Computer Networks and Communications
https://ojs.wiserpub.com/index.php/CNC

Research Article

Ensemble Learning-Based DDOS Attack Recognition in IoT Networks

Muhammad Saibtain Raza[1], Mohammad Nowsin Amin Sheikh[2]*, I-Shyan Hwang[1]*.

1. Department of Computer Science and Engineering, Yuan Ze University, Taoyuan, 32003, Taiwan.
2. Department of Computer Science and Engineering, Jashore University of Science and Technology, Jashore 7408, Bangladesh.
*Correspondence: I-Shyan Hwang (ishwang@saturn.yzu.edu.tw), Mohammad Nowsin Amin Sheikh (n.amin@just.edu.bd).

Abstract: According to a quote by Brendan O'Brien, "If you think the Internet has changed your life, think again. The Internet of Things is about to change it all over again!". By improving data analytics, IoT operations, and human–machine interaction, the Internet of Things (IoT) and Artificial Intelligence (AI) are coming together to form AIoT, which is transforming modern production in the era of Industry 4.0. Although AIoT promises more sustainability, efficiency, and safety, the increasing use of IoT devices also increases their susceptibility to cyberthreats, among which Distributed Denial-of-Service (DDoS) attacks are among the most common. Unlocking the full potential of AIoT in linked and industrial environments requires addressing these security issues. In this paper, we leverage the publicly available CIC IoT 2023 dataset to conduct a comprehensive analysis of IoT-based cyber threats, focusing on the detection of seven major attack types and their respective subcategories. To guarantee the accuracy and applicability of the input data, a thorough feature extraction procedure was carried out. To evaluate detection performance, we applied a diverse set of six machine learning and deep learning models. Notably, the most successful approach was an ensemble learning strategy, which produced better accuracy and resilience. Thorough validation procedures were used to verify the results' generalizability and dependability, highlighting the promise of advanced learning methods in fortifying AIoT security infrastructures. Our research indicates that ensemble learning and deep learning models are a promising option for implementation in practical AIoT security frameworks as, when appropriately set up, they provide notable benefits for processing and categorizing tabular IoT data.

Keywords: AIoT DDoS Attacks, Features selection, Machine learning, Deep learning, Ensemble learning

1. Introduction

One of the most recent developments in the field of networking is the Internet of Things (IoT). The "interconnection of things" with limited computational capacity and power. It can be utilized to transmit and receive data via the internet without the need for human-to-human or human-to-computer communication. The term "things" describes networked, IP-enabled devices, both virtual and physical. Internet of Things (IoT) devices support a variety of technical infrastructures and trends, including smart cities, healthcare, transportation, homes, and the lives of ordinary people. During their activities, these internet-connected gadgets create, process, and share massive volumes of data. According to Statista the number of IoT devices presently exceeds 15.14 billion and is expected to reach 29.42 billion by 2030 [1]. Even though security needs are not given the same importance as product innovation, both enterprises and academics have focused increasingly on the development of security solutions for IoT devices [2] Software-defined Network (SDN) is one of the best security solutions that can detect and mitigate DDoS attacks very effectively[3-5]. The most common and serious IoT attacks are distributed denial-of-service (DDoS) and denial-of-service (DoS) said [6]. These attacks are intended to overload servers or online applications, causing service interruption or termination. To combat these threats, several methods have been

DOI: https://doi.org/10.37256/cnc.3220256755

developed, with intrusion detection/prevention systems and firewalls being the most common [7]. These security measures are often based on whitelisting or blacklisting. These methods, however, frequently fail to successfully counter more complex attack approaches. Furthermore, typical security techniques like encryption, authentication, access control, network security, and application security may be ineffective in safeguarding IoT devices and vulnerabilities [8]. As a result, present security solutions need be improved to offer acceptable security for the IoT ecosystem.

Attackers usually have one of five main reasons for launching an attack: financial gain, retaliation, ideological conviction, intellectual challenge, and cyberwarfare. As a result, each attacker's motivation for starting an attack will be different. Given the growing impact of these attacks, it is essential to identify and mitigate them before they reach their target. Among the most prevalent, frequent, and effective cyberattacks are Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. They can originate from several sources and take diverse shapes. Consequently, authorized users' access to the target network is limited by consuming the available bandwidth and resources [9] .

1.1 *Contributions*

To address these challenges, we propose an automated detection framework that reduces the feature space, thereby minimizing overfitting and lowering computational overhead. The process begins with thorough data preprocessing to enhance the model's generalizability. Subsequently, feature selection techniques are employed to identify the most relevant attributes, significantly improving classification accuracy. In addition, fine-tuning of learning algorithm parameters guided by methodologies such as those outlined by [10] further optimizes model performance. The key contributions of this study are summarized as follows

- We utilize the CIC IoT-2023 dataset to identify and classify seven major IoT attack types along with their subcategories, offering a more granular detection approach than prior studies.
- We apply and compare three feature selection techniques to isolate the most informative attributes, reducing dimensionality while enhancing model efficiency and accuracy.
- We further evaluate the importance of selected features using classifier-based analysis. This provides insight into which attributes most influence model decisions.
- A diverse set of machine learning, deep learning, and ensemble models is implemented and rigorously evaluated, providing a comparative assessment across methodologies.
- All models are validated on separate test data to ensure robustness, avoiding overfitting and confirming the models' applicability in real-world scenarios.
- Based on empirical results, we recommend deep learning models for tabular IoT data, highlighting their strong performance in high-dimensional, structured datasets.

The rest of the paper is organized as follows: Section 2 reviews related work in the domain of IoT security and attack detection. Section 3 outlines the proposed methodology for identifying IoT-based attacks. In Section 4, we provide a detailed description of the machine learning, deep learning, and ensemble models employed. Section 5 presents the experimental setup, results, and performance evaluation. Finally, Section 6 concludes the study and outlines directions for future research.

2. Related Work

The most current and popular methods for identifying IOT attacks are briefly covered in this section. Table I shows the existing works.

The author[11] implements different machine learning algorithms such as Random Forest (RF), Naive Bayes (NB), and Decision Tree (DT) to analyze the performance of DDoS attacks using the BoT-IoT dataset. Twenty-one features were employed; however, no feature selection techniques were applied in their work.

