evaluation metrics, making it the most effective model for this classification task.

Table 4 presents the performance metrics of three deep learning models—CNN, Autoencoder, and LSTM—used for a classification task. The CNN model achieves the best results, with an accuracy of 0.995, precision of 0.9952, recall of 0.995, and an F1 score of 0.9951, indicating its strong capability. The LSTM model is close behind, with an accuracy of 0.993, precision of 0.9933, recall of 0.993, and an F1 score of 0.9932. Although the Autoencoder ranks slightly lower, it still delivers solid performance, with an accuracy of 0.99, precision of 0.9905, recall of 0.99, and an F1 score of 0.9902. Overall, the CNN model outperforms the others, making it the most effective choice for this classification task.

Table 3: Cross-validation of the proposed deep learning model binary classification

Cross-validation	Evaluation metrics	CNN	Autoencoder	LSTM
Fold 1	Accuracy	0.996	0.995	0.9958
	Precision	0.9962	0.9951	0.996
	Recall	0.996	0.995	0.9958
	F1 score	0.9961	0.995	0.9959
Fold 2	Accuracy	0.9958	0.9948	0.9956
	Precision	0.996	0.9949	0.9958
	Recall	0.9958	0.9948	0.9956
	F1 score	0.9959	0.9948	0.9957
Fold 3	Accuracy	0.9961	0.9952	0.9959
	Precision	0.9963	0.9953	0.9961
	Recall	0.9961	0.9952	0.9959
	F1 score	0.9962	0.9952	0.996
Fold 4	Accuracy	0.9959	0.995	0.9957
	Precision	0.9961	0.9951	0.9959
	Recall	0.9959	0.995	0.9957
	F1 score	0.996	0.995	0.9958
Fold 5	Accuracy	0.996	0.9951	0.9958
	Precision	0.9962	0.9952	0.996
	Recall	0.996	0.9951	0.9958
	F1 score	0.9961	0.9951	0.9959

Table 4: Comparison of the proposed deep learning model multiclass classification

Model	Accuracy	Precision	Recall	F1 score
CNN	0.995	0.9952	0.995	0.9951
Autoencoder	0.99	0.9905	0.99	0.9902
LSTM	0.993	0.9933	0.993	0.9932

Table 5 presents a cross-validation performance analysis of three deep learning models—CNN, Autoencoder, and LSTM—for multiclass classification. The CNN model consistently delivers the highest metrics, with accuracy ranging from 0.9958 to 0.9961, precision from 0.9960 to 0.9963, recall from 0.9958 to 0.9961, and an F1 score from 0.9959 to 0.9962. The LSTM model follows closely, achieving accuracy between 0.9956 and 0.9959, precision from 0.9958 to 0.9961, recall from 0.9956 to 0.9959, and an F1 score from 0.9957 to 0.9960. The Autoencoder, while performing well, has slightly lower metrics, with accuracy from 0.9948 to 0.9952, precision from 0.9949 to 0.9953, recall from 0.9948 to 0.9952, and an F1 score from 0.9948 to 0.9952. Overall, the CNN model stands out as the most effective for this task, achieving the maximum performance across all metrics.

Table 5: Cross-validation of the proposed deep learning model multiclass classification

Cross-validation	Evaluation metrics	CNN	Autoencoder	LSTM
Fold 1	Accuracy	0.996	0.995	0.9958
	Precision	0.9962	0.9951	0.996
	Recall	0.996	0.995	0.9958
	F1 score	0.9961	0.995	0.9959
Fold 2	Accuracy	0.9958	0.9948	0.9956
	Precision	0.996	0.9949	0.9958
	Recall	0.9958	0.9948	0.9956
	F1 score	0.9959	0.9948	0.9957
Fold 3	Accuracy	0.9961	0.9952	0.9959
	Precision	0.9963	0.9953	0.9961
	Recall	0.9961	0.9952	0.9959
	F1 score	0.9962	0.9952	0.996
Fold 4	Accuracy	0.9959	0.995	0.9957
	Precision	0.9961	0.9951	0.9959
	Recall	0.9959	0.995	0.9957
	F1 score	0.996	0.995	0.9958
Fold 5	Accuracy	0.996	0.9951	0.9958
	Precision	0.9962	0.9952	0.996
	Recall	0.996	0.9951	0.9958
	F1 score	0.9961	0.9951	0.9959

