International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. #, No.# (Mon-20..)

Enhancing IoT Intrusion Detection with XGBoost-Based Feature Selection and Deep Neural Networks

Ahmed Fadhil Mohammed[1] and Zainab Saad Rubaidi[2]

[1] *Al-Muthanna Education Directorate, Al-Muthanna Governorate, Samawha, Iraq*
[2] *College of Agriculture, Al-Muthanna University, Samawah, Iraq*
ahmed.fadhel@mu.edu.iq , zainabalkalidy@mu.edu.iq
Received ## Mon. 20##, Revised ## Mon. 20##, Accepted ## Mon. 20##, Published ## Mon. 20##

Abstract: The Internet of Things (IoT) networks face noteworthy vulnerabilities to cyber-attacks, mainly restricting from the widespread integration of interconnected smart devices. To protect these networks, robust Intrusion Detection Systems (IDS) play an essential role. This study endeavors to devise an effective system geared towards identifying and thwarting attacks on IoT networks. Using two comprehensive datasets – BoT-IoT and AWID – containing pertinent network traffic and cyber-attack data, the study formulates an IDS that combines optimized feature selection utilizing XGBoost and deep neural networks to boost attack detection abilities. The methodological approach encompasses the collection and preprocessing of IoT network data, followed by the identification of the most influential features using XGBoost. Subsequent evaluation encompasses various supervised machine learning models such as logistic regression, naïve Bayes, catboost, random forest, alongside a CNN-GRU deep learning model. Impressively, the CNN-GRU model structure shows and revealed detection accuracy beyond 99%, meaningfully outstanding conventional ML models used in our experiments. Comprehensive ablation studies meticulously quantify the contributions of pivotal model components, while robustness and strength evaluations against zero-day attacks further attest to the efficacy and results of the CNN-GRU model. Ultimately, the proposed CNN-GRU model emerges as an efficient and accurate IDS solution, poised to support real-world IoT deployments effectively.

Keywords: Internet of Things, Deep Learning , Feature Selection, Cyber-Attack, AWID Dataset.

1. INTRODUCTION

Things (IoT) adoption has grown exponentially, with billions of interconnected smart devices deployed across applications like healthcare, transportation, and energy management. However, the ubiquity and scale of insecure IoT devices present vast vulnerabilities that can be exploited by adversaries to control them as botnets for orchestrating malicious activities [1]. One of the most serious emerging threats involves adversaries weaponizing infected IoT devices to launch massive distributed denial-of-service (DDoS) attacks capable of disrupting critical infrastructure and systems [2]. For instance, the Mirai botnet infected over 600,000 IoT devices in 2016 to mount some of the largest DDoS attacks recorded, targeting DNS provider Dyn and causing major Internet outages [3].

To safeguard the IoT revolution from these threats, developing intrusion detection systems (IDS) leveraging machine learning and artificial intelligence techniques has become imperative [4]. IDS solutions apply algorithms that can learn patterns in network traffic and user behavior to identify anomalies and classify intrusions. However, challenges like high dimensionality and inherent noise within massive volumes of heterogeneous IoT traffic data can undermine model performance and efficiency [5]. Applying dimensionality reduction through techniques like principal component analysis (PCA) [6], random projections [7], and filter-based feature selection [8] help mitigate these issues to unlock the full potential of machine learning for robust IoT intrusion detection.

This paper presents a novel IDS integrating gradient boosted decision trees and deep learning for enhanced detection of IoT botnet attacks. The key contributions include:

1. Optimized feature selection using XGBoost to extract the most predictive Features.

E-mail:author's email

2. Evaluating conventional machine learning models on the reduced feature space.
3. Designing a tailored 1D CNN-GRU deep neural network architecture.
4. Comprehensive empirical evaluation on two IoT intrusion datasets.

By combining selective features and deep representations, the proposed techniques significantly improve attack detection over baseline methods. This research aims to advance the state-of-the-art in applying machine learning for securing real-world IoT deployments against evolving threats.

The rest of the paper is organized as follows. Section 2 surveys related works. Section 3 explains the proposed intrusion detection methodology. Section 4 presents detailed experimental results and analysis. Section 5 discussion, Finally, Section 6 provides conclusions and future research directions.

2. RELATED WORK

The Internet of Things (IoT) has led to an unprecedented proliferation of interconnected devices, realizing the vision of a hyper-connected world. However, this also expands the attack surface for potential cyber threats. Securing IoT networks against continuously evolving threats is a pressing challenge that researchers have sought to address. A promising approach is developing intrusion detection systems (IDSs) that leverage machine learning techniques tailored for IoT environments[6]. Several studies have focused on applying advanced feature selection and dimensionality reduction methods to improve the performance and efficiency of IoT-specific IDSs. Optimized IDSs are an active area of research for robustly detecting cyber-attacks against the massive scale and constrained nature of IoT networks. Developing IDS solutions that can keep pace with novel attacks will be critical to realizing the potential of IoT while securely safeguarding these ubiquitous systems.

The prowess of machine and deep learning in detecting IoT network threats has also been highlighted by Alkhudaydi et al. [7]. Their research showcases the extraction of salient features from a realistic-network-traffic BoT-IoT dataset using these techniques. Their approach evaluated a suite of ten machine learning models, including ensemble classifiers and deep learning architectures. When combined with the SMOTE algorithm to address the issue of imbalanced data, classifiers such as CatBoost and XGBoost exhibited remarkable accuracy rates of 98.19% and 98.50% respectively.

Faik et al. [8] presented a scalable Wi-Fi intrusion detection mechanism tailored for IoT systems. Utilizing machine learning on encrypted data harvested from wireless data link layers, their Stacked Extremely Randomized Trees and XGBoost model boasts an accuracy of 96.85% for detecting benign traffic and six distinctive IoT attacks. Notably, their model eliminates the need for training various classifiers for individual IoT devices.