Utilized a variety of classification algorithms—including Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Naïve Bayes (NB), and Extreme Gradient Boosting (XGBoost)—to

detect nine different attack types within the Kitsune IoT dataset. The study employed 115 features, with ANOVA and Chi-square tests used for feature selection to improve model performance. However, the approach faced certain limitations, including high computational overhead due to the large feature set and potential overfitting in some models. Additionally, the evaluation was limited to a single dataset, which may restrict the

generalizability of the findings across diverse IoT environments [12] .

Also author employed CatBoost and XGBoost algorithms to detect threats using the Realistic Edge-IIoTset dataset. While the study demonstrated the potential of gradient boosting methods in Industrial IoT (IIoT) environments, it lacked detail on the number and type of features used. Furthermore, the absence of a feature selection process and limited model diversity may restrict the interpretability and scalability of the proposed approach [13].

One more author utilized Decision Tree (DT), Random Forest (RF), and Naïve Bayes (NB) classifiers to analyze the IoT-23 dataset for attack detection. However, the study did not specify the feature set used, nor did it incorporate feature selection techniques. Additionally, the limited range of models and lack of detailed evaluation metrics may constrain the comprehensiveness of the results [14].

To employed Light Gradient Boosting Machine (LGBM), Random Forest (RF), K-Nearest Neighbors (KNN), and Support Vector Machine (SVM) classifiers to detect attacks using a self-generated IoT dataset. However, the study did not provide details on the feature set or employ feature selection methods. Additionally, the use of proprietary data limits the reproducibility and generalizability of the findings [15].

One more author presented a mechanism for identifying and counteracting suspicious activities and harmful attacks as fast as possible. Linear SVM and Non-Linear SVM algorithms were used. However, the study lacked specification of the dataset and feature set used, and did not incorporate feature selection or validation techniques, limiting the clarity and generalizability of the results [16].

Another author applied Convolutional Neural Networks (CNN) and Support Vector Machines (SVM) to detect attacks within the UNSW-NB15 dataset. However, the study did not provide details on feature selection or preprocessing steps, and lacked comprehensive validation, which may affect the robustness and generalizability of the findings [17].

Furthermore, author first developed a comprehensive dataset encompassing 33 types of scans and 60 varieties of DDoS attacks, using both the Kitsune and a self-generated 5G dataset. They proposed a two-stage machine learning framework employing Decision Tree (DT), Random Forest (RF), K-Nearest Neighbors (KNN), and stacking models for the prevention and detection of IoT botnet attacks. A robust feature selection strategy

Table 1. Existing algorithms and Feature selection techniques.

Paper	Algorithms	Features Extracted	Dataset	Features Selection Techniques
[11]	RF, NB and DT	21	BoT-IoT	Not (Smart city)
[12]	DT, RF, SVM, KNN, NB and XGB	115 features	Kitsune IoT (9 different attacks)	ANOVA and Chi-square
[13]	CatBoost, and XGBoost	Not Mentions	Realistic EdgeIIoTset	Ensemble models
[14]	DT, RF, NB	No	Iot-23	Not Mention
[15]	LGBM, RF, KNN, and SVM	No	Self-generated Data	Not (SD-IoT)
[16]	Linear SVM and Non-Linear SVM	Not mention (how many)	No	Principal component analysis
[17]	CNN, SVM	No	UNSW-NB15	No
[18]	DT, RF, KNN and Stacking	Yes	Kitsune and 5G self-generated	Filter, wrapper and embedded used
[19]	ANN, DT, RF	Yes	VARIoT	Not mentioned
[20]	CNN	Yes	CICDDoS 2019	ANOVA, Chi-Squared, Mutual info
[21]	SVM, KNN, NB,DT,RF	Yes	Not Mention	Correlation
[22]	XGBoost	Yes	TON-IoT	wrapper and embedded based method
[23]	LSTM, CNN	No	N-BaIoT	Not Mention
Our paper	**CNN, Decision Tree Classifier, Random Forest Classifier, HistGradient Boosting Classifier.**	**29 Features extracted**	**CIC IoT 2023**	**By AI models check feature importance. Correlation on threshold values. From sckit learn used mutual_info_classif and Select Best method used.**

combining filter, wrapper, and embedded methods was applied to enhance model performance. While the approach demonstrates promising results, further validation on diverse datasets would strengthen its applicability across varied IoT environments [18].

Another study proposed a curated dataset, VARIoT, which integrates 40 contemporary IoT behavior datasets through feature reduction and class balancing techniques. Using this dataset, the study evaluated several machine learning algorithms, including Artificial Neural Networks (ANN), Decision Trees (DT), and Random Forests (RF), applied to preprocessed and SMOTE-balanced network data for detecting compromised IoT devices. However, specific details regarding feature selection methods were not reported [19].

One more study to addressed the challenge of detecting DDoS attacks on IoT devices without compromising detection performance by selecting a subset of the most relevant features from the original dataset to reduce input dimensionality. Prior to dimensionality reduction, an efficient and cost-effective model was developed to preprocess and clean the raw data. To identify the most significant features and minimize the data required for accurate detection, a hybrid feature selection approach was employed, combining Mutual Information (MI), Analysis of Variance (ANOVA), Chi-Squared tests, L1-based feature selection, and tree-based methods. However, the complexity of this multi-step feature selection process may increase computational overhead, and the approach was evaluated on a limited dataset, potentially restricting its generalizability to diverse IoT environments [20].

Author proposed ENTER, an evolutionary algorithm-based feature selection method for detecting Low-Rate Distributed Denial of Service (LDDoS) attacks in Software-Defined IoT (SD-IoT) within smart grids. ENTER utilizes multi-correlation information to evaluate relationships among features and adaptively modifies population variations. It incorporates a novel local optimal bounce technique and a unique gene mutation approach guided by multi-correlation data. Evaluation results show that ENTER improves LDDoS detection performance while achieving a high feature compression ratio, enhancing recall, precision, F-score, accuracy, and detection time. However, the complexity of the algorithm may lead to increased computational overhead, which could limit its applicability in real-time or resource-constrained environments [21].