6 Conclusion

The significance of this study lies in its demonstration of the reflective influence deep learning models can have on the efficacy of Intrusion Detection Systems (IDS) within IoT networks. Utilizing the UNSW-NB15 dataset, our research applied advanced preprocessing practices such as data cleaning, normalization, and feature selection, setting a robust foundation for subsequent analysis. The employment of CNN, Autoencoder, and LSTM models revealed that CNNs, in particular, consistently delivered superior performance across various classification tasks. Specifically, the CNN model accomplished an accuracy of 99.89%, precision of 99.90%, recall of 99.88%, and an F1 score of 99.89%, clearly demonstrating its capacity to capture complex network traffic patterns and substantially enhance intrusion detection accuracy.

A key achievement of this study was the successful implementation of a balanced dataset approach, which proved crucial in mitigating the risks associated with class imbalance. Traditional datasets often bias models towards majority classes, thus impairing the detection of less frequent, minority-class attacks. Our balanced approach ensured equitable class representation, which enhanced the models' generalization capabilities and robustness, enabling them to effectively identify a diverse spectrum of cyber threats, with performance improvements of up to 15%–20% in detecting minority-class intrusions compared to unbalanced datasets.

Despite these advancements, the study is not without limitations. One notable challenge was the inherent complexity of tuning deep learning models for optimal performance, which requires significant computational resources and expertise. Additionally, the generalizability of our findings may be limited by the specific nature of the UNSW-NB15 dataset, which, while comprehensive, may not capture all types of network behaviors and attack vectors encountered in real-world IoT environments. Future research should focus on several key areas to address these limitations and further the efficacy of IDS in IoT. Firstly, exploring advanced optimization methods could enhance model training efficiency and effectiveness. Secondly, developing hybrid models that combine the strengths of CNNs with other machine learning procedures could provide a more holistic approach to threat detection. Finally, testing the models against a wider array of datasets could improve their adaptability and applicability across diverse IoT scenarios, ultimately contributing to a more secure and resilient IoT ecosystem.

Acknowledgement: The authors sincerely acknowledge the Princess Sumaya University for Technology for supporting steps of this work.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Ammar Odeh: Writing—original draft, conceptualization, methodology, software, validation. Anas Abu Taleb: Writing—review & editing, visualization, validation, supervision, software, project administration. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data openly available in a public repository. The data that support the findings of this study are openly available in UNSW-NB15 at (<https://research.unsw.edu.au/projects/unsw-nb15-dataset>) (accessed on 04 November 2024).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

- [1] A. Rejeb *et al.*, “The Internet of Things (IoT) in healthcare: Taking stock and moving forward,” *Internet Things*, vol. 22, 2023, Art. no. 100721 doi: [10.1016/j.iot.2023.100721](https://doi.org/10.1016/j.iot.2023.100721).
- [2] A. A. Laghari, K. Wu, R. A. Laghari, M. Ali, and A. A. Khan, “A review and state of art of Internet of Things (IoT),” *Arch. Comput. Methods Eng.*, vol. 29, no. 3, pp. 1395–1413, 2021. doi: [10.1007/s11831-021-09622-6](https://doi.org/10.1007/s11831-021-09622-6).
- [3] M. Thakur, “Cyber security threats and countermeasures in digital age,” *J. Appli. Sci. Educ.*, vol. 4, no. 1, pp. 1–20, 2024. doi: [10.54060/a2zjournals.jase.42](https://doi.org/10.54060/a2zjournals.jase.42).
- [4] A. Alaa Hammad, M. Adnan Falih, S. Ali Abd, and S. Rashid Ahmed, “Detecting cyber threats in IoT networks: A machine learning approach,” *Int. J. Comput. Digit. Syst.*, vol. 16, pp. 1–16, 2024.
- [5] V. M. Sivagami, K. Kiruthika Devi, V. Vidhya, and S. Swarna Parvathi, “Threat Models and Attack Strategies in the Internet of Things,” in *Secure Communication in Internet of Things*, Boca Raton, FL, USA: CRC Press, pp. 253–265.
- [6] M. R. Shaffique, “Cyber resilience act 2022: A silver bullet for cybersecurity of IoT devices or a shot in the dark?,” *Comput. Law Secur. Rev.*, vol. 54, 2024, Art. no. 106009. doi: [10.1016/j.clsr.2024.106009](https://doi.org/10.1016/j.clsr.2024.106009).
- [7] S. Rani, A. Sharma, and M. Zohaib, “Study for integrating IoT-IDS datasets: Machine and deep learning for secure IoT network system,” in *Proc. 28th Int. Conf. Eval. Assess. Softw. Eng.*, 2024, pp. 686–691.