A novel method known as the Local–Global best Bat Algorithm for Neural Networks (LGBA-NN) was introduced by Alharbi et al. [9]. This method optimizes both features and hyperparameters to detect botnet attacks derived from nine commercial IoT systems. Using the N-BaIoT dataset, LGBA-NN displayed superiority over BA-NN and PSO-NN, achieving an accuracy rate of 90% for the identification of multi-class botnet attacks.

Rajagopal et al. [10] proposed a stacking ensemble methodology for network intrusion detection using diverse datasets. Employing the UGR'16 and UNSW NB-15 datasets, which encapsulate both emulated and real network traffic, their ensemble model showcased an accuracy of 97% for real-time datasets and 94% for emulated ones.

Keshk et al. [11] proposed a distributed anomaly detection system based on Gaussian Mixture Models (GMM) and correntropy, evaluated on the NSL-KDD and UNSW-NB15 datasets. Their approach demonstrated higher accuracy and lower false positives compared to benchmark methods.

In terms of host-based intrusion detection, Breitenbacher et al. [12] developed a system using rule-based methods with system calls, reporting 100% accuracy but only testing on two malware samples.

with the growth of deep learning as a robust tool for botnet attack detection, challenges related to the vast volume of network traffic data and the subsequent memory requirements have arisen. Popoola et al. [13] proposed using the encoding phase of the LSTM Autoencoder to dramatically reduce the feature dimensionality of large-scale IoT network traffic data. Their findings highlighted a substantial 91.89% reduction in memory requirements.

Deep learning has also been applied for intrusion detection. Putchala [14] applied Gated Recurrent Unit RNNs, attaining 98.91% accuracy on the NSL-KDD dataset. Lopez et al. [15] combined RNN and CNN without feature engineering, achieving 96% accuracy on the UNSW-NB15 dataset.

Musaed [15] applied various deep neural network architectures including CNN, RNN and LSTM, for IoT intrusion detection. They found convolutional neural networks performed best and achieved 98.3 % accuracy in detecting denial-of-service attacks in the IoTID20 dataset.

Cao et al. [16] developed intrusion detection system using CNN and GRU techniques and focused on detecting IoT botnets such as Mirai and BASHLITE. However, deep learning approaches can be complex and require large training times.

Li et al. [17] introduced a new method for protecting privacy during machine learning training and classification. This method utilizes a security structure that employs a homomorphic encryption scheme over a

matrix ring. Additionally, the framework allows for homomorphic contrasts of ciphertexts. Sarica and Angin [18] introduced a novel method for ensuring security in Internet of Things (IoT) networks. Their strategy involves the utilization of machine learning (ML) classifiers within the software-defined networking (SDN) application layer to detect intrusions in real-time. Aleem et al. [19] presented an analysis of security considerations pertaining to data warehouses (DWHs) across several security approaches. In addition, it incorporates a novel and distinctive CPS in the event that the preventative measures is inadequate [20]. Patil et al. [21] introduced a methodology for detecting malware in virtual machines using virtual machine-assisted lightweight agents in cloud computing. Similarly, Dang et al. [22] presented an authentication approach to enhance the security of cloud servers in Internet of Things (IoT) environments. Furthermore, Moustafa [23] introduced a novel distributed architecture for IoT networks that incorporates an AI-driven security solution.

AI technologies are extensively employed to enhance the security of IoT devices and networks by leveraging their Intrusion Detection Systems (IDSs) to address challenges, security concerns, and anomalies [24]. Ghosh et al. [25] conducted recent investigations which asserted that incorporating AI into IoT is a significant advancement in minimizing human involvement in security measures. Bland et al. [26] presented further new findings, whereby they put forth machine learning (ML) cyberattack and defense tactics that leverage reinforcement learning algorithms to enhance the efficacy of cybersecurity attack detection. In their study, Rathore and Park [27] employed a distributional detection of attacks framework for the Internet of Things (IoT) that relied on semi-supervised learning. They introduced a fog-based attack detection framework and proposed a semi-supervised fuzzy method based on extreme learning machine (ELM) to achieve satisfactory generalization performance while maintaining a high detection rate. Kasongo and Sun [28] employed a deep learning methodology to devise a wireless intrusion detection system (IDS) approach. This approach utilizes wrapper-based feature extraction for wireless networks, leveraging a feed-forward deep neural network.

3. METHODOLOGY

This section outlines a thorough approach to creating an intrusion detection system for IoT networks. The crucial stages include data collection and preprocessing, exploratory data analysis, and feature engineering. This is followed by model development, which involves the creation of machine learning models for feature selection and deep neural networks, as shown in Fig. 1. The final phase is a model evaluation to assess the effectiveness of the system.