A hybrid framework author proposed that utilizes the XGBoost algorithm for feature ranking to develop an

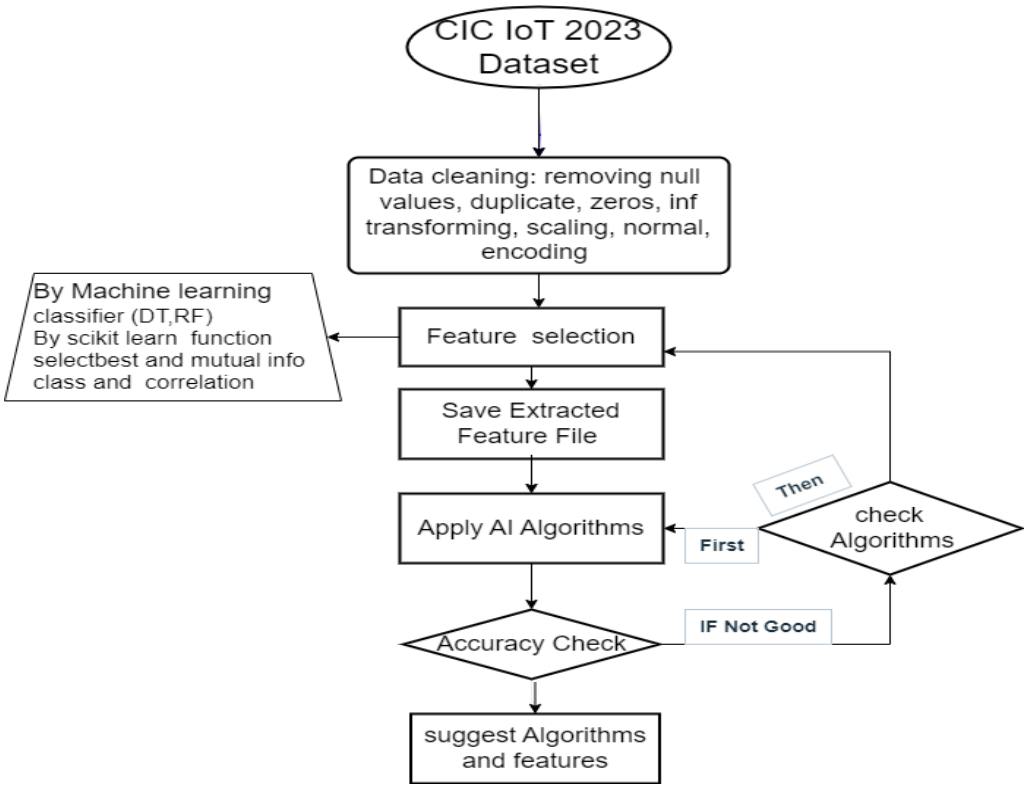

Fig. 1. Methodology.

effective intrusion detection system (IDS) for the Internet of Things. The framework was evaluated using the TON-IoT dataset, specifically designed for IoT intrusion detection. Experimental results demonstrate that the proposed method achieves optimal model performance while conserving computational resources. However, the framework's evaluation was limited to a single dataset, which may affect its generalizability across diverse IoT environments [22].

Liu proposed a deep learning-based detection model combining Random Forest with cascade architecture to

identify anomalous traffic in IoT environments. Random Forest was used for feature evaluation to eliminate irrelevant attributes, and the selected features were then used as input for Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) models to detect abnormal traffic patterns. The study also included a comparative analysis of different feature dimensions, classification methods, and algorithmic approaches. However, the model's effectiveness may be constrained by the absence of real-world deployment scenarios and potential scalability challenges in large-scale IoT networks [23].

While numerous existing studies have explored intrusion detection in IoT environments using machine learning and deep learning techniques, many are limited by outdated datasets, insufficient feature selection processes, or narrow focus on specific attack types such as DDoS. In contrast, our work addresses these gaps by leveraging the recent and comprehensive CIC IoT 2023 dataset, enabling the detection of seven major IoT attack categories and their subtypes. We apply a combination of statistical and model-based feature selection techniques to extract 29 relevant features, and evaluate multiple models including CNN, Decision Tree, Random Forest, and Histogram-Based Gradient Boosting for robust and interpretable detection performance, shown in Table 1.

3. Proposed Work

In this study, we utilized the CIC IoT 2023 Dataset, specifically designed for the categorization of IoT-based attacks. Our workflow began with a meticulous data preprocessing phase, where we ensured the dataset's integrity by removing null values, duplicates, and irrelevant entries. This step was critical in eliminating noise and enhancing the quality of the data. Subsequently, we applied standardized preprocessing techniques such as transformation, scaling, and normalization to ensure uniformity across the dataset and prepare it for accurate machine learning analysis.

To identify the most influential predictors for our machine learning models, we employed advanced feature selection methods provided by Scikit-Learn. This approach allowed us to isolate the features that had the highest predictive value, enhancing the models' efficiency and reducing computational complexity. To maintain a systematic approach and ensure analytical reliability, these selected features were archived separately for future reference and reproducibility.

Our methodology, shown in Figure 1, then focused on evaluating the dataset using various machine learning classifiers, with a particular emphasis on Decision Trees and Random Forest algorithms. These models were chosen for their proven effectiveness in handling structured data and their ability to capture complex patterns in IoT attack scenarios. The key features extracted during the selection process are summarized in Table 2, while their distribution and relationships are visually represented in a heat map, shown in Figure 2.