- [8] M. Al-Ambusaidi, Y. J. Zhang, Y. Muhammad, and A. Yahya, "ML-IDS: An efficient ML-enabled intrusion detection system for securing IoT networks and applications," *Soft Comput.*, vol. 28, no. 2, pp. 1765–1784, 2024. doi: [10.1007/s00500-023-09452-7](https://doi.org/10.1007/s00500-023-09452-7).
- [9] K. Shalabi, Q. A. Al-Haija, and M. Al-Fayoumi, "A blockchain-based intrusion detection/prevention systems in IoT network: A systematic review," *Procedia Comput. Sci.*, vol. 236, no. 4, pp. 410–419, 2024. doi: [10.1016/j.procs.2024.05.048](https://doi.org/10.1016/j.procs.2024.05.048).
- [10] Q. A. Al-Haija, S. Altamimi, and M. AlWadi, "Analysis of extreme learning machines (ELMs) for intelligent intrusion detection systems: A survey," *Expert. Syst. Appl.*, vol. 253, no. 2, 2024, Art. no. 124317. doi: [10.1016/j.eswa.2024.124317](https://doi.org/10.1016/j.eswa.2024.124317).
- [11] I. Ullah, I. U. Khan, M. Ouaisa, M. Ouaisa, and S. El Hajjami, *Future Communication Systems Using Artificial Intelligence, Internet of Things and Data Science*, 1st ed. Boca Raton, FL, USA: CRC Press, 2024.
- [12] M. Khalil, Q. A. Al-Haija, and S. Ahmad, "Healthcare IoT networks using LPWAN," in *Low-Power Wide Area Network for Large Scale Internet of Things*. Boca Raton, FL, USA: CRC Press, 2024, pp. 203–216.
- [13] E. Altulaihah, M. A. Almaiah, and A. Aljughaiman, "Anomaly detection IDS for detecting DoS attacks in IoT networks based on machine learning algorithms," *Sensors*, vol. 24, no. 2, 2024, Art. no. 713. doi: [10.3390/s24020713](https://doi.org/10.3390/s24020713).
- [14] S. Muneer, U. Farooq, A. Athar, M. Ahsan Raza, T. M. Ghazal and S. Sakib, "A critical review of artificial intelligence based approaches in intrusion detection: A comprehensive analysis," *J. Eng.*, vol. 2024, no. 3, 2024, Art. no. 3909173. doi: [10.1155/2024/3909173](https://doi.org/10.1155/2024/3909173).
- [15] U. I. Okoli, O. C. Obi, A. O. Adewusi, and T. O. Abrahams, "Machine learning in cybersecurity: A review of threat detection and defense mechanisms," *World J. Adv. Res. Rev.*, vol. 21, no. 1, pp. 2286–2295, 2024. doi: [10.30574/wjarr.2024.21.1.0315](https://doi.org/10.30574/wjarr.2024.21.1.0315).
- [16] H. Arif, A. Kumar, M. Fahad, and H. K. Hussain, "Future horizons: AI-enhanced threat detection in cloud environments: Unveiling opportunities for research," *Int. J. Multidiscip. Sci. Arts*, vol. 3, no. 2, pp. 242–251, 2024. doi: [10.47709/ijmdsa.v2i2.3452](https://doi.org/10.47709/ijmdsa.v2i2.3452).
- [17] P. T. Quinlan, "The visual detection of threat: A cautionary tale," *Psychono. Bull. Rev.*, vol. 20, no. 6, pp. 1080–1101, 2013. doi: [10.3758/s13423-013-0421-4](https://doi.org/10.3758/s13423-013-0421-4).
- [18] J. Tipples, A. W. Young, P. Quinlan, P. Brooks, and A. W. Ellis, "Searching for threat," *Q. J. Exp. Psychol. Sec. A*, vol. 55, no. 3, pp. 1007–1026, 2002. doi: [10.1080/02724980143000659](https://doi.org/10.1080/02724980143000659).
- [19] J. Verma, A. Bhandari, and G. Singh, "iNIDS: SWOT analysis and TOWS inferences of state-of-the-art NIDS solutions for the development of intelligent network intrusion detection system," *Comput. Commun.*, vol. 195, pp. 227–247, 2022. doi: [10.1016/j.comcom.2022.08.022](https://doi.org/10.1016/j.comcom.2022.08.022).
- [20] A. N. Lone, S. Mustajab, and M. Alam, "A comprehensive study on cybersecurity challenges and opportunities in the IoT world," *Secur. Priv.*, vol. 6, no. 6, 2023, Art. no. e318. doi: [10.1002/spy2.318](https://doi.org/10.1002/spy2.318).
- [21] M. Mayuranathan, M. Murugan, and V. Dhanakoti, "Retracted article: Best features based intrusion detection system by RBM model for detecting DDoS in cloud environment," *J. Ambient Intell. Humaniz. Comput.*, vol. 12, no. 3, pp. 3609–3619, 2021. doi: [10.1007/s12652-019-01611-9](https://doi.org/10.1007/s12652-019-01611-9).
- [22] R. SaiSindhuTheja and G. K. Shyam, "An efficient metaheuristic algorithm based feature selection and recurrent neural network for DoS attack detection in cloud computing environment," *Appl. Soft Comput.*, vol. 100, no. 1, 2021, Art. no. 106997. doi: [10.1016/j.asoc.2020.106997](https://doi.org/10.1016/j.asoc.2020.106997).
- [23] M. H. Qais, H. M. Hasanien, and S. Alghuwainem, "Transient search optimization: A new metaheuristic optimization algorithm," *Appl. Intell.*, vol. 50, no. 11, pp. 3926–3941, 2020. doi: [10.1007/s10489-020-01727-y](https://doi.org/10.1007/s10489-020-01727-y).
- [24] E. Kfoury, J. Saab, P. Younes, and R. Achkar, "A self organizing map intrusion detection system for RPL protocol attacks," *Int. J. Interdiscip. Telecomm. Netw. (IJITN)*, vol. 11, pp. 30–43, 2019. doi: [10.4018/IJITN](https://doi.org/10.4018/IJITN).
- [25] N. Waheed, X. He, M. Ikram, M. Usman, S. S. Hashmi and M. Usman, "Security and privacy in IoT using machine learning and blockchain: Threats and countermeasures," *ACM Comput. Surv. (csur)*, vol. 53, pp. 1–37, 2020.