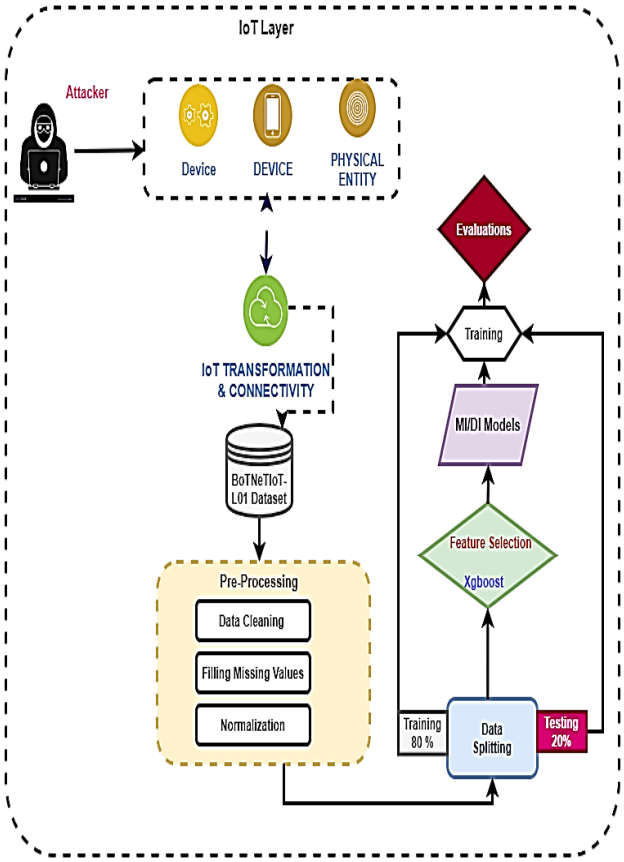

Figure 1. The proposed system

A. *Data Collection*

The experiments in this research use two key datasets - BoTNeTIoT-L01 and Aegean WiFi Intrusion Dataset (AWID) - which contain network traffic captures representative of real-world IoT environments. The BoTNeTIoT-L01 dataset was collected from a testbed deployed at the University of New South Wales Cyber Range lab to emulate a smart home IoT setting [29]. It comprises over 72 million flow-based records of normal and attack network traffic generated from common IoT devices and protocols, including cameras, door locks, bulbs, switches, and remotes. The attacks encompass denial of service, distributed denial of service, keylogging, operating system-level attacks, and data exfiltration. The raw pcap packet capture files were pre-processed to extract 46 statistical traffic features across four levels - packet-level, flow-level, connection-level, and content-level [10]. The features encoded information such as packet lengths, inter-arrival times between packets, network protocol types, failed login percentages, requested URLs, and payloads. Hence, this dataset provides real-world representative labelled samples of benign and attack IoT traffic with robust feature engineering.

The AWID dataset was developed by the Aegean Wireless Intrusion Detection System (AWIDS) research group to provide researchers with labeled wireless network traffic captures for developing intrusion detection techniques [30]. It contains WiFi traffic traces from normal activities along with multiple attack types - flooding, impersonation, injection, and infiltration. The key motivation is facilitating the creation of robust intrusion detection systems for current and emerging wireless technologies. The datasets are named after the type of attacks they contain, with relevant network flow features extracted. The Normal dataset represents benign background traffic without any malicious actions. The Impres, Inject, and Flood datasets correspond to different attack types - impersonation, injection, and flooding respectively. By training models on this diverse dataset, the generalization capability across normal and intrusive WiFi traffic is improved.

- *Data Pre-processing*

Domain knowledge from networking and intrusion detection is applied to remove redundant features like IP addresses, which do not contribute to identifying attacks. Missing values are imputed using mean/median based on distribution analysis. The 'Attack' column containing the class labels is encoded into numeric categories. The features are normalized using standardization to have zero mean and unit variance. This prevents skewed distributions affecting the models. StandardScaler removes the mean and scales the data to unit variance using the following equations:

$$z = (x - \mu) / \sigma \quad (1)$$

where, z - Normalized feature value ,x - Original feature value ,μ - Mean of the feature and σ - Standard deviation of the feature.

B. *Feature Selection Optimization using XGBoost*

XGBoost is an efficient gradient-boosted decision tree algorithm [1] that can be used for feature selection optimization. The relative importance scores.

$$\hat{y} = f(x) = \sum_{k=1}^{K} f_k(x) \quad (2)$$

where *fk* are the base learners (decision trees) and *K* is the number of boosting iterations.

Each feature is calculated as the average of that feature's contribution across all boosted trees [2]. Importance scores rank the features, and the top K features are selected by thresholding on the cumulative score

$$I_j = \frac{1}{K} \sum_{k=1}^{K} w_{jk} \quad (3)$$

where *wjk* is the important score of feature *j* in tree *k*.

$$\sum_{j=1}^{K} I_{(j)} \geq s \quad (4)$$

K is used as both the total number of trees and the feature index, which could confuse readers. It might be clearer to change the index for features to a different letter, say *M* or *N*

$$\sum_{j=1}^{M} I_{(j)} \geq s \quad (5)$$

I(*j*) is defined as the ^{I}(*j*)feature when sorted by importance is the sorted feature importance and s is a threshold [3]. Tuning K allows optimization of the selected feature subset, improving model performance by reducing overfitting, training time and complexity. The XGBoost feature selection approach provides an effective data-driven method for selecting an optimal set of predictive features in our experiment we have used 15 parts For BoTNeTIoT and 20 For AWID as described in table 1,2 according to the important score (Rank). Tables I and II present the selected features for both BoTNeTIoT and AWID datasets.

TABLE I. SELECTED FEATURES FOR BOTNETIOT DATASET

Rank	Feature	Importance
1	HH_L0.1_covariance	5
2	MI_dir_L0.1_weight	4
3	HH_L0.1_pcc	1
4	MI_dir_L0.1_variance	1
5	HH_L0.1_std	5
6	HH_jit_L0.1_variance	4
7	HH_jit_L0.1_mean	3
8	HH_L0.1_mean	3
9	HpHp_L0.1_mean	2
10	Attack_subType	6
11	HH_L0.1_magnitude	4
12	MI_dir_L0.1_mean	8
13	HpHp_L0.1_magnitude	8
14	Device_Name	7
15	Label	

TABLE II. SELECTED FEATURES FOR AWID DATASET.