When a model's performance fell short of expectations, we undertook a detailed review of the selected

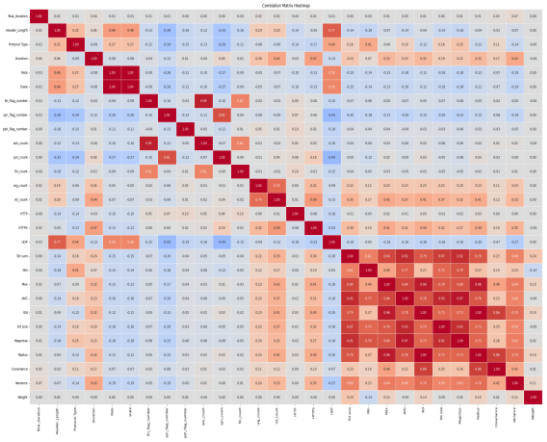

Fig. 2. Heat map of correlated features

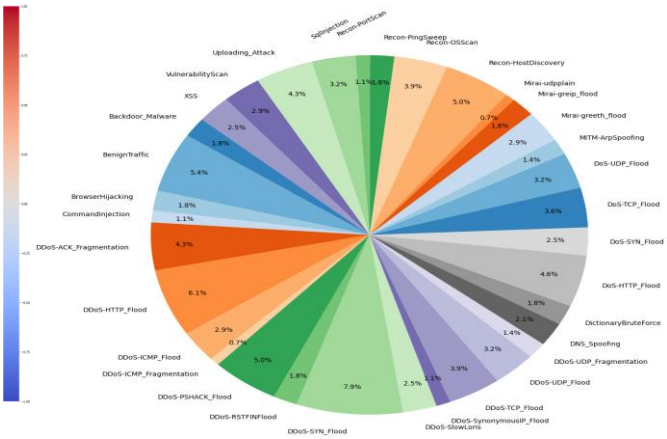

Fig. 3 Classification of IoT Dataset.

features and preprocessing techniques. This iterative process allowed us to refine our approach and systematically improve the model's accuracy. Such adaptive adjustments highlight the importance of continuous evaluation and optimization in achieving robust machine learning solutions for IoT security challenges.

3.1 *Dataset*

In our research, we utilized the publicly available CIC IoT 2023 dataset provided by Neto [24]. This dataset

represents an IoT environment comprising 105 devices subjected to 33 distinct types of attacks. These attacks are categorized into seven main types: DDoS, DoS, Recon, Web-based, Brute Force, Spoofing, and Mirai. Each attack scenario simulates malicious IoT devices aiming to compromise the functionality or security of other IoT devices. Figure 3 illustrates the distribution of attack types within the dataset, highlighting the dominance of DDoS-related attacks, which constitute nearly 70% of the total. This high percentage underscores the prevalence and severity of DDoS attacks in IoT ecosystems, emphasizing the importance of robust defense mechanisms tailored to mitigate such threats effectively.

3.2 *Data Pre-processing*

Preparing (cleaning and organizing) raw data is the main goal of this critical step in any machine learning approach, as it enables the development and training of any machine learning model. The steps involved in pre-processing are as follows:

Exploratory Data Analysis: The accurate information in the dataset is analyzed, visualized, and summarized

Table 2. Selected Features.

Feature Names	Details
flow_duration	Total duration of the flow in microseconds.
Header_Length	Total length of the headers in the flow.
Protocol_Type	Type of protocol used in the flow.
Duration	Duration of the flow session.
Rate	Rate of the flow in bytes/sec.
Srate	Source rate of packets in the flow.
fin_flag_number	Number of FIN flags in the flow.
syn_flag_number	Number of SYN flags in the flow.
psh_flag_number	Number of PSH flags in the flow.
ack_count	Count of acknowledgments in the flow.
syn_count	Count of SYN flags in the flow
fin_count	Count of FIN flags in the flow
urg_count	Count of URG flags in the flow.
rst_count	Count of RST flags in the flow.
HTTP	Traffic identified as HTTP.
HTTPS	Traffic identified as HTTPS.
UDP	Traffic identified as UDP.
Tot_sum	Total sum of bytes in the flow.
Min	Minimum packet size in the flow.
Max	Maximum packet size in the flow.
AVG	Average size of packets in the flow.
Std	Standard deviation of packet size in the flow.
Tot_size	Total size of all packets observed in the flow.
Magnitude	Magnitude of the flow.
Radius	Radius of the flow data points.
Covariance	Covariance of the flow features.
Variance	Variance of the flow in some context.
Weight	Importance of the flow in some context.

using EDA. It allows us to derive several crucial statistical insights and figures for analyzing the dataset, such as count, unique, top, and frequency for categorical data, and count, mean, standard deviation, min value, 25th, 50th, and 75th percentiles, and maximum value for numerical data.

Data Cleaning: To reduce computing costs, duplicates are removed. Moreover, the tuples that include negative, inf, NaN, or missing values are left in place. Because each piece of data has some crucial information for categorizing the assault. Lastly, the noisy data is processed by appending the missing and inf values with the feature's median values and imputing the negative values with 0.

Feature Scaling: For feature scaling, we may use a variety of scalers (such as robust, min-max, etc.), depending on your model and data. We used a standard scaler and then converted it to the machine learning model. We transformed x_data via a Minmax scaler for the deep learning (CNN) model. Next, our target data must be

encoded. To do this, we used a lab encoder before transforming.

Feature selection: Feature selection plays a pivotal role in achieving a high-performing model by directly influencing its accuracy, interpretability, and generalization capabilities. In this study, we utilized various feature selection methods to identify the most relevant features for attack classification. These selected features significantly influenced the model's behavior, enhancing its ability to differentiate between attack types and reducing the risk of overfitting. First, AI-based models such as Decision Trees and Random Forests were employed to identify features with high information gain, enabling the classification algorithm to make more accurate and effective decisions. The selected features, illustrated in Figure 4, contributed to the model's predictive power by prioritizing critical decision points. Second, we used the Scikit-learn library's mutual_info function, as shown in Figure 5, to identify features with strong statistical relationships to the target variable. This approach allowed the model to capture subtle patterns in the data, further boosting its classification performance. Lastly, we applied correlation-based feature selection by setting a threshold value of 0.6 to eliminate highly correlated features, ensuring that only the most informative and non-redundant features were retained. This step reduced noise and improved the model's robustness. Together, these feature selection techniques enhanced the model's efficiency and accuracy, ultimately leading to a reliable and high-performing classification system.

4. AI Models

We have used machine learning, deep learning, and ensemble learning models for the classification of IoT attacks. Convolution Neural Network, XGBOOST Classifier, HistGradient Boosting Classifier, Random Forest Classifier, and Decision Tree classifier.

