computers

MDPI

Article

Machine Learning-Based Dynamic Attribute Selection Technique for DDoS Attack Classification in IoT Networks

Subhan Ullah [1], Zahid Mahmood [2], Nabeel Ali [3], Tahir Ahmad [4,*] and Attaullah Buriro [5,*]

[1] Department of Computer Science, National University of Computer and Emerging Sciences (NUCES-FAST), Islamabad 44000, Pakistan; subhan.ullah@nu.edu.pk
[2] Department of Computer Science and IT, University of Kotli Azad Jammu and Kashmir, Kotli 11100, Pakistan; zahidmahmood575@uokajk.edu.pk
[3] Department of Electrical Engineering, Capital University of Science and Technology (CUST), Islamabad 44000, Pakistan; mirza.nabeel.jarral@gmail.com
[4] Center for Cybersecurity, Brunno Kessler Foundation, 38123 Trento, Italy
[5] Faculty of Engineering, Free University Bozen-Bolzano, 39100 Bolzano, Italy
* Correspondence: ahmad@fbk.eu (T.A.); attaullah.buriro@unibz.it (A.B.)

Abstract: The exponential growth of the Internet of Things (IoT) has led to the rapid expansion of interconnected systems, which has also increased the vulnerability of IoT devices to security threats such as distributed denial-of-service (DDoS) attacks. In this paper, we propose a machine learning pipeline that specifically addresses the issue of DDoS attack detection in IoT networks. Our approach comprises of (i) a processing module to prepare the data for further analysis, (ii) a dynamic attribute selection module that selects the most adaptive and productive features and reduces the training time, and (iii) a classification module to detect DDoS attacks. We evaluate the effectiveness of our approach using the CICI-IDS-2018 dataset and five powerful yet simple machine learning classifiers—Decision Tree (DT), Gaussian Naive Bayes, Logistic Regression (LR), K-Nearest Neighbor (KNN), and Random Forest (RF). Our results demonstrate that DT outperforms its counterparts and achieves up to 99.98% accuracy in just 0.18 s of CPU time. Our approach is simple, lightweight, and accurate for detecting DDoS attacks in IoT networks.

Keywords: dynamic attribute selection; DDoS attack classification; CICI-IDS-2018 dataset

Citation: Ullah, S.; Mahmood, Z.; Ali, N.; Ahmad, T.; Buriro, A. Machine Learning-Based Dynamic Attribute Selection Technique for DDoS Attack Classification in IoT Networks. *Computers* **2023**, *12*, 115. https://doi.org/10.3390/computers12060115

Academic Editor: Paolo Bellavista

Received: 9 March 2023
Revised: 9 May 2023
Accepted: 26 May 2023
Published: 29 May 2023

1. Introduction

The Internet of Things (IoT) affects our lifestyle, including how we act and behave. It can be seen in the air conditioning that we can control through our smartphones, the E-Health care in which patients wear sensors on their bodies to track their health, and our intelligent watches that track our daily activities. IoT consists of many devices that are connected to a large network. These devices gather and share data. The IoT provided the world with an easy way of operating and monitoring their devices. With time, the use of the internet is growing. Therefore, IoT devices are growing in number. Consider where they are being used to understand how big it they have become. The use of IoT can be seen in industries and health departments. Business is changing in the way the paradigm shift in cloud computing operates because of IoT. This type of dependence of today's world on IoT can result in generating, monitoring, and analyzing big data. The analysis of big data is undoubtedly beneficial.

However, at the same time, numerous security risks from the attacks of malicious bots can cause problems for the protection of IoT devices. With the speedy growth of the IoT, the botnet can easily perform many wider scales of attacks using IoT devices. A malicious bot is a device that has been infected, and that device could be an IoT device. The infected bots are sometimes connected and form botnets. These botnets then perform activities such as DDoS attacks. A DDoS attack is a form of attack in which malicious traffic overloads

the target or associated infrastructure. This is achieved by deploying bots, a network of malware-infected computers and other devices known as zombies, which an attacker may remotely manage, as shown in Figure 1 [1]. It significantly restricts bandwidth and connection, causing all network services to fail [2]. Cloud ecosystems incur the most losses due to service denial and degradation [3]. A primary objective is to impair the availability of resources for legitimate users. Attack traffic in a DDoS attack is difficult to identify due to its resemblance to normal traffic [4]. They behave like ordinary network packets.

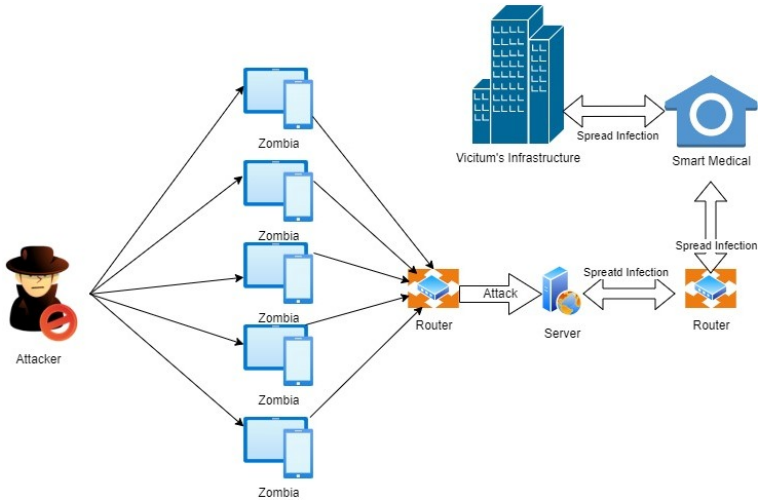

Figure 1. Distributed denial-of-service (DDoS) Attack.

The research focuses on the need for an efficient machine learning-based classification technique to counteract DDoS attacks. The current approaches include solutions based on protocols, trust-based solutions, machine learning, deep learning, SDN, and blockchain technologies. The proposed system comprises three subsystems: preprocessing, feature selection, and detection. The preprocessing subsystem involves collecting and normalizing attributes from traffic. The feature selection subsystem selects the top ten attributes using automatic threshold techniques. The objective is to select a minimum number of attributes to use the limited system's resources and accurately classify attack and normal traffic. The problem statement highlights the need for an efficient system that can identify DDoS attacks quickly while minimizing the impact on system resources.

DDoS attacks pose a significant security threat to IoT devices and can cause disruptions to their regular operations. Machine learning-based approaches have been proposed as effective solutions to detect DDoS attacks in IoT networks. Machine learning algorithms can detect patterns in network traffic and make predictions, which is useful in identifying DDoS attacks. Algorithms such as K-Nearest Neighbor, Decision Tree, and neural networks are commonly used to detect DDoS attacks in IoT networks. The algorithm selection depends on specific requirements and the type of DDoS attack. Ensuring the accuracy of the machine learning model is a challenge that can be addressed by utilizing large and diverse datasets and appropriate evaluation metrics. Traffic filtering, rate limiting, and traffic shaping are other techniques employed to mitigate DDoS attacks and can be used alongside machine learning. Overall, machine learning-based approaches present a promising solution for detecting DDoS attacks in IoT networks, which can prevent disruptions to normal device functioning.

We propose a machine learning-based classification technique to improve the detection of DDoS attacks. The proposed system consists of three phases: (1) preprocessing, (2) feature selection, and (3) detection and presentation system. The top 30 features are initially collected and normalized in the first phase from traffic. In the feature selection phase, the features are chosen using different Random Forest classifier techniques. We only used a selected list of features (i.e., dynamic attribute selection approach) to detect the DDoS attack more efficiently and minimize the training period to detect the attack more efficiently. In the final phase, the traffic is classified as DDoS and Benign traffic.