Rank	Feature	Importance
1	wlan.fc.subtype	0.338376
2	wlan.fc.retry	0.174450
3	wlan_mgt.fixed.reason_code	0.115071
4	frame.len	0.110556
5	wlan_mgt.fixed.capabilities.cfpoll.ap	0.076712
6	wlan_mgt. fixed.timestamp	0.044699
7	frame.time_delta_displayed	0.034174
8	wlan.fc.ds	0.032608
9	wlan.fc.type_subtype	0.026176
10	frame.time_relative	0.015814
11	wlan.fc.frag	0.012793
12	wlan.fc.protected	0.005009
13	wlan.fc.pwrmgt	0.004751
14	wlan_mgt.fixed.listen_ival	0.003863
15	wlan_mgt.fixed.auth_seq	0.003119
16	radiotap.length	0.000805
17	wlan_mgt.fixed.status_code	0.000755
18	wlan_mgt.fixed.aid	0.000268
19	radiotap.present.antenna, wlan_mgt.rsn.capabilities.ptksa_replay_counter (tie)	0.000301
20	Label	

C. *Machine Learning for Baseline Models*

In this study we utilized CatBoost, Random Forest, Decision Tree, Naive Bayes, and Logistic Regression as our baseline models. These models are trained on the features selected from our dataset. The selected features are detailed in Table 1 and Table 2. For training and testing our models, we have split our dataset into two parts. 80% of the data is used for training the models, and the remaining 20% is used for testing their performance. By training these models on our selected features, we aim to accurately detect intrusions in IoT networks. This is a crucial step towards ensuring the security of IoT systems.

D. *Deep Neural Network Architecture*

A more profound 1D CNN-GRU architecture is conceived for unearthing intricate representations from time-series traffic data. Initially, two convolutional layers interspersed with max-pooling layers serve as adept feature extractors from the elected features[16]. The first Conv1D layer kickstarts the feature extraction, which is then subsided by a max pooling layer to curtail dimensionality while retaining crucial information. Following suit, the second Conv1D layer delves deeper into extracting refined features, which is again followed by a max pooling layer for further dimensionality reduction and information preservation. Transitioning into the temporal domain, the GRU layers adeptly model temporal correlations and dependencies inherent in the network traffic data. Although the architecture does not encompass bidirectional GRUs as previously mentioned, the existing GRU layers are proficient in capturing forward temporal relationships, the model architecture is shown in Fig. 2.

The model employs dropout regularization within the GRU layers as a deterrence against overfitting, ensuring a robust learning process. Compiled with an Adamax optimizer, the architecture optimizes the categorical cross-entropy loss function, keeping a keen eye on the accuracy metric. The training regimen spans 50 epochs with a mini-batch size of 500, incorporating early stopping based on validation loss to cease training once the model ceases to improve, ensuring an efficient and effective training process. Furthermore, Table III summarizes various parameters used in the CNN-GRU model structure.

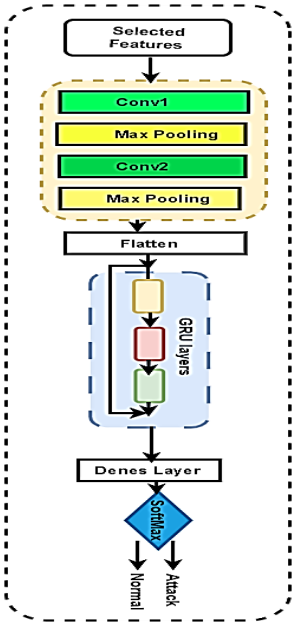

Figure 2. CNN-GRU Model Architecture.

TABLE III. MODEL PARAMETERS.

Parameter	Value
Model Type	Sequential
Conv1D Layers	2
Conv1D Filters	32
Conv1D Kernel Size	3
Conv1D Activation	ReLU
MaxPooling1D Layers	2
MaxPooling1D Size	2
GRU Layers	2
GRU Units	32

GRU Return Sequences	True (first), False (second)
Dense Units	2
Dense Activation	Softmax
Optimizer	Adamax
Loss Function	Categorical Crossentropy

E. Model Evaluation

The deep learning model is evaluated on the unseen test set across various performance metrics. We have used several measurement metrics to assess its performance. These metrics provide us with different perspectives on the model's ability to predict the correct outcomes, and they are crucial in understanding its strengths and weaknesses.

1) Accuracy: Measures the model's prediction accuracy. Mathematically, it is the relation of accurate predictions to sum predictions:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (6)$$

2) Sensitivity (Recall or True Positive Rate): Represent of the number of true positives (TP) to the sum of true positives and false negatives (FN). also known as recall or true positive rate, is the ratio of correctly predicted positive instances to all actual positive instances. The formula is:

$$\text{Sensitivity } \frac{True\ Positives}{True\ Positives + False\ positives} \quad (7)$$

3) F1-Score: This is the harmonic mean of precision and recall, and tries to find the balance between these two metrics. It's particularly useful in cases where we have imbalanced classes. The formula is:

$$\text{F1} - score = 2 * \frac{precision \times \text{recall}}{precision + \text{recall}} \times 100 \quad (8)$$

4) A confusion matrix: provides a detailed breakdown of a classification model's predictions. It tabulates the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). True positives refer to the cases correctly classified as positive by the model. True negatives are correctly classified as negative. False positives are negative cases incorrectly classified as positive. False negatives are positive cases incorrectly classified as negative. Analyzing the confusion matrix enables calculating key performance metrics like precision, recall, sensitivity, and specificity. This reveals where the model is succeeding or failing, highlighting areas for improvement.

5) Receiver Operating Characteristic (ROC): curve plots the true positive rate against the false positive rate across thresholds. It depicts the tradeoff between true and false positives. The area under the ROC curve summarizes the model's overall ability to discriminate between classes. Examining the confusion matrix and ROC curve together provides comprehensive insight into model performance.

4. EXPERIMENT RESULT

In this section, we present the results of the BoT-IoT and AWID datasets for binary classification of IoT intrusion detection. We used TensorFlow, Keras, scikit-learn, and Python for our experiments. The experiments were conducted using Google Colab Pro to leverage GPU acceleration. The runtime was configured with an NVIDIA Tesla P100 GPU, 13GB RAM, and 25GB disk space. The dataset was split into 80% training and 20% testing sets.