4.1 *Convolution Neural Network (CNN)*

CNNs are a type of deep neural network that is mainly utilized to analyze visual data. Through layers that analyze the image in segments, they are especially known for the ability to identify patterns and characteristics in images, such as edges, textures, and objects [20].

```
Feature ranking:
1. feature UDP (0.14400334946314547)
2. feature Min (0.13090500623361379)
3. feature syn_flag_number (0.07586295187770954)
4. feature flow_duration (0.06586976102724942)
5. feature rst_count (0.06467344734647154)
6. feature syn_count (0.0627057389413103)
7. feature Protocol Type (0.05305074090180044)
8. feature psh_flag_number (0.04806863030679595)
9. feature Header_Length (0.04645970198495097)
10. feature Magnitue (0.03872408836577557)
11. feature fin_flag_number (0.03566608688694938)
12. feature Rate (0.03478735312581005)
13. feature Tot size (0.030854455720142723)
14. feature Srate (0.03029107850682911)
15. feature Max (0.02917407349284574)
```

Fig. 4. Features selection by AI models.

Tot size	1.509003e+00
Magnitue	1.467665e+00
Min	1.444503e+00
AVG	1.433813e+00
Tot sum	1.429576e+00
Max	1.369269e+00
Header_Length	1.340041e+00
Protocol Type	1.297696e+00
Srate	8.096916e-01
Rate	8.095099e-01
TCP	7.471078e-01
syn_count	7.365692e-01
Std	7.334632e-01
Radius	7.224601e-01
Covariance	6.924445e-01
flow_duration	6.918046e-01
rst_count	6.010023e-01
UDP	5.991476e-01

Fig. 5. Features selection by Mutual info.

4.2 *XGBooST Classifier (Boosting)*

Developed to improve gradient boosting methods' efficiency and speed. It can easily handle huge datasets, which makes it useful for machine learning applications including classification. To minimize a loss function (such as the logistic loss for classification tasks), XGBoost introduces predictors to an ensemble one after the other, correcting previous ones in the process described by[22].

4.3 *HistGradient Boosting Classifier*

According to the scikit-learn library, this estimator excels in handling large datasets, outperforming the Gradient Boosting Classifier in terms of speed. It natively supports missing values (NaNs), ensuring robust performance even with incomplete data. During training, the tree grower dynamically determines the placement of samples with missing values at each split point, based on the potential information gain. For prediction, samples

with missing values are assigned to either the left or right child node. If no missing values are present for the feature during training, these samples default to the child node containing the majority of the data. This adaptive approach enhances the model's ability to manage missing data efficiently.

4.4 *Random Forest Classifier*

An algorithm integrated into the RF ensemble learning technique is one of the most widely used ones for multi-classification and prediction problems. It combines ensemble learning with bagging and random subspace. As its name suggests, RF generates output in bagging by using a random approach [11].

4.5 *Decision Tree Classifier (Ensemble Learning)*

The Random Forest classifier constructs multiple decision trees using randomly selected subsets of the training data. It aggregates the predictions from these decision trees by collecting their votes to determine the final output, as explained by [11].

4.6 *Long short-term memory (LSTM)*

An RNN type called the LSTM algorithm can remember sequential data in memory and identify long-term dependencies. When gradient inversion procedures gradually deteriorate during computation, it resolves the vanishing gradient problem. When it comes to time series-related issues, the LSTM algorithm is appropriate. It applies to algorithms in speech recognition, video processing, and language processing because of these qualities. Cells are memory chunks that make up the LSTM algorithm explained by [24].

5. Experiment Result

This section provides a comprehensive analysis of our experimental methodology and results. We begin by detailing the computational environment and setup, including hardware specifications and software configurations. Next, we present a rigorous evaluation of model performance, focusing on key metrics such as accuracy, ROC-AUC curves, and validation scores across multiple architectures. A comparative assessment of different models is also included to highlight their strengths and limitations in addressing the given task.

5.1 *Experiment Environment*

The experiments were conducted on a Windows 10 workstation with 72GB of RAM to handle large datasets and complex model training efficiently. For accelerated deep learning, we used an NVIDIA RTX A2000 GPU (12GB VRAM), taking advantage of its CUDA cores and Tensor Cores for optimized neural network training.

The software setup relied on Anaconda for managing dependencies, with Python 3.9 and TensorFlow as the core framework. GPU acceleration was enabled through CUDA Toolkit and CuDNN. We developed and tested our models in Visual Studio Code (VSCode), which provided useful features like debugging, an integrated terminal, and Python/Jupyter extensions for smoother experimentation

5.2 *Experiment Results*

We use ROC curves, validation scores, and accuracy to evaluate each model's performance. After model training, we use Receiver Operating Characteristic (ROC) curves to display the classification outcomes for each attack type. Our evaluations indicate the proposed ensemble XGBoost model performs well, consistently and precisely recognizing the possible IoT threat.

ROC: A graphical representation that shows how a binary classifier model which is also capable of being utilized for multi-class classification performs at different threshold settings.

ROC Curves of the CNN Model: The CNN model was designed with dense layers of 2000, 1500, 800, 400, 150, and 34 neurons, using ReLU activation for intermediate layers and Softmax for multiclass classification. Dropout layers were interspersed to mitigate overfitting. The model was trained with a batch size of 256 over 20 epochs, leveraging 4,645,984 trainable parameters to classify IoT-related data effectively.

Evaluation was performed using Receiver Operating Characteristic (ROC) curves, shown in Figure 6(a), highlighting the model's ability to distinguish between 34 target classes.

ROC Curves of the Decision Tree Classifier: The Decision Tree Classifier demonstrated varying levels of performance across the 34 classes. While four classes achieved outstanding ROC values, indicating excellent prediction capabilities, the majority of classes achieved prediction accuracies ranging between 20% and 60%. These results are visualized in Figure 6(b), reflecting the model's challenges in maintaining consistent performance across all categories.