The following are the main contributions of this work:

- Proposal of a dynamic attribute selection technique for identifying and preventing DDoS attacks in IoT networks, which reduces the number of features from 79 to 30;
- Categorization of the technique into three modules: pre-processing, feature selection, and classification, which employ machine learning algorithms for the dynamic attribute selection module;
- Evaluation of the proposed technique using five different classifiers: Random Forest, Gaussian, Logistic Regression, K-Nearest Neighbor, and Decision Tree;
- Identification of the DT classifier as the best-performing classifier, achieving an accuracy of 99.98% by using only 0.18 s of CPU time;
- Providing an effective approach for detecting and preventing DDoS attacks in IoT networks is critical for ensuring the security of these interconnected systems.

Paper Organization The rest of the paper is organized as follows. Section 2 presents the related work. Section 3 explains the basic idea of our approach. Section 4 reports the experimental results as part of the evaluation of the proposed approach. Section 5 discusses the evaluation results. Section 6 concludes the paper with a summary of our major findings and the potential future research dimensions.

2. Related Work

Cybercriminals frequently use distributed denial-of-service (DDoS) attacks to cause disruptions in computer networks and gain some advantage. To combat these attacks, ongoing efforts in the computing world develop methods to detect and prevent them. This study focuses on the IoT environments and solutions considering specific characteristics of IoT networks. The state-of-the-art approaches can be classified into three categories: protocol-based, trust mechanism-based, and machine learning-based.

Protocol-based solutions enhance security by utilizing existing protocols or creating new ones on top of them. For example, Glissa et al. [5] proposed 6lowPSec, a framework that uses chained message verification codes and improved encryption standards to encrypt packet payloads operating under the adaptive layer's MAC security sublayer. This system can counter denial-of-service attacks, but adding new nodes slows down the system and increases processing time. Wallgren et al. [6] studied the security measures of IoT technologies and demonstrated usual routing attacks on RPL-based 6LoWPAN networks using the Cooja simulator and Contiki operating system. The RPL protocol has internal measures to resist some attacks but is still vulnerable to others. They introduced the heartbeat protocol to eliminate selective forwarding threats by adding the heartbeat protocol to the IPsec capabilities in the IPv6 protocol. Hossain et al. [7] proposed a four-layer biometric architecture for secure communication, but it may result in communication overhead and the use of terminal resources due to its data footprint. Glissa et al. [8] also proposed a security protocol that verifies the validity of each node using hash chain authentication. Pu et al. [9] suggested a lightweight validation approach using a map hash function to transmit frequent acknowledgement packets and a checkpoint node to resist selective relay attacks. However, this method is not more effective for topologies that change frequently. Securing resource-constrained devices against insider assaults is a significant issue for the RPL protocol.

In addressing the 6LoWPAN fragmentation attack, the Border Router (BR) uses encryption for a secure connection. Hossain et al. [10] proposed a technique that uses a SecPAN message authentication code with implicit certificate-based encryption where the BR assigns a temporary address to each node and chooses the parent node based on network position. They omit the temporary address after the establishment of a secure channel. In their approach, duplicate packets or overlapping of the packet due to lack of authentication is possible. Gara et al. [11] proposed a statistical model to detect selective replay attacks by estimating lost packets if the number reaches a certain level, declaring it malicious, and then removing it. Yin et al. [12] proposed a DDoS attack detection method using a software-defined IoT network with a cosine-similarity algorithm. The approach fails to detect DDoS attacks in the case of a large amount of traffic. Sabrina et al. [13] suggested

a reaction-based approach to detect and counter DDoS attacks in the IoT by using metrics such as the number of requests, packet count, and invalid packets, however, a flooding attack is possible and can lead to other attacks. Li et al. [14] proposed an entropy-based technique to counter volumetric DDoS attacks, consisting of processing traffic, calculating entropy, and deciding based on the calculation.

The purpose of IoT devices and networks is to enhance business value by connecting various devices and objects regardless of their resources. As a result, this ecosystem includes many devices with limited storage and processing power. The ecosystem must be taken into account when developing a trust management system. Khan et al. [15] proposed a trust mechanism based on confidence values calculated using a subjective logical approach and the Opinion Triangle (OP) to assess the trust level. The OP considered three traits—trust, mistrust, and anxiety, and used the uncertainty attribute to analyze grey areas. The trust score is calculated based on the trust rating of surrounding nodes, with a low trust rating indicating a suspect node. They recommended countering selective forwarding, sinkhole, and version number attacks. However, this approach requires more objective justification to define the optimum threshold value for the trust evaluation.

Airehrour et al. [16] focused on trust-based solutions to combat black hole attacks in RPL networks. They calculated the trust score per node based on the number of packets sent and received through the parent node, with limitations in its approach. Ahmad et al. [17] proposed indiscriminate nodes to detect black hole attacks. The local decision-making mechanism uses specific criteria to determine if a node is suspect and employs a validation procedure, then analyzes it further. Alaba et al. [18] proposed a mechanism for managing context trust, which uses different trust computing features for various node services and has a dynamic trust score based on the context and condition of the node. The centralized system design reduces network overhead and provides a single point of failure. The author does not explain how the system would scale in large or dense networks.

Diro and Chilamkurti [19] proposed a solution for zero-day attack detection by deploying a distributed deep learning model at the network edge (e.g., fog layer). They used a simple deep feedforward network to make nuanced decisions that cannot perform with traditional machine learning methods. They applied 1-to-n encoding to the NSL-KDD dataset to test their proposed method and showed an accuracy exceeding 99%. However, they did not test their model on edge equipment. They also did not design the model with the performance constraints of the model in mind. Additionally, the NSL-KDD dataset used in their work does not contain data from an IoT network. Meidan et al. [20] proposed a method to detect botnet activity on an IoT network using unsupervised deep learning with deep auto-encoders. They trained the model on non-malicious network traffic and then applied it to anomalies with a DDoS attack. The method of Goodfellow et al. [21] showed a 100% detection rate but did not devise any mitigation technique. Additionally, training a separate detection model for each device on the network would not scale well in diverse IoT environments. Sharma et al. [22] proposed a method for detecting attacks in IoT using Software Defined Wireless Networks (SDN) and cloud, with a deep belief network. The simulation results showed the effectiveness of their method. However, a deep belief network is prone to failure if the input is unclear, a problem in the resource-constrained IoT environment. McDermott et al. [23] tackled the challenge of detecting IoT botnets by constructing a dataset based on a real-life deployment of the Mirai botnet on an IoT testbed built in their laboratory. They trained two different deep-learning models. One was a conventional long short-term memory (LSTM) recurrent neural network (RNN), and the other is a bidirectional LSTM on the dataset.

Bhunia and Gurusamy [24] presented a novel approach for detecting DDoS attacks by utilizing SDN and deploying the detection system close to the network edge. They proposed support vector machines (SVM) in a control plane to achieve the goal of detection close to the network edge. Compared with deep learning models, SVMs are much less computationally demanding as they only have a single activation function, while deep models use multiple activation functions. This characteristic makes SVMs suitable for environments with limited

computational resources, such as onboard IoT devices [25]. Liu et al. [26] presented a defence system and divided it into three subsystems. In the first subsystem, the pre-processing stage was divided into two modules. In the first module, they extracted the properties from incoming data flow using the IoT network intrusion dataset. Then, the data were split into 75% training data and 25% testing data for various machine learning algorithms. They achieved high levels of accuracy while maintaining the efficiency of the IoT intrusion network dataset. They achieved 99% accuracy with the KNN algorithm, while XGBoost had an accuracy rate of 97%. They matched the results of the F1 scores achieved using various machine learning techniques by utilizing all of the dataset's attributes. They used binary classification, and the initial experimental results are promising. However, the data must be normalized to counteract the inaccuracies generated through the LR technique.