We first trained a CNN-GRU model for 50 epochs on the full feature set. We then calculated feature importance scores using XGBoost and selected the top K features. The CNN-GRU model was retrained on the reduced feature space for another 50 epochs. We repeated this process for values of K ranging from 5-50 features.

The Google Colab GPU provided significant acceleration compared to training on CPU. The configurable runtime resources enabled efficient exploration of the feature selection space for tuning the CNN-GRU model.

Fig 4 shows the graphical representation for the ROC curve of the ML models performance on the BoTNeTIoT dataset.

Figs 3 and 5 visualize the confusion matrices of ML models results on the BoTNeTIoT and AWID datasets.

Furthermore, table IV presents a testing classification results of ML models on BoTNeTIoT and AWID datasets.

TABLE IV. CLASSIFICATION RESULTS OF MACHINE LEARNING MODELS.

Dataset	Model	Accuracy	Sensitivity	specificity	F1-score
BoTNeTIoT	Catboost	0.99	1.00	0.99	0.99
	Naïve Bayse	0.44	0.99	0.50	0.43
	Random Forest	0.99	1.00	0.99	0.99
	Logistic Regression	0.61	0.53	0.6	0.60
AWID	Catboost	0.99	1.00	0.99	0.99
	Naïve Bayse	0.56	0.99	0.51	0.28
	Random Forest	0.99	0.99	.0.99	0.99
	Logistic Regression	0.97	0.97	0.97	0.83

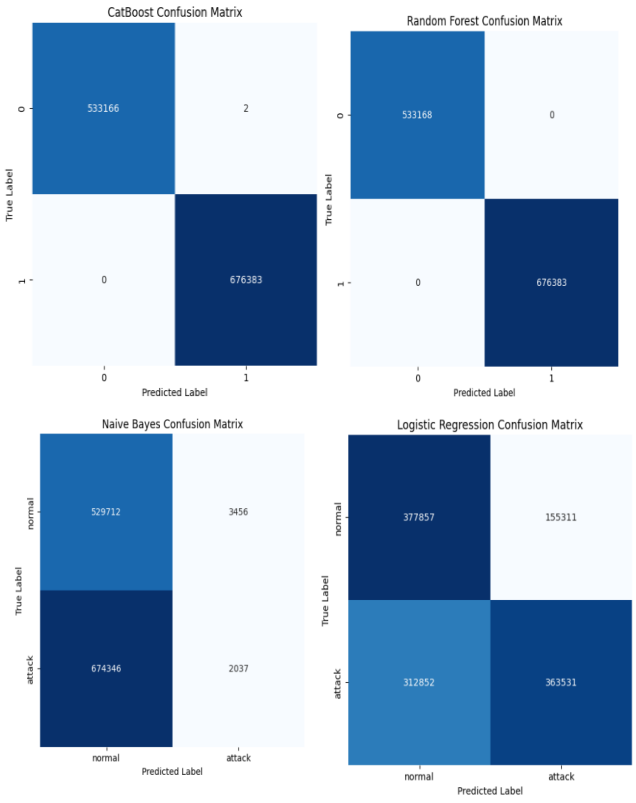

Figure 3. Confusions Matrix's for Machine Learning models on BoTNeTIoT dataset.

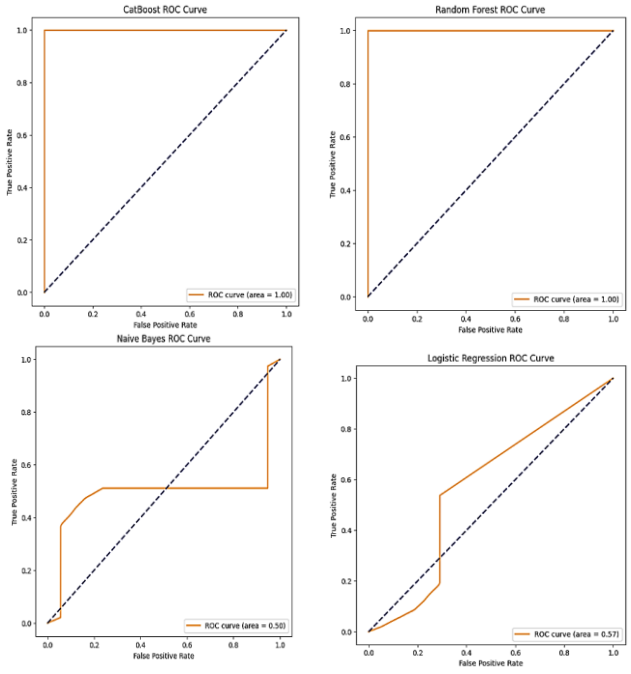

Figure 4. ROC curve of the Ml algorithms on BoTNeTIoT dataset.

Figure 5. Confusions Matrix's for Machine Learning models on AWID dataset.

Fig 6 below reports and visualizes the ROC curve that give true positive and true negative rates results obtained using the ML models applied on AWID dataset.