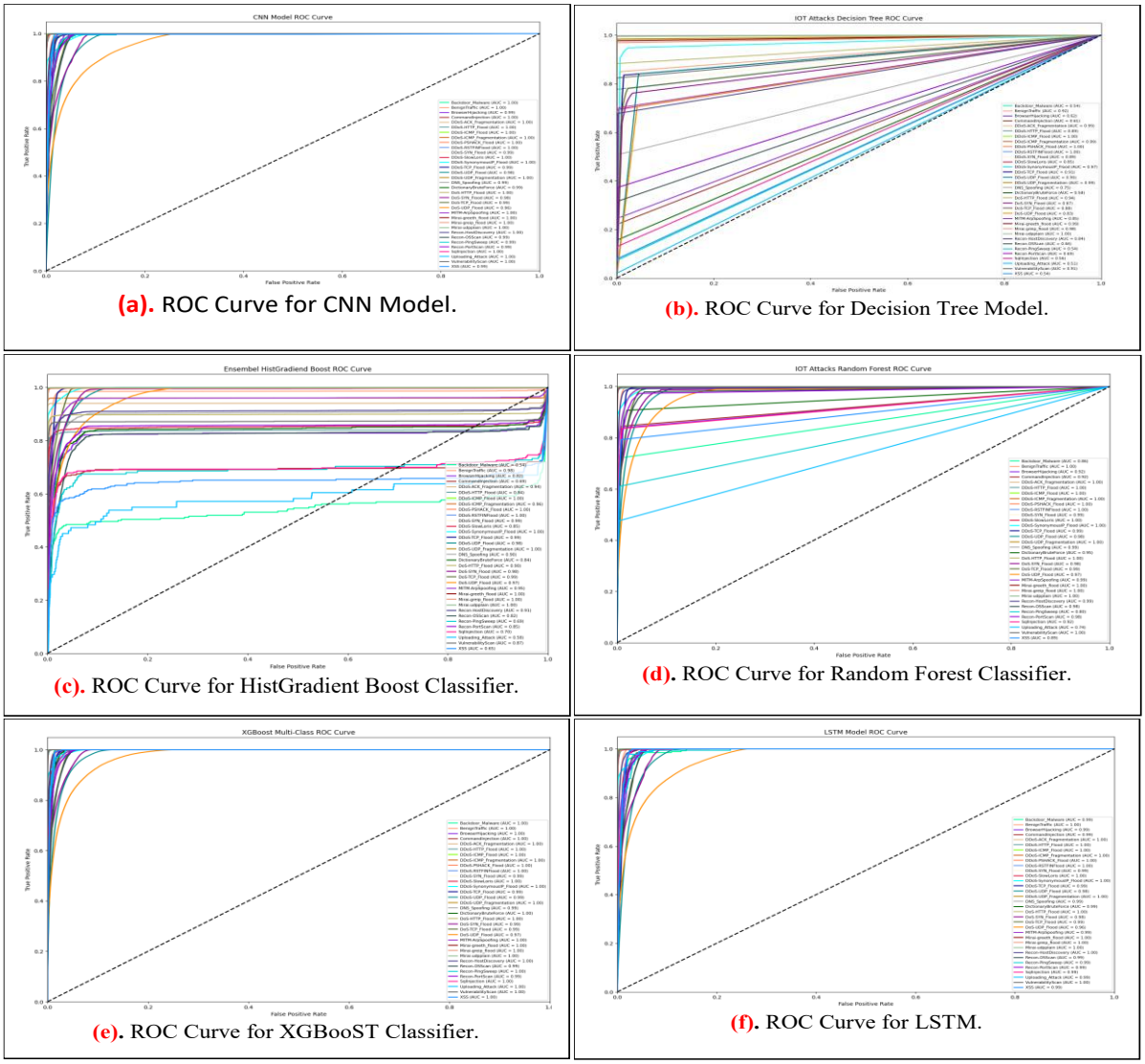

Fig. 6. ROC Curve of different Models.

ROC Curve of the HistGradient Boost Classifier: The HistGradient Boost Classifier, a boosting ensemble model, exhibited strong predictive performance across nearly all 34 classes. However, a subset of 5 to 6 classes showed prediction accuracies ranging between 40% and 70%, suggesting some difficulty in distinguishing certain patterns. These results, as depicted in Figure 6(c), highlight the model's robustness in most cases while indicating areas for further optimization.

ROC Curve of the Random Forest Classifier: Among the models tested, the Random Forest Classifier emerged as a strong performer, second only to the CNN model in terms of ROC metrics. As illustrated in Figure 6(d), this model effectively captured complex decision boundaries, resulting in high predictive accuracy across the majority of classes. Its ensemble approach proved effective in handling the diverse IoT attack dataset.

ROC Curve of the XGBoost Classifier: The XGBoost Classifier demonstrated exceptional performance, surpassing most other classifiers in terms of prediction accuracy. Nearly all classes were correctly predicted, with only minimal deviations. Figure 6(e) showcases the model's ability to effectively classify 34 different types of IoT attacks, underscoring its utility as a robust and reliable classification tool.

ROC Curve of the LSTM Model: The Long Short-Term Memory (LSTM) model proved to be a powerful

choice for categorization tasks. Utilizing the Adam optimizer and Softmax activation function, this model achieved near-perfect prediction accuracy across all classes. As shown in Figure 6(f), the LSTM model performed comparably to the CNN model, providing precise classification results for 34 distinct types of attacks. Its ability to model temporal dependencies likely contributed to its superior performance.

Accuracy is defined as the percentage of successfully predicted instances for a class compared to all predictions made for that class. Figure 7 displays the accuracy of six different machine-learning models. According to the results, Random Forest and XGBoost led the field with an accuracy of 0.87, showcasing their adaptability in handling complex, high-dimensional datasets. The Decision Tree achieves a slightly lower performance score of 0.84, which may indicate limitations in its ability to handle the dataset's complexity or variance. In comparison, both the CNN and LSTM models achieve scores of 0.85, highlighting their effectiveness and versatility, particularly for tasks involving sequential data. The Histogram Gradient Boosting Tree further displays its efficacy in scaled environments with an excellent performance of 0.85.