- [26] J. Li, M. S. Othman, H. Chen, and L. M. Yusuf, "Optimizing IoT intrusion detection system: Feature selection versus feature extraction in machine learning," *J. Big Data*, vol. 11, no. 1, 2024, Art. no. 36. doi: [10.1186/s40537-024-00892-y](https://doi.org/10.1186/s40537-024-00892-y).
- [27] S. A. Khanday, H. Fatima, and N. Rakesh, "A novel data preprocessing model for lightweight sensory IoT intrusion detection," *Int. J. Math. Eng. Manag. Sci.*, vol. 9, pp. 188–204, 2024. doi: [10.33889/24557749](https://doi.org/10.33889/24557749).
- [28] J. Ali, H. H. Song, V. Sharma, and M. A. Al-Khasawneh, "DDoS intrusions detection in low power SD-IoT devices leveraging effective machine learning," in *IEEE Trans. Consum. Electron.*, 2024. doi: [10.1109/TCE.2024.3472707](https://doi.org/10.1109/TCE.2024.3472707).
- [29] R. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Mil. Commun. Inform. Syst. Conf. (MilCIS)*, Canberra, ACT, Australia, 2015, pp. 1–6. doi: [10.1109/MilCIS.2015.7348942](https://doi.org/10.1109/MilCIS.2015.7348942).