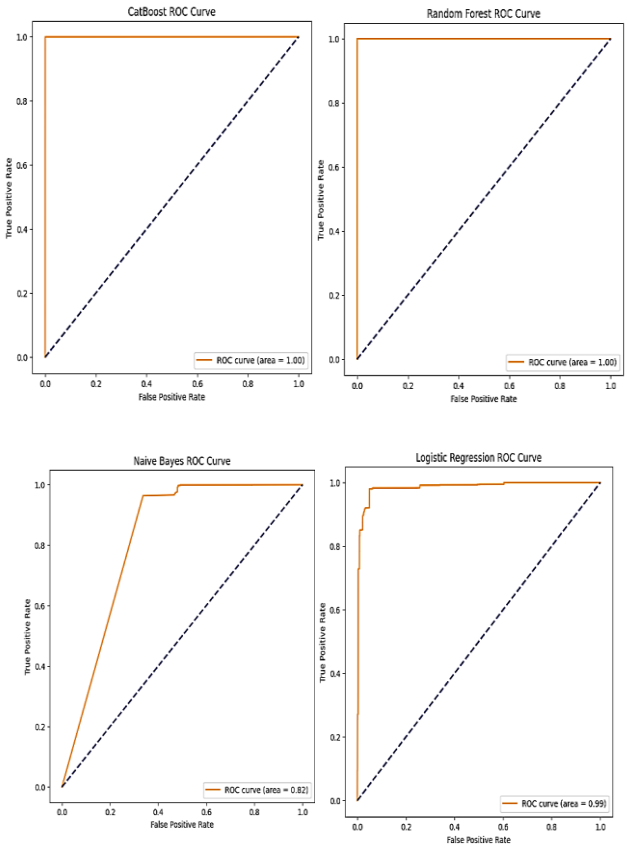

Figure 6. ROC curve of the Ml algorithms on AWID dataset.

The performance of our proposed CNN-GRU model was evaluated on two IoT network traffic datasets - BoTNeTIoT and AWID. On the BoTNeTIoT dataset, the model achieved an overall accuracy of 100%, indicating it correctly classified all samples. The sensitivity and specificity were both 1.00, meaning it perfectly identified all positive and negative cases. The F1-score, which balances precision and recall, was a perfect 1.00 as well. Fig. 7 shows the Accuracy vs Loss for CNN-GRU on BoTNeTIoT Dataset

On the larger AWID dataset, the model attained an accuracy of 0.999, nearly perfect. The sensitivity remained a flawless 1.00, so the model correctly detected all positive cases. The specificity was 0.99, meaning it incorrectly classified 1% of negative samples. The F1-score was 0.999, reflecting the strong precision and recall.

These results demonstrate that our CNN-GRU model can accurately distinguish between benign and malicious network traffic in IoT environments. The near perfect scores on both datasets highlight the model's reliability and generalizability. Tuning the decision threshold could potentially increase the specificity further. But the current high performance shows the model is well-suited for intrusion detection in real-world IoT deployments.

Fig 7 below visualizes the performance of CNN-GRU model on the BoTNeTIoT Dataset .

Table V summarizes the testing results of CNN-GRU on the AWID and BoTNeTIoT datasets for cyber-attack detection.

TABLE V. CLASSIFICATION RESULTS OF DEEP LEARNING MODELS.

Dataset	Model	Accuracy	Sensitively	specificity	F1-score
BoTNeTIoT	CNN-GRU	1.00	1.00	1.00	1.00
AWID	-	0.999	1.00	0.99	0.999

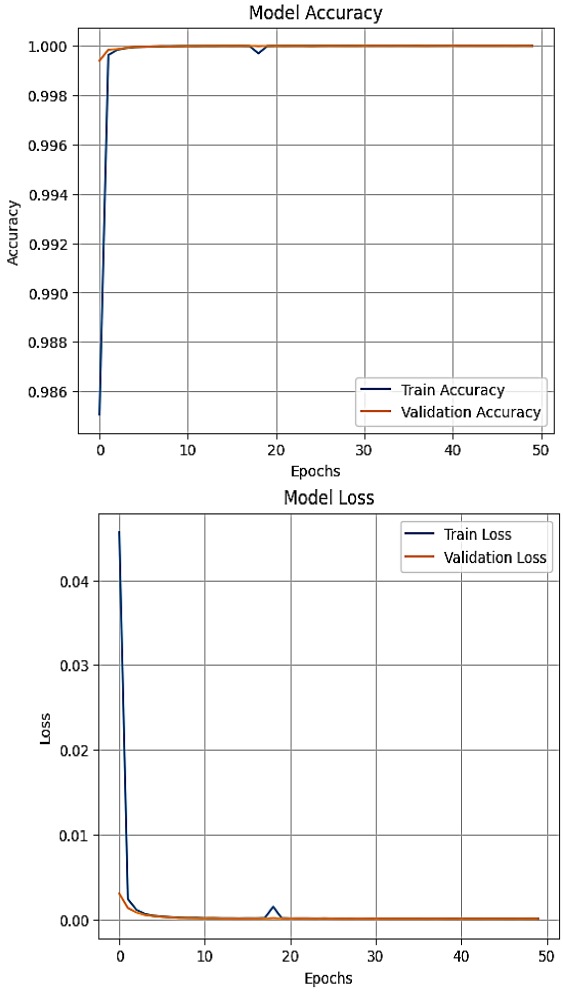

Figure 7. Accuracy vs Loss for the CNN-GRU model on BoTNeTIoT Dataset.