Validation of Models: Model validation assesses a model's ability to generalize to new, unseen data, ensuring reliable and accurate predictions. It employs techniques such as cross-validation and metrics like accuracy to evaluate performance. Validation is critical for fine-tuning models, reducing overfitting, and selecting the most effective algorithm for a given task.

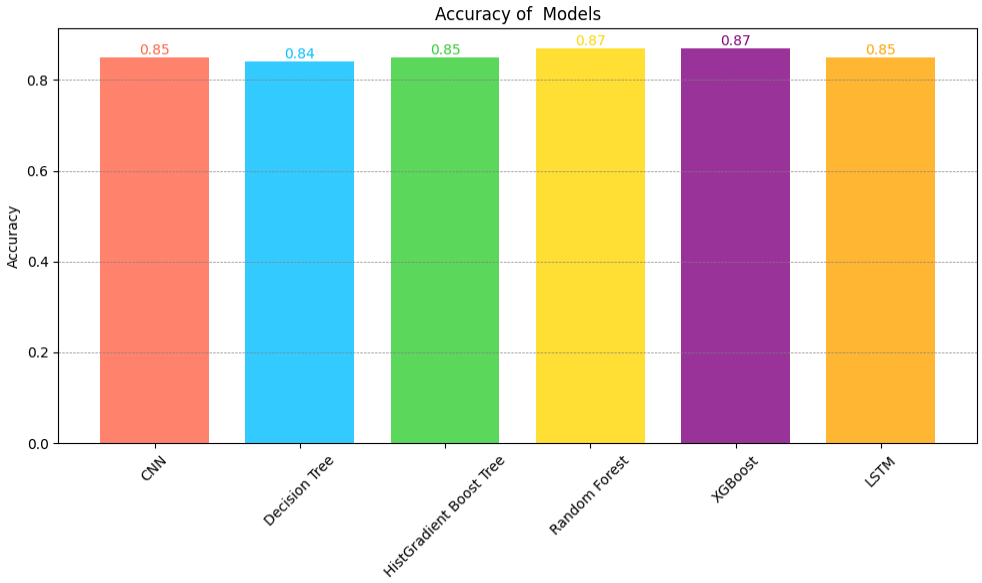

Fig. 7. Accuracy of Applied Models.

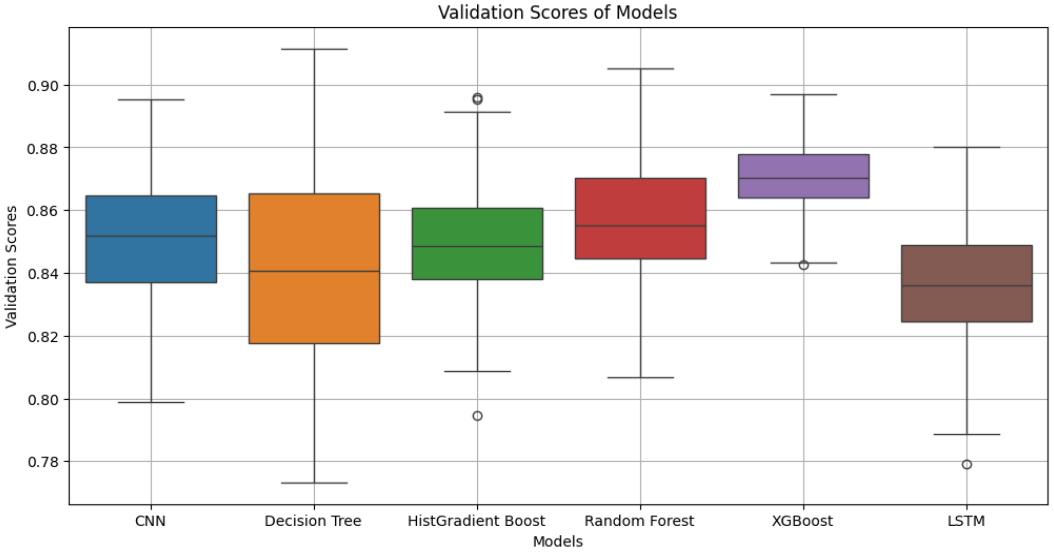

Fig. 8. Validation score of Models.

Figure 8 presents the validation scores of six machine learning models, highlighting notable performance differences. Among these, XGBoost achieves the highest validation score of 0.87, demonstrating superior

generalization capabilities. Random Forest follows closely with a validation score of 0.86, showcasing strong predictive reliability. Similarly, HistGradient Boost performs commendably, attaining a validation score of 0.85. In comparison, the CNN, LSTM, and Decision Tree models exhibit slightly lower validation scores of 0.84, indicating competent but less competitive performance when handling unseen data.

When considering both validation and accuracy metrics, XGBoost and Random Forest lead the group, with both achieving a score of 0.87. This dual strength underscores their effectiveness and reliability in prediction tasks. HistGradient Boost also shows significant potential, with validation and accuracy scores of 0.85 each. In contrast, CNN and LSTM maintain consistent validation and accuracy scores of 0.84, reflecting steady but average performance. The Decision Tree model, however, lags behind, achieving the lowest scores of 0.84 for both validation and accuracy, highlighting its relative inefficiency compared to the other models.

6. Conclusions

A reliable approach to addressing IoT security issues has been demonstrated through the application of multiple AI algorithms for feature extraction and classification using the CIC IoT 2023 dataset. This work integrates machine learning, deep learning, and ensemble methods, including boosting techniques, to enhance detection accuracy and evaluate the relative performance of various models through rigorous comparative analysis. The proposed ensemble learning framework, leveraging XGBoost and Random Forest, outperformed other algorithms in terms of classification accuracy and robustness. Moreover, Convolutional Neural Networks (CNNs), traditionally recognized for their effectiveness in image processing, demonstrated strong performance in handling structured IoT sensor data. This finding suggests the potential of CNNs to extend beyond their conventional applications, enabling advanced deep learning methodologies for IoT data analysis. The results provide a foundation for exploring CNN-based solutions in non-image data contexts, such as IoT security.

The study also highlights the role of AI-driven feature selection techniques in enhancing predictive accuracy, showcasing their superiority over traditional statistical methods like mutual information and correlation. These advanced methods enable the identification of high-value features, which significantly improve model performance. Future work will focus on generating a dedicated dataset based on the optimal features identified, followed by an in-depth exploration of deep learning techniques, particularly CNNs, to further enhance IoT attack detection capabilities.