State-of-the-art protocol-based solutions summarised in Table 1 have the following limitations: First, most studies have not reported cross-layer integration, and biometric solutions have a large footprint that may not be feasible for restricted devices in IoT. Second, most evaluations are based on simulation and do not consider real-world factors such as noise and signal distribution. Third, the literature has not addressed the heterogeneity of IoT networks and has not investigated the issues of architectural biometrics and multiple technology non-uniformity. Similarly, state-of-the-art trust management approaches in IoT focus on the trust of communication (e.g., network layer) and ignore the limitations of IoT devices. However, trust management solutions require large computational resources that may not be available on IoT devices. These approaches also focus on single-layer solutions and ignore trust difficulties at other layers of the IoT ecosystem, which require cross-layer solutions. State-of-the-art solutions also use simulation for evaluation purposes and ignore real-world scenarios, which are a crucial consideration.

Table 1. Summary of literature review.

References	Approach	Technique	Strengths	Weaknesses
Glissa et al. [5]	Protocol-based	6lowPSec framework plus OS adaptive MAC security sublayer	Encounterung DDOS attack	Adding new nodes slows down the system and increases processing time
Wallgren et al. [6]		Heartbeat capabilities for IPsec in the IPv6 protocol	Resisting routing attacks and eliminates selective forwarding	Vulnerable to other attacks
Hossain et al. [7]		Four-layer biometric architecture	Secure communication	Communication overhead and the use of terminal resources due to its data footprint
Pu et al. [9]		Lightweight validation approach using a map hash function	Securing resource-constrained devices against insider assaults	Not more effective for topologies that change frequently
Hossain et al. [7]		SecPAN: Message Authentication code and implicit certificate-based encryption	Establishment of a secure channel with reduced communication latency and energy consumption	Duplicate packets or overlapping fragments
Yin et al. [12]		Software-defined IoT network with a cosine-similarity algorithm	DDoS attack detection method	Fails to detect DDoS attacks in case of large traffic
Sabrina et al. [13]		Suggesting a reaction-based approach	Detecting and countering DDoS attacks in the IoT	A flooded attack is possible
Li et al. [14]		Entropy-based technique	Countering volumetric DDoS attacks	The server still answers to malicious attacks

Table 1. *Cont.*

References	Approach	Technique	Strengths	Weaknesses
Khan et al. [15]	Trust-based	Confidence values are calculated using a subjective logical approach and the Opinion Triangle (OP)	Recommended countering selective forwarding, sinkhole, and version number attacks	Trust evaluation needs more objective justification for an optimum threshold value
Airehrour et al. [16]		Calculating the trust score per node based on the number of packets sent and received through the parent node	Combatting black hole attacks in RPL networks	Not to mention the utilization of trust value to prevent a blackhole attack The minimum lifespan of battery-powered nodes
Ahmad et al. [17]		Local decision-making mechanism	Indiscriminate nodes to detect black hole attacks	No authentication on network layer
Alaba et al. [18]		Context trust using different trust features and dynamic trust based on the context and condition of the node	Reduces network overhead	Single point of failure, no scaling discussed in large or dense networks
Diro and Chilamkurti [19]	ML based	Deployment of a distributed deep learning model at the network edge (e.g., fog layer)	Solution for zero-day attack detection	Command processing in a signature-based system is expensive and cannot effectively handle the latest threats
Meidan et al. [20]		Using unsupervised deep learning with deep auto-encoders	Detect botnet activity on an IoT network	Difficult to capture some normal behaviour of IoT devices
Goodfellow et al. [21]		Deep learning	Showed a 100% detection rate	Did not devise any mitigation technique and not scalable in diverse IoT environments
Sharma et al. [22]		Software Defined Wireless Networks (SDN) and cloud	Detecting attacks in IoT	Prone to failure if the input is unclear, a problem in the resource-constrained IoT
McDermott et al. [23]		A bidirectional LSTM on the dataset	Detection of a botnet in consumer IoT and networks	A simulated approach only and detection for a range of attacks is not possible
Bhunia and Gurusamy [24]		Utilizing SDN close to the network edge and SVM in a control plane	Suitable for environments with limited computational IoT devices	A simulated base for traffic in a mini-net that does not show the behaviour of real IoT devices.
Liu et al. [26]		KNN algorithm and XGBoost	Achieved 99% accuracy with the KNN and 97% with XGBoost	For large datasets, SVM is not recommended

The limitation of the above state-of-the-art machine learning approaches is the selection of all the features in the dataset for classification. However, filtering out unimportant features from the dataset is necessary as they increase time and space complexity [27]. Therefore, an efficient machine learning-based classification technique is required, which selects a minimum number of attributes to use a limited system's resources while classifying the attack and normal traffic. In this paper, we explore a technique to achieve the same or even better accuracy by considering fewer features and using an efficient and intelligent machine learning DDoS detection technique.

3. Proposed Approach

The proposed system consists of three modules: preprocessing, feature selection, detection and presentation. Features are collected and normalized from traffic in the Preprocessing Module at the beginning. The top 30 features are selected and formed into a set. The features are chosen using different Random Forest classifier techniques in the feature selection module.

Based on the analysis, existing research uses all of the features in the dataset. We attempt to minimize the training period to detect the attack more efficiently. Finally, the detection and presentation module is responsible for classifying traffic data as DDoS and normal.

Algorithm 1 demonstrates the algorithm of the proposed method. The dynamic attribute selection approach applied to feature selection involves ranking the features based on their relevance and selecting the top-ranking features. This method uses a scoring metric to evaluate the importance of each feature in the context of the current model and then update the feature set as the model evolves.

3.1. Preprocessing Module

Before splitting the CICI-IDS-2018 dataset, we shuffle it. After shuffling, the dataset is divided into a training dataset and a testing dataset: 80% split is used for training and 20% for testing. Overall, the data packets are rich in information that could be exploited for classification. Such attributes help to discriminate between legitimate traffic and malicious traffic. In the preprocessing module, the dataset is first cleaned—by cleaning, we refer to the blank spaces in the dataset, null values in the data set and the duplicate values are removed. After cleaning the dataset, we normalise it to a standard scale of 0 and 1. The benign samples are labelled 0, and the DDoS samples are labelled 1.

3.2. Feature Selection Module

Selecting features is the process of removing irrelevant and unproductive features that will improve classifiers' learning capability and hence, predictability. One of the most widely used machine learning methods used for feature selection is Random Forest. They are often quite successful because they have good predictive performance, practically little overfitting, and are simple to understand. The fact that it is simple to deduce the relevance of each variable in determining the tree contributes to its interpretability. Technically speaking, we can quickly determine which features can influence the accuracy of the model. Random Forest feature selection falls under the area of embedded techniques. Filter and wrapper methods are combined in embedded methods. Algorithms with built-in feature selection techniques are used to implement them. The following are some of the advantages of embedded methods: 1. High level of accuracy; 2. Better generalization; and 3. Easy to understand.

3.3. Detection and Presentation Module

The test dataset is utilized as input in the detection and prevention subsystem and presented to the feature selection subsystem to assess the best feature collection. The classifiers map the test data from the trained dataset's optimum attribute set across the feature vector the classifiers developed during training. The data are then classified into DDoS and benign requests using classifiers. Our classification toolbox contains multiple, simple yet state-of-the-art machine learning classifiers including Random Forest (RF), Gaussian, Logistic Regression, K-Nearest Neighbor (KNN), and Decision Tree (DT), to classify between the legitimate and illegitimate (DDoS) samples.