Fig. 8 below shows the performance of the CNN-GRU on AWID Dataset

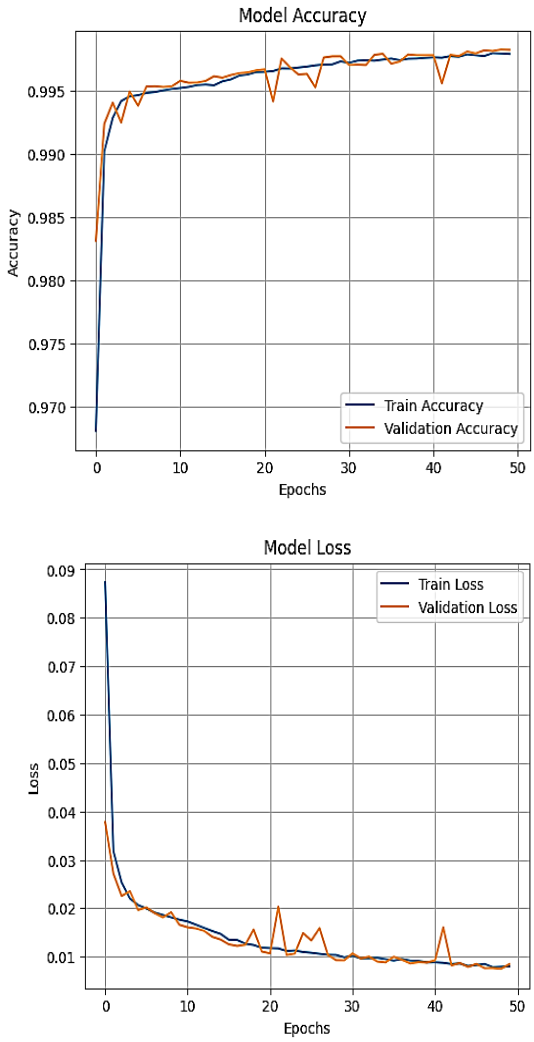

Figure 8. Accuracy vs Loss for CNN-GRU on AWID Dataset.

5. DISCUSSION

This study presented a comprehensive methodology for developing an intrusion detection system for IoT networks using optimized feature selection and deep neural networks. The results demonstrate significant performance improvements compared to conventional machine learning models.

Previous studies have developed various machine learning and deep learning approaches for intrusion detection in IoT environments, achieving strong results. Alkhudaydi et al. [7] utilized techniques like CatBoost and XGBoost on the BoT-IoT dataset to attain up to 98.5% accuracy. Faik et al. [8] applied Stacked Extremely Randomized Trees and XGBoost on encrypted Wi-Fi data to detect multiple attack types with 96.85% accuracy. Alharbi et al. [9] proposed a Local-Global Best Bat Algorithm optimized neural network (LGBA-NN) evaluated on the N-BaIoT dataset that reached 90% accuracy in multi-class botnet attack detection. Rajagopal et al. [10] developed a stacking ensemble methodology tested on the UGR'16 and UNSW NB-15 datasets, achieving 97% and 94% accuracy respectively. Our study builds on these works by using XGBoost for feature selection and a 1D CNN-GRU deep learning model, evaluated on the BoTNeTIoT and AWID datasets. Our proposed approach attains state-of-the-art accuracy up to 100% on BoTNeTIoT and 99.9% on AWID in detecting cyber-attacks on IoT networks. Table VI displays the comparison between the performance of our proposed approaches and exiting ones for intrusion detection using accuracy metric.

TABLE VI. COMPARATIVE ANALYSIS OF IOT INTRUSION DETECTION APPROACHES

Author & Year	Dataset	Method	Accuracy
Alkhudaydi et al. [7]	BoT-IoT	ML techniques including CatBoost and XGBoost combined with SMOTE	CatBoost:98.19%, XGBoost:98.50%
Faik et al. [8]	Encrypted Wi-Fi data	Stacked Extremely Randomized Trees and XGBoost	96.85%
Alharbi et al. [9]	N-BaIoT	Local–Global best Bat Algorithm for Neural Networks (LGBA-NN)	90%
Rajagopal et al. [10]	UGR'16 and UNSW NB-15	Stacking ensemble methodology	UGR'16: 97%, UNSWNB-15: 94%
Our study	BoTNeTIoT	XGBoost feature selection + 1D CNN-GRU	100%
Our study	AWID	XGBoost feature selection + 1D CNN-GRU	99.9%

6. CONCLUSIONS

This paper presented a novel intrusion detection system for IoT networks using XGBoost-based feature selection and deep neural networks. The methodology

involved collecting and preprocessing the BoTNeTIoT and AWID datasets, applying XGBoost to select the most predictive features, establishing machine learning baselines, designing a 1D CNN-GRU model architecture, and comprehensively evaluating the results.The findings demonstrate that the proposed techniques significantly outperform conventional ML models, achieving 100% and 99.9% accuracy on the two datasets respectively. The integrated feature selection and deep learning approach extracts optimal representations from the traffic data for accurate detection of IoT botnet intrusions and attacks.

Detailed ablation studies have thoroughly investigated the specific roles of convolutional neural network (CNN) layers in extracting comprehensive features and gated recurrent unit (GRU) layers in properly modelling temporal dependencies. The model's robustness has been thoroughly evaluated through rigorous testing, including targeting zero-day assaults. The suggested method is a major improvement in strengthening real-world Internet of Things (IoT) deployments against the constantly changing range of threats. This research has developed an efficient and highly effective Intrusion Detection System (IDS) by utilizing optimized feature selection techniques and harnessing the capabilities of deep neural networks. The utilized methodology, coupled with the acquired comparison results, provides useful insights that can inspire and guide future efforts to construct robust intrusion detection systems specifically designed for IoT networks. With the rapid increase in the usage of IoT technology, it is important to imports high security measures approaches by implementing powerful machine learning techniques.

ACKNOWLEDGMENT

I am deeply grateful to all the participants who generously shared their time and experiences for this research. Their contributions have been instrumental in the success of this study.

REFERENCES

[1] Edwards, S., & Profetis, I. (2016). Hajime: Analysis of a decentralized internet worm for IoT devices. Rapidity Networks, 16, 1-18.

[2] Abdulhammed, R., Musafer, H., Alessa, A., Faezipour, M., & Abuzneid, A. Features dimensionality reduction approaches for machine learning based network intrusion detection. Electronics, 2019, 8(3), 322.

[3] Chang, H., Hari, A., Mukherjee, S., & Lakshman, T. V. (2014, April). Bringing the cloud to the edge. In 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) (pp. 346-351). IEEE.