Declaration

This project was supported by NSTC 113-2221-E-155-055. The authors have no conflicts of interest to declare that are relevant to the content of this article.

Conflict of Interest

There is no conflict of interest for this study.

References

[1] L. S. Vailshery, "Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2021, with forecasts from 2022 to 2030," *Accessed: Dec,* vol. 22, 2022.

[2] O. Cheikhrouhou, I. Amdouni, K. Mershad, M. Ammi, and T. N. Gia, "Blockchain for the cybersecurity of smart city applications," *arXiv preprint arXiv:2206.02760,* 2022.

[3] M. Sheikh, M. Halder, S. Kabir, and R. Mohalder, "Performance evaluation on software defined networking through external controller Floodlight and internal controller NOX," *International Journal of Scientific & Engineering Research,* vol. 9, no. 7, 2018.

[4] M. N. A. Sheikh, I.-S. Hwang, E. Ganesan, and R. Kharga, "Performance assessment for different SDN-based controllers," in *2021 30th Wireless and Optical Communications*

Conference (WOCC), 2021: IEEE, pp. 24-25.
[5] M. N. A. Sheikh, I.-S. Hwang, M. S. Raza, and M. S. Ab-Rahman, "A qualitative and comparative performance assessment of logically centralized SDN controllers via Mininet emulator," *Computers,* vol. 13, no. 4, p. 85, 2024.
[6] A. Aminu Ghali, R. Ahmad, and H. S. A. Alhussian, "Comparative analysis of DoS and DDoS attacks in Internet of Things environment," in *Artificial Intelligence and Bioinspired Computational Methods: Proceedings of the 9th Computer Science On-line Conference 2020, Vol. 2 9*, 2020: Springer, pp. 183-194.
[7] N. Gupta, V. Naik, and S. Sengupta, "A firewall for internet of things," in *2017 9th International Conference on Communication Systems and Networks (COMSNETS)*, 2017: IEEE, pp. 411-412.
[8] K. Mershad and O. Cheikhrouhou, "Lightweight blockchain solutions: Taxonomy, research progress, and comprehensive review," *Internet of Things,* vol. 24, p. 100984, 2023.
[9] K. Singh, K. S. Dhindsa, and D. Nehra, "T-CAD: A threshold based collaborative DDoS attack detection in multiple autonomous systems," *Journal of Information Security and Applications,* vol. 51, p. 102457, 2020.
[10] R. K. Batchu and H. Seetha, "A generalized machine learning model for DDoS attacks detection using hybrid feature selection and hyperparameter tuning," *Computer Networks,* vol. 200, p. 108498, 2021.
[11] A. Sharma and H. Babbar, "Bot-iot: Detection of ddos attacks in internet of things for smart cities," in *2023 10th International Conference on Computing for Sustainable Global Development (INDIACom)*, 2023: IEEE, pp. 438-443.
[12] K. Lee, Majd, Nahid, *Anomaly Detection and Attack Classification in IoT Networks Using Machine Learning*. 2023/11/17.
[13] K. Keserwani, A. Aggarwal, and A. Chauhan, "Attack detection in industrial IoT using novel ensemble techniques," in *2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN)*, 2023: IEEE, pp. 1-6.
[14] M. J. Gul and M. K.-u.-R. R. Syed, "Network attack detection in iot using artificial intelligence," in *2023 International Multi-disciplinary Conference in Emerging Research Trends (IMCERT)*, 2023, vol. 1: IEEE, pp. 1-6.
[15] P. Chauhan and M. Atulkar, "An efficient LGBM based DDoS attack Detection Approach for SD-IoT," in *2023 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, 2023: IEEE, pp. 1-10.
[16] A. Srivastava, S. Tiwari, P. K. Saini, V. Sawan, and S. A. Dhondiyal, "Attack Detection and Mitigation in IoT using SVM," in *2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, 2023: IEEE, pp. 1547-1551.
[17] S. Singh, S. V. Fernandes, V. Padmanabha, and P. Rubini, "Mcids-multi classifier intrusion detection system for iot cyber attack using deep learning algorithm," in *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, 2021: IEEE, pp. 354-360.
[18] F. Hussain *et al.*, "A two-fold machine learning approach to prevent and detect IoT botnet attacks," *Ieee Access,* vol. 9, pp. 163412-163430, 2021.
[19] Y. R. Siwakoti and D. B. Rawat, "Detect-iot: A comparative analysis of machine learning algorithms for detecting compromised iot devices," in *Proceedings of the Twenty-fourth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, 2023, pp. 370-375.
[20] L. Hong, K. Wehbi, and T. H. Alsalah, "Hybrid feature selection for efficient detection of DDoS attacks in IoT," in *Proceedings of the 2022 6th International Conference on Deep Learning Technologies*, 2022, pp. 120-127.
[21] X. Huan, J. Zhao, G. Chen, X. Li, Y. Cui, and H. Yang, "Towards Feature Selection for Detecting LDDoS in SD-IoT of Smart Grids: A Multi-correlation Information EA-based Method," in *Proceedings of the 2023 2nd International Symposium on Computing and Artificial Intelligence*, 2023, pp. 60-66.
[22] T. N. Fatyanosa and M. Data, "Hybrid Feature Selection Framework for Building Resource Efficient Intrusion Detection Systems Model in the Internet of Things," in *Proceedings of the 8th International Conference on Sustainable Information Engineering and Technology*, 2023,

pp. 16-22.
[23] Y. Liu, Z. Lv, and Z. Liu, "Research on abnormal traffic detection of Internet of Things based on feature selection," in *Proceedings of the 2023 4th International Conference on Computing, Networks and Internet of Things*, 2023, pp. 576-582.
[24] S. Lin and H. Tian, "Short-term metro passenger flow prediction based on random forest and LSTM," in *2020 IEEE 4th information technology, networking, Electronic and Automation Control Conference (ITNEC)*, 2020, vol. 1: IEEE, pp. 2520-2526.