Algorithm 1 Proposed DDoS Detection System

Initialization:
Raw Traffic Data X_{raw}
Predicted DDoS labels Y_{pred}
$X \leftarrow \text{extract\_features}(X_{\text{raw}})$ {Preprocessing}
$X_{\text{norm}} \leftarrow \text{normalize}(X)$ {Preprocessing}
$X_{\text{sel}} \leftarrow \text{select\_features}(X_{\text{norm}})$ {Feature selection}
$X_{\text{sel\_top30}} \leftarrow \text{select\_top30}(X_{\text{sel}})$ {Feature selection}
$Y_{\text{pred}} \leftarrow \text{detect\_ddos}(X_{\text{sel\_top30}})$ {Detection}
$\text{present\_results}(Y_{\text{pred}})$ {Presentation}
Preprocessing module:
$\text{proposed\_system}(X_{\text{raw}})$
$X \leftarrow \text{extract\_features}(X_{\text{raw}})$
Procedure: extract_features
Input: Raw Traffic Data X_{raw}
Perform feature extraction on raw traffic data X_{raw}
return Feature matrix X
$X_{\text{norm}} \leftarrow \text{normalize}(X)$
Procedure: normalize
Input: Feature matrix X
Normalize the feature matrix X
return Normalized feature matrix X_{norm}
Feature Selection Module:
$X_{\text{sel}} \leftarrow \text{select\_features}(X_{\text{norm}})$
$X_{\text{sel\_top30}} \leftarrow \text{select\_top30}(X_{\text{sel}})$
Procedure: select_features
Input: Normalized feature matrix X_{norm}
Perform feature selection on normalized feature matrix X_{norm}
return Selected feature matrix X_{sel}
Procedure: select_top30
Input: Selected feature matrix X_{sel}
Select top 30 features from selected feature matrix X_{sel} using Random Forest Classifier techniques
return Top 30 feature matrix $X_{\text{sel\_top30}}$
Detection and Presentation Module:
$Y_{\text{pred}} \leftarrow \text{detect\_ddos}(X_{\text{sel\_top30}})$
$\text{present\_results}(Y_{\text{pred}})$
Procedure: detect_ddos
Input: Top 30 feature matrix $X_{\text{sel\_top30}}$
Use a DDoS detection model to predict the presence of DDoS attacks using the top 30 selected features $X_{\text{sel\_top30}}$
return The predicted labels y_{pred}
$\text{present\_results}(Y_{\text{pred}})$
Present the results of the DDoS detection algorithm, including the predicted labels Y_{pred} and any other relevant information. The presentation can be a report, graphical representation or any other suitable method.

4. Experimental Evaluation

Our test-bed setup comprised a computer system running on a 64-bit Windows 10 Operating System, specifically equipped with machine learning capabilities. These capabilities are integrated into the operating system, allowing it to learn from data, recognize patterns, and make predictions based on what it has learned. This setup provides an excellent platform for experimenting with and testing machine learning algorithms. Moreover, the Windows 10 OS provides a stable and secure environment for running tests, ensuring that the results are accurate and dependable.

4.1. Preprocessing

In this research, we used the CICI-IDS-2018 dataset. It has a total number of 79 features. In this section, the blank spaces are removed so that the proposed model can accept the dataset. For this, we replace the blank spaces with an underscore. Then, we assign labels to the dataset "benign" and "Malicious". We began our analysis with the identification and removal of null values and redundant observations (duplicates). After this preprocessing, we observed that we are left with an unbalanced dataset containing 40% benign and 58% malicious DDoS observations. In Figure 2, we illustrate the observations available in the dataset.

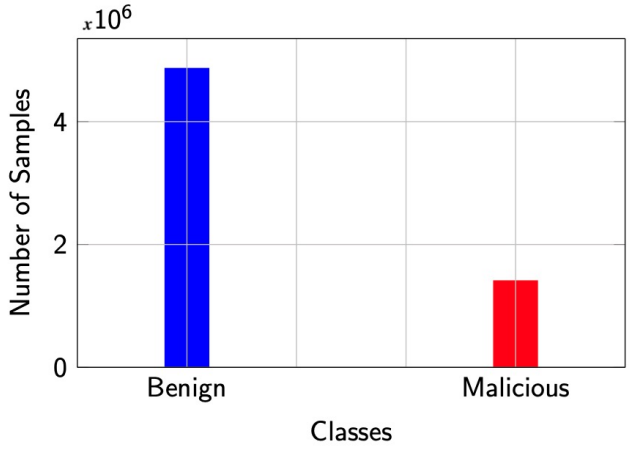

Figure 2. Unbalanced Distribution (CICIDS-IDS-2018) and Attacks Scenario.

Classifiers' training on an imbalanced dataset could lead to biased predictions [28]—the classifier could learn to prioritize the majority class and make biased predictions, with lower recall—so it becomes difficult for the classifier to make itself learn well on fewer samples and classify accurately, thus potentially resulting in lower recall, and finally, overfitting—overfitting is common in imbalanced data classification tasks as the classifier does not generalize well. Hence, balancing the data becomes more important to address all these concerns. We either had to chose between oversampling (increasing the number of samples of minority class using SMOTE (https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html?highlight=smote#imblearn.over_sampling.SMOTE (accessed on 25 May 2023)) or Generative Adversarial Network (GAN) [29]) or undersampling (decreasing the samples of majority class to match the number of samples of minority class). We preferred undersampling and created a balanced dataset containing 50% benign and 50% malicious observations, as shown in Figure 3.

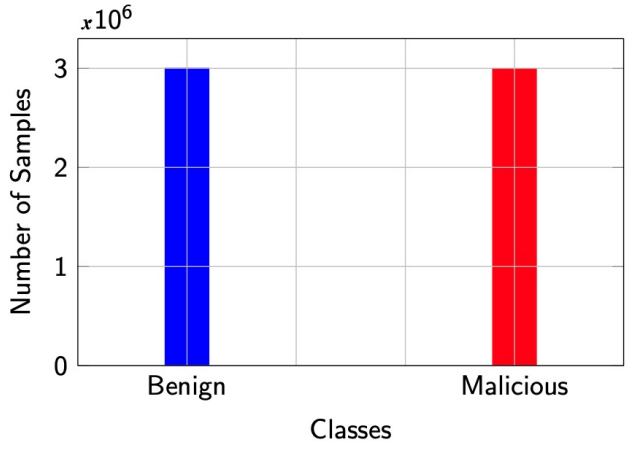

Figure 3. Balanced Distribution.

After balancing the dataset, we shuffled the data and shook the dataset. To pass the data to our classification model, we converted our categorical values into a new categorical column and assigned a binary value of '1' or '0'. In our case, the '1' label represents DDoS flow (malicious samples) and '0' represents benign flow.

4.2. Features Selection

RF—an ensemble technique, could be exploited for feature selection. RF grows multiple decision trees on random subsets of features and combines them to make predictions. The importance of each feature depends upon its contribution to the accuracy. A random subset of features is selected from the available features at each split to reduce the correlation between the trees in the forest, making it more diverse. The selection of the features subset is random, which creates more variations and helps to reduce overfitting. The importance of each feature is calculated based on how much the decision tree reduces the impurity. The Gini impurity metric calculates the impurity of the split. The lesser the impurity, the higher the importance of the feature. The feature importance scores are then aggregated to determine the overall importance of the feature. Finally, the algorithm selects the top features based on the threshold value. Figure 4 shows the importance of the feature. The first two features are overfitted, and the below 46 features are less important. We chose the most important 30 features among them.