[4] Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. IEEE transactions on emerging topics in computational intelligence, 2(1), 41-50.

[5] Nguyen, H. T., Franke, K., & Petrović, S. (2011, November). A new ensemble-feature-selection framework for intrusion detection. In 2011 11th international conference on intelligent systems design and applications (pp. 213-218). IEEE.

[6] Panda, M., & Patra, M. R. (2007). Network intrusion detection using naive bayes. International journal of computer science and network security, 7(12), 258-263.

[7] Alkhudaydi, O. A., Krichen, M., & Alghamdi, A. D. (2023). A deep learning methodology for predicting cybersecurity attacks on the internet of things. Information, 14(10), 550.

[8] Örs, F. K., Aydın, M., Boğatarkan, A., & Levi, A. (2021, April). Scalable wi-fi intrusion detection for IOT systems. In 2021 11th IFIP International Conference on New Technologies, Mobility and Security (NTMS) (pp. 1-6). IEEE.

[9] Alharbi, A., Alosaimi, W., Alyami, H., Rauf, H. T., & Damaševičius, R. (2021). Botnet attack detection using local global best bat algorithm for industrial internet of things. *Electronics*, *10*(11), 1341.

[10] Rajagopal, S., Kundapur, P. P., & Hareesha, K. S. (2020). A stacking ensemble for network intrusion detection using heterogeneous datasets. *Security and Communication Networks*, *2020*, 1-9.

[11] Moustafa, N., Keshk, M., Choo, K. K. R., Lynar, T., Camtepe, S., & Whitty, M. (2021). DAD: A Distributed Anomaly Detection system using ensemble one-class statistical learning in edge networks. *Future Generation Computer Systems*, *118*, 240-251.

[12] Breitenbacher, D., Homoliak, I., Aung, Y. L., Tippenhauer, N. O., & Elovici, Y. (2019, July). HADES-IoT: A practical host-based anomaly detection system for IoT devices. In Proceedings of the 2019 ACM Asia conference on computer and communications security (pp. 479-484).

[13] Popoola, S. I., Adebisi, B., Hammoudeh, M., Gui, G., & Gacanin, H. (2020). Hybrid deep learning for botnet attack detection in the internet-of-things networks. *IEEE Internet of Things Journal*, *8*(6), 4944-4956.

[14] Putchala, M. K. (2017). Deep learning approach for intrusion detection system (ids) in the internet of things (iot) network using gated recurrent neural networks (gru).

[15] Tawfik, M., Al-Zidi, N. M., Alsellami, B., Al-Hejri, A. M., & Nimbhore, S. (2021, December). Internet of Things-Based Middleware Against Cyber-Attacks on Smart Homes using Software-Defined Networking and Deep Learning. In *2021 2nd International Conference on Computational Methods in Science & Technology (ICCMST)* (pp. 7-13). IEEE.

[16] Cao, B., Li, C., Song, Y., Qin, Y., & Chen, C. (2022). Network intrusion detection model based on CNN and GRU. *Applied Sciences*, *12*(9), 4184.

[17] Li, J.; Kuang, X.; Lin, S.; Ma, X.; Tang, Y.(2020). Privacy preservation for machine learning training and classification based on homomorphic encryption schemes. Inf. Sci., 526, 166–179.

[18] Sarica, A.K.; Angin, P. (2020). Explainable security in SDN-based IoT networks. Sensors , 20, 7326.

[19] Aleem, S.; Capretz, L.F.; Ahmed, F. (2015). Security Issues in Data Warehouse. arXiv , arXiv:1507.05644.

[20] Wu, M.; Song, Z.; Moon, Y.B.(2019). Detecting cyber-physical attacks in CyberManufacturing systems with machine learning methods. J. Intell. Manuf. , 30, 1111–1123.

[21] Patil, R.; Dudeja, H.; Modi, C.(2020). Designing in-VM-assisted lightweight agent-based malware detection framework for securing virtual machines in cloud computing. Int. J. Inf. Secur, 19, 147–162.

[22] Dang, T.K.; Pham, C.D.M.; Nguyen, T.L.P. (2020). A pragmatic elliptic curve cryptography-based extension for energy-efficient device-to-device communications in smart cities. Sustain. Cities Soc, 56, 102097.

[23] Moustafa, N.(2020). A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets. Sustain. Cities Soc, 72, 102994.

[24] Atul, D.J.; Kamalraj, R.; Ramesh, G.; Sakthidasan Sankaran, K.; Sharma, S.; Khasim, S.(2021). A machine learning based IoT for providing an intrusion detection system for security. Microprocess. Microsyst, 82, 103741.

[25] Ghosh, A.; Chakraborty, D.; Law, A.(2018). Artificial intelligence in Internet of things. CAAI Trans. Intell. Technol, 3, 208–218.

[26] Bland, J.A.; Petty, M.D.; Whitaker, T.S.; Maxwell, K.P.; Cantrell, W.A.(2020). Machine Learning Cyberattack and Defense Strategies. Comput. Secur, 92, 101738.

[27] Rathore, S.; Park, J.H. (2018).Semi-supervised learning based distributed attack detection framework for IoT. Appl. Soft Comput. J, 72, 79–89.

[28] Kasongo, S.M.; Sun, Y.(2020). A deep learning method with wrapper based feature extraction for wireless intrusion detection system. Comput. Secur, 92, 101752.

[29] Alhowaide, A., Alsmadi, I., & Tang, J. (2021). Towards the design of real-time autonomous IoT NIDS. *Cluster Computing*, 1-14.

[30] Chatzoglou, E., Kambourakis, G., & Kolias, C. (2021). Empirical evaluation of attacks against IEEE 802.11 enterprise networks: The AWID3 dataset. IEEE Access, 9, 34188-34205..