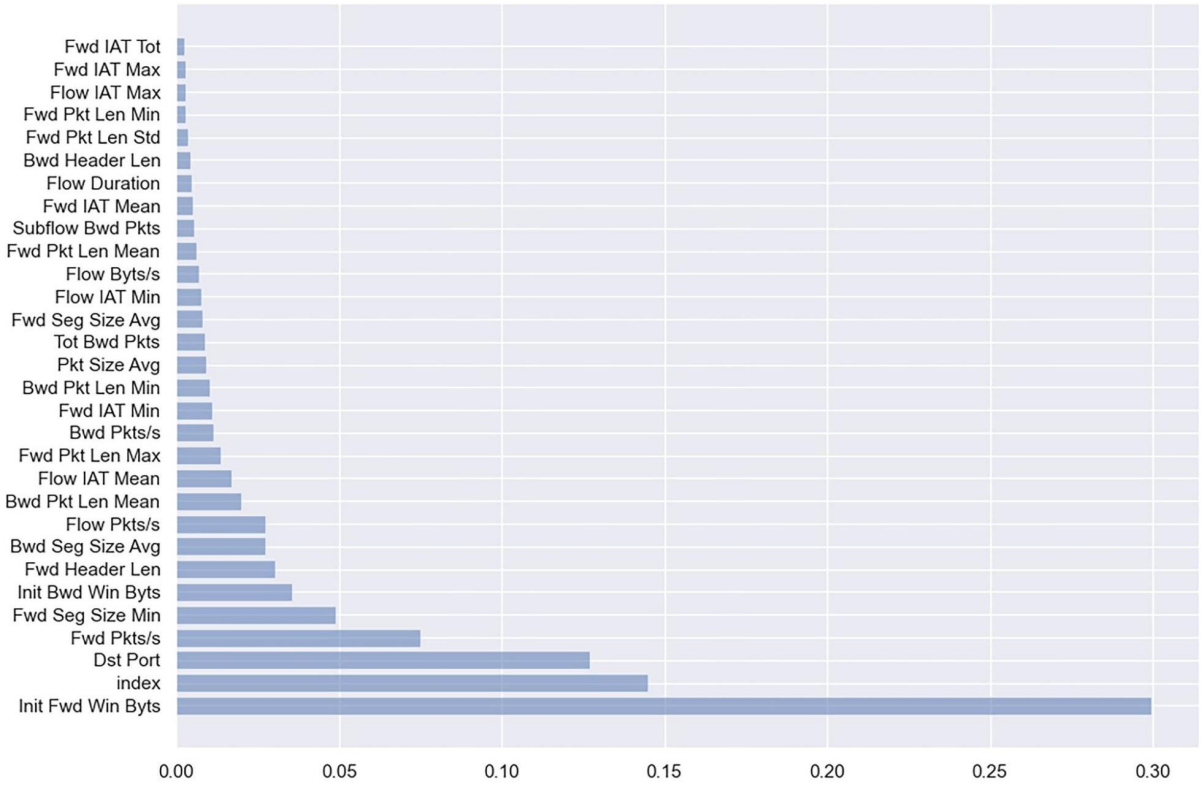

Figure 4. Random Forest Feature Importance.

After removing the redundant and less productive features, we managed to obtain a subset of vectors with the 30 best features to evaluate our chosen machine learning model. Correlation matrix—a graphical representation illustrating the relationship of the features—a positive and negative value indicates a positive and negative correlation between them. Technically speaking, a positive correlation indicates the increase in the value of a feature with respect to the increase in the value of another feature and a decrease in the case of a negative correlation. We show the correlation of different attributes in Figure 5.

The selection of a machine learning classifier depends upon various factors such as the problem the algorithm is expected to solve, the size of the dataset, and the time they

take for training and testing, etc. Knowing which classifier will work well on a particular dataset is practically impossible. To this end, our classification toolbox consists of five simple yet state-of-the-art machine learning classifiers, namely, RF, Gaussian Naive Bayes (GNB), Logistic Regression (LR), K-Nearest Neighbor (KNN), and Decision Tree (DT). The predictive model based on the Decision Tree classification algorithm outperformed all its counterparts and achieved 99.98% accuracy within just 0.18 s.

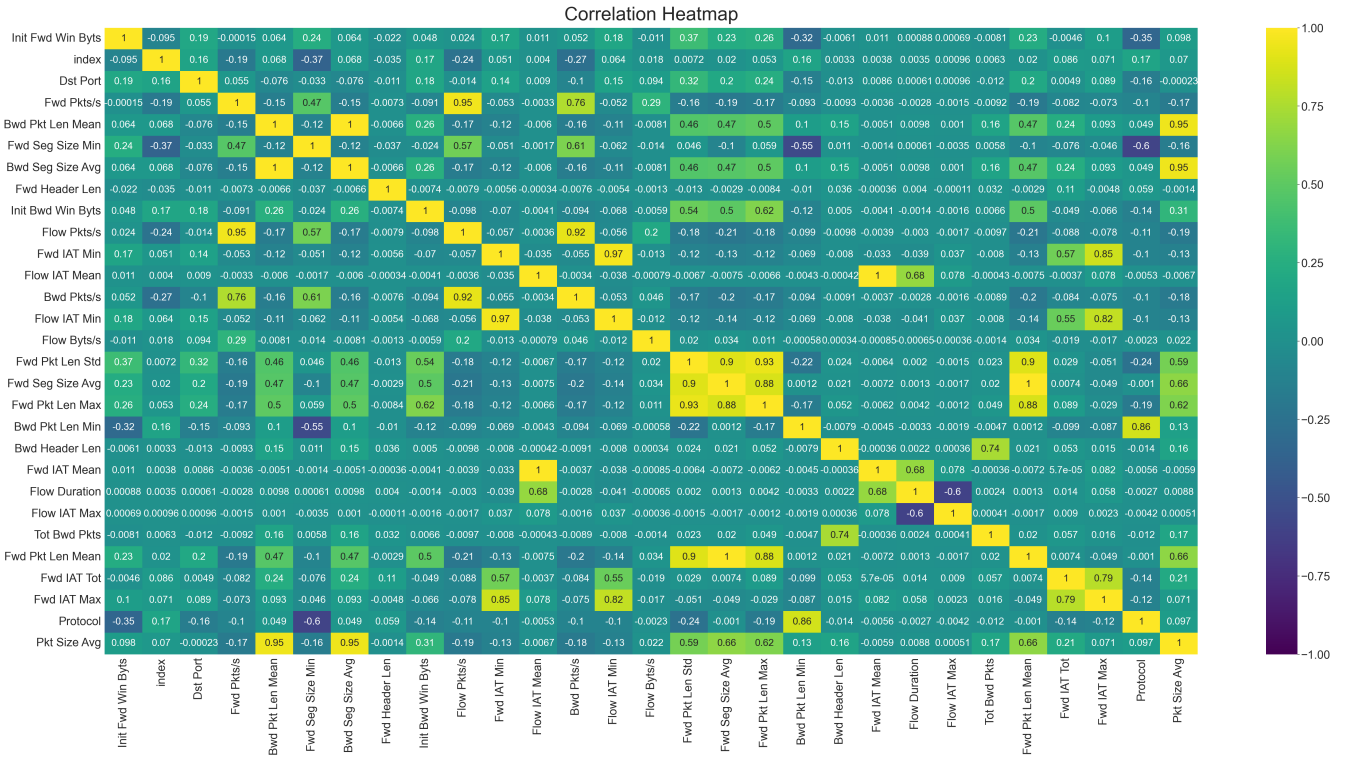

Figure 5. Correlation of Features.

4.3. Success Metric

The success metric can be defined in terms of TPR, FPR, and accuracy based on the solved problem. Our success metric includes the measurement of True Positive Rate (TPR), False Positive Rate (FPR), and Accuracy. We define below these parameters:

- **True Positive Rate (TPR)** is defined as the ratio of correctly predicted positive instances to the total number of positive instances;
- **False Positive Rate (FPR)** is the ratio of incorrectly predicted positive instances to the total number of negative instances;
- **Accuracy** is the ratio of correctly predicted instances to the total number of instances.

4.4. Comparison of Different Classifiers

The study focused on minimising the number of features to reduce the computational complexity without compromising accuracy. The classifiers are trained on thirty productive features selected by the algorithm. The accuracy of 92% is achieved on the selected subset of 30 features, whereas on the original 79 features, it is 86%. Thus, our approach reduces the computational resources and improves the accuracy (from 86% to 92%). For quick reference, we refer readers to Table 2.

From the confusion matrix (see Figure 6), it is evident that 437,066 are correctly classified as benign and 118,394 as malicious DDoS samples; it can be seen that 437,066 are correctly identified as benign, and 118,394 are correctly identified as DDoS samples. The accuracy of 99% is achieved with the KNN classifier. The obtained recall and precision are as high as 99%. The CPU time of the LR model was 0.52 s.

Table 2. Comparison of ML Classifier results.

ML Classifier	All Features				Reduced Features			
	Accuracy	CPU Time		Model Size	Accuracy	CPU Time		Model Size
		Training	Inference Test Set			Training	Inference Test Set	
RF	99.98%	1816.35 s	3.65 s	3.30 MB	99.93%	666.56 s	1.63 s	2.33 MB
GNB	57.97%	10.97 s	1.83 s	4.22 KB	57.82%	2.31 s	0.61 s	2.05 KB
LR	78.8%	150.82 s	1.12 s	2.52 KB	61.78%	57.21 s	0.12 s	1.44 KB
KNN	99.98%	5602.87 s	5.7 s	2.59 GB	99.94	1899 s	2.3 s	836 MB
DT	99.99%	231.84 s	1.08 s	47.8 KB	99.98%	79.86 s	0.18 s	25.7 KB

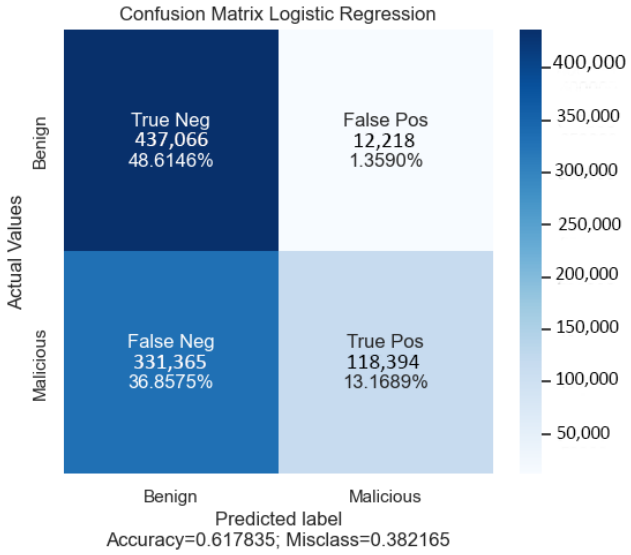

Figure 6. Logistic Regression Confusion Matrix.

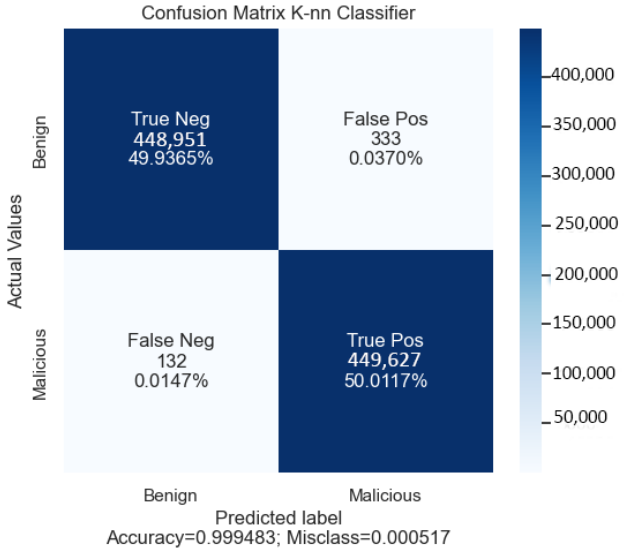

Figure 7. KNN Confusion Matrix.

Similarly, for our chosen KNN classifier, the obtained accuracy was 99.94%. As depicted in the confusion matrix (see Figure 7), 448,951 benign and 449,687 malicious samples are correctly classified as benign and malicious, respectively.

With DT as a classifier, we report our obtained accuracy of 99.98%. The computed recall of this classifier is 89% and the precision is 99%. The CPU time of the model was only 0.18 s. The confusion matrix for this classifier (DT) is shown in Figure 8. The confusion matrix shows the correctly classified benign and DDoS (malicious) classes.

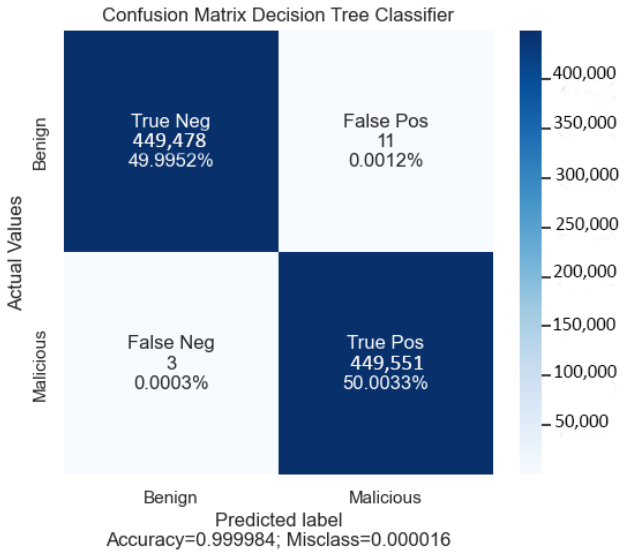

Figure 8. Decision Tree Confusion Matrix.

Interestingly, Random Forest, which grows multiple decision trees and creates of forest for performing regression and classification tasks, correctly classified 449,478 benign and 449,551 malicious samples (as seen in Figure 9. Thus, an overall accuracy of 99.93% is achieved with this classifier.

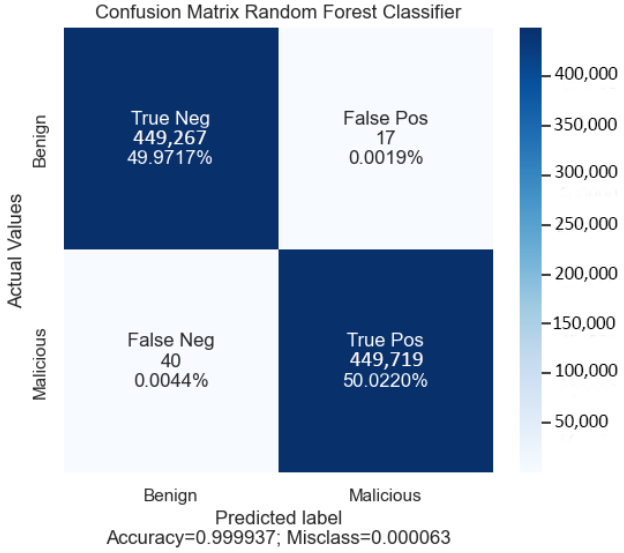

Figure 9. Random Forest Confusion Matrix.

Finally, our last chosen GaussianNB classifier was able to classify 449,767 benign samples as benign and 449,719 malicious samples as DDoS. It has the least accuracy among the classifiers used. The recall is 75.97%, and the precision is 99%. The CPU time of the model was 1.31 s. The confusion matrix is shown in Figure 10.

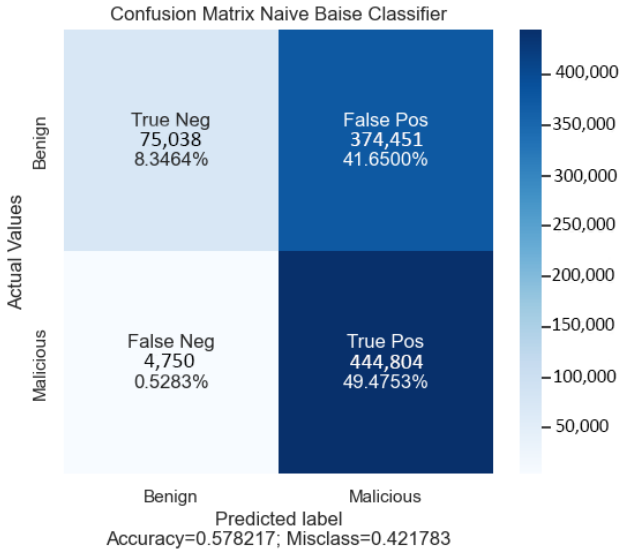

Figure 10. GaussianNB Forest Confusion Matrix.

5. Discussion

Distributed denial-of-service (DDoS) attacks have continuously been a significant threat to network security. As the frequency and complexity of these attacks grow exponentially, it becomes important to develop effective detection mechanisms to prevent unwanted network disconnectivity.

This paper aims to develop a machine-learning-based approach to detect DDoS attacks from network traffic effectively. To evaluate our approach, we exploited CICI-IDS-2018. Each of the observations in this dataset has a dimension of 79 features. We applied preprocessing, such as removing blank spaces and null values, to prepare the dataset before applying the model. We also balanced the dataset using the undersampling technique to avoid bias. Further, we considered reducing the dimensions of the observations by applying hybrid feature selection using Random Forest. The algorithm returned the list of the top 30 productive features used for analysis.

The reasons for the highest accuracy being attained by RF and DT classifiers are their robustness, ability to handle non-linear relationships, ensemble learning, and interpretability. These factors make them suitable for detecting DDoS attacks in network traffic and other complex classification tasks.

Overall, this study significantly contributes to machine learning-based DDoS attack detection. The findings emphasize the importance of effective preprocessing and feature selection in developing accurate and efficient models. The results provide essential insights for practitioners and researchers working in network security.

Comparative Analysis

We compared our proposed approach with the state-of-the-art approaches from the literature. The following are the four existing approaches used for this analysis.

- In [30], the increasing data generation and internet connectivity have necessitated a machine learning intrusion detection system (IDS) for security purposes. However, using a single learning model may not effectively capture the unique patterns of attacks. The authors propose "BDHDLS", which uses behavioural and content features to address this. Each deep learning model focuses on learning the unique data distribution in one cluster. This approach improves the detection rate of attacks, and big data techniques and parallel training are used to reduce model construction time;
- In [31], the authors propose a joint optimization algorithm that uses particle swarm optimization (PSO) and genetic operators to optimize the Deep Belief Network (DBN) for intrusion detection classification. The proposed algorithm improves the

classification accuracy and detection time of the DBN-IDS compared with other optimization algorithms;

- Machine learning techniques have been used to develop intrusion detection systems based on anomaly detection, and the KDD dataset is commonly used to evaluate such systems. In [32], the authors propose a Convolutional Neural Network (CNN) model for the CSE-CIC-IDS 2018 dataset, which contains the most up-to-date common network attacks. The CNN model outperforms the dataset's Recurrent Neural Network (RNN) model. The authors also suggested ways to improve its performance further;
- Finally, in [33], the authors discuss the importance of intrusion detection systems in mitigating network attacks and how deep learning and machine learning are used to develop an effective system. The authors propose a Convolutional Recurrent Neural Network (CRNN) as a DL-based hybrid ID framework that predicts and classifies malicious cyberattacks in the network. The proposed HCRNNIDS outperforms current ID methodologies, achieving a high malicious attack detection rate accuracy of up to 97.75% on the CSE-CIC-DS2018 dataset.

Table 3 compares the proposed machine learning approach for DDoS attack detection in IoT networks with the four existing approaches from the literature. It can be seen that our proposed approach achieves the highest accuracy (99.98%) with the lowest training time (0.18 s). The other approaches achieve lower accuracy with greater training times and computational complexity. Overall, the proposed approach is simple, lightweight, and accurate for DDoS attack detection in IoT networks.

Table 3. Comparative analysis of the proposed approach.

Study	Dataset	Method	Feature Selection Method	Training Time (s)	Accuracy (%)
Wei Zhong et al. [30]	CICIDS2017	BDHDLS	PCA	45 s–60 (approx.)	99.00
Peng et al. [31]	CSE-CIC-IDS2018	DBN	Genetic Algorithm	1800–3600	95.00
Kim et al. [32]	CSE-CIC-IDS2018	CNN	Manual Feature Extraction	300–900	96.00
Khan [33]	CSE-CIC-IDS2018	HCRNNIDS	Random Feature Selection	200–250	97.25
Our Model	CSE-CIC-IDS2018	DT	Dynamic Attribute Selection	0.18	99.98

6. Conclusions

DDoS attack detection from network traffic is a crucial aspect of network security, and machine learning-based classification techniques have been shown to be effective in improving this process. This study proposes a machine learning-based technique for detecting DDoS attacks which comprises three modules: preprocessing, attribute selection, and a detection and prevention system. Technically speaking, the incoming traffic attributes are first normalized on a standard scale during the preprocessing phase. The Random Forest technique was then employed to select the most productive features, yielding the 30 most productive features out of 79. The approach was evaluated on a publicly available dataset, and its performance was computed based on accuracy. The results indicated that the proposed technique achieved an accuracy of >99% with Random Forest (RF) and DT classifiers proving their robustness to noise and overfitting again. It has been empirically found that reducing the number of features in the dataset and using machine learning techniques to find important features leads to better results in detecting DDoS attacks. The effectiveness of the proposed technique is attributed to the use of robust machine learning algorithms and effective preprocessing techniques.

Overall, the findings of this study highlight the potential of machine learning-based techniques in improving the detection of DDoS attacks in network traffic. The proposed technique, which combines preprocessing, feature selection, and classification algorithms, has shown promising results in accurately detecting DDoS attacks. Further research can explore the generalizability of the proposed technique to other datasets and its effectiveness in real-world scenarios.

As future work, it is important to continue exploring tiny machine learning in the smart Internet of Things (IoT) environment. This includes investigating how these lightweight and efficient algorithms can enhance the security of IoT-based smart systems, particularly in resource-constrained environments where traditional machine learning algorithms may not be practical. Furthermore, there may be opportunities to leverage tiny machine learning to enable more intelligent decision-making at the network's edge, which is critical for many IoT applications. This includes developing new algorithms and techniques to process data locally and in real time, enabling IoT devices to respond quickly to changing conditions and make decisions that optimize performance and efficiency. By continuing to invest in developing and deploying tiny machine-learning techniques in the IoT space, we can ensure that these systems remain secure, reliable, and capable of meeting the complex demands of modern applications. This will be crucial as IoT continues to evolve and become more integrated into our daily lives, with implications for everything from healthcare and transportation to energy and manufacturing.

Author Contributions: S.U.—Conceptualization, Data curation, Software, Writing—original draft, Investigation, Validation and Visualization; Z.M.—Methodology, Supervision, Resources, Writing—original draft, and Visualization; N.A.—Software, Investigation, Writing—original draft, and Validation; T.A.—Writing—original draft, Writing—review and editing, Funding acquisition, Validation, and Visualization; A.B.—Methodology, Formal Analysis, Resources, Writing—original draft, Validation, and Investigation. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Available upon request.

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

References

1. Agrawal, N.; Tapaswi, S. Defense mechanisms against DDoS attacks in a cloud computing environment: State-of-the-art and research challenges. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 3769–3795. [CrossRef]
2. Banitalebi Dehkordi, A.; Soltanaghaei, M.; Boroujeni, F.Z. The DDoS attacks detection through machine learning and statistical methods in SDN. *J. Supercomput.* **2021**, *77*, 2383–2415. [CrossRef]
3. Fazeldehkordi, E.; Owe, O.; Ramezanifarkhani, T. A language-based approach to prevent DDoS attacks in distributed financial agent systems. In Proceedings of the Computer Security: ESORICS 2019 International Workshops, IOSec, MSTEC, and FINSEC, Luxembourg, 26–27 September 2019; Revised Selected Papers 2; Springer: Cham, Switzerland, 2020; pp. 258–277.
4. Cheng, J.; Li, J.; Tang, X.; Sheng, V.S.; Zhang, C.; Li, M. A novel DDoS attack detection method using optimized generalized multiple kernel learning. *arXiv* **2019**, arXiv:1906.08204.
5. Glissa, G.; Meddeb, A. 6LowPSec: An end-to-end security protocol for 6LoWPAN. *Ad Hoc Netw.* **2019**, *82*, 100–112. [CrossRef]
6. Wallgren, L.; Raza, S.; Voigt, T. Routing attacks and countermeasures in the RPL-based internet of things. *Int. J. Distrib. Sens. Netw.* **2013**, *9*, 794326. [CrossRef]
7. Hossain, M.S.; Muhammad, G.; Rahman, S.M.M.; Abdul, W.; Alelaiwi, A.; Alamri, A. Toward end-to-end biomet rics-based security for IoT infrastructure. *IEEE Wirel. Commun.* **2016**, *23*, 44–51. [CrossRef]
8. Glissa, G.; Meddeb, A. 6LoWPAN multi-layered security protocol based on IEEE 802.15.4 security features. In Proceedings of the 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), Valencia, Spain, 26–30 June 2017; pp. 264–269.
9. Pu, C.; Lim, S. A light-weight countermeasure to forwarding misbehavior in wireless sensor networks: Design, analysis, and evaluation. *IEEE Syst. J.* **2016**, *12*, 834–842. [CrossRef]
10. Hossain, M.; Karim, Y.; Hasan, R. Secupan: A security scheme to mitigate fragmentation-based network attacks in 6lowpan. In Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy, Tempe, AZ, USA, 19–21 March 2018; pp. 307–318.
11. Gara, F.; Saad, L.B.; Ayed, R.B. An intrusion detection system for selective forwarding attack in IPv6-based mobile WSNs. In Proceedings of the 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), Valencia, Spain, 26–30 June 2017; pp. 276–281.
12. Yin, D.; Zhang, L.; Yang, K. A DDoS attack detection and mitigation with software-defined Internet of Things framework. *IEEE Access* **2018**, *6*, 24694–24705. [CrossRef]

13. Sicari, S.; Rizzardi, A.; Miorandi, D.; Coen-Porisini, A. REATO: REActing TO Denial of Service attacks in the Internet of Things. *Comput. Netw.* **2018**, *137*, 37–48. [CrossRef]
14. Li, J.; Liu, M.; Xue, Z.; Fan, X.; He, X. RTVD: A real-time volumetric detection scheme for DDoS in the Internet of Things. *IEEE Access* **2020**, *8*, 36191–36201. [CrossRef]
15. Khan, Z.A.; Herrmann, P. A trust based distributed intrusion detection mechanism for internet of things. In Proceedings of the 2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA), Taipei, Taiwan, 27–29 March 2017; pp. 1169–1176.
16. Airehrour, D.; Gutierrez, J.; Ray, S.K. A lightweight trust design for IoT routing. In Proceedings of the 2016 IEEE 14th International Conference on Dependable, Autonomic and Secure Computing, 14th International Conference on Pervasive Intelligence and Computing, 2nd International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), Auckland, New Zealand, 8–12 August 2016; pp. 552–557.
17. Ahmed, F.; Ko, Y.B. Mitigation of black hole attacks in routing protocol for low power and lossy networks. *Secur. Commun. Netw.* **2016**, *9*, 5143–5154. [CrossRef]
18. Alaba, F.A.; Othman, M.; Hashem, I.A.T.; Alotaibi, F. Internet of Things security: A survey. *J. Netw. Comput. Appl.* **2017**, *88*, 10–28. [CrossRef]
19. Diro, A.A.; Chilamkurti, N. Distributed attack detection scheme using deep learning approach for Internet of Things. *Future Gener. Comput. Syst.* **2018**, *82*, 761–768. [CrossRef]
20. Meidan, Y.; Bohadana, M.; Mathov, Y.; Mirsky, Y.; Shabtai, A.; Breitenbacher, D.; Elovici, Y. N-baiot—Network-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Comput.* **2018**, *17*, 12–22. [CrossRef]
21. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
22. Sharma, P.K.; Singh, S.; Park, J.H. OpCloudSec: Open cloud software defined wireless network security for the Internet of Things. *Comput. Commun.* **2018**, *122*, 1–8. [CrossRef]
23. McDermott, C.D.; Majdani, F.; Petrovski, A.V. Botnet detection in the internet of things using deep learning approaches. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.
24. Bhunia, S.S.; Gurusamy, M. Dynamic attack detection and mitigation in IoT using SDN. In Proceedings of the 2017 27th International Telecommunication Networks and Applications Conference (ITNAC), Melbourne, Australia, 22–24 November 2017; pp. 1–6.
25. Ullah, S.; Ahmad, T.; Buriro, A.; Zara, N.; Saha, S. TrojanDetector: A Multi-Layer Hybrid Approach for Trojan Detection in Android Applications. *Appl. Sci.* **2022**, *12*, 10755. [CrossRef]
26. Liu, Z.; Thapa, N.; Shaver, A.; Roy, K.; Yuan, X.; Khorsandroo, S. Anomaly detection on iot network intrusion using machine learning. In Proceedings of the 2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD), Durban, South Africa, 6–7 August 2020; pp. 1–5.
27. Buriro, A.; Buriro, A.B.; Ahmad, T.; Buriro, S.; Ullah, S. MalwD&C: A Quick and Accurate Machine Learning-Based Approach for Malware Detection and Categorization. *Appl. Sci.* **2023**, *13*, 2508.
28. He, H.; Garcia, E.A. Learning from Imbalanced Data. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1041–4347.
29. Buriro, A.; Ricci, F.; Crispo, B. SwipeGAN: Swiping Data Augmentation Using Generative Adversarial Networks for Smartphone User Authentication. In Proceedings of the 3rd ACM Workshop on Wireless Security and Machine Learning, Abu Dhabi, United Arab Emirates, 28 June–2 July 2021; pp. 85–90.
30. Zhong, W.; Yu, N.; Ai, C. Applying big data based deep learning system to intrusion detection. *Big Data Min. Anal.* **2020**, *3*, 181–195. [CrossRef]
31. Wei, P.; Li, Y.; Zhang, Z.; Hu, T.; Li, Z.; Liu, D. An optimization method for intrusion detection classification model based on deep belief network. *IEEE Access* **2019**, *7*, 87593–87605. [CrossRef]
32. Kim, J.; Shin, Y.; Choi, E. An intrusion detection model based on a convolutional neural network. *J. Multimed. Inf. Syst.* **2019**, *6*, 165–172. [CrossRef]
33. Khan, M.A. HCRNNIDS: Hybrid convolutional recurrent neural network-based network intrusion detection system. *Processes* **2021**, *9*, 834. [CrossRef]