telecom

MDPI

Article

Feature-Selection-Based DDoS Attack Detection Using AI Algorithms

Muhammad Saibtain Raza [1,*], Mohammad Nowsin Amin Sheikh [1], I-Shyan Hwang [1,*] and Mohammad Syuhaimi Ab-Rahman [2]

[1] Department of Computer Science and Engineering, Yuan Ze University, Taoyuan 32003, Taiwan; s1089104@mail.yzu.edu.tw
[2] Electrical and Electronic Engineering Department, Universiti Kebangsaan Malaysia, Bangi 43600, Selangor, Malaysia; syuhaimi@ukm.edu.my
* Correspondence: s1116042@mail.yzu.edu.tw (M.S.R.); ishwang@saturn.yzu.edu.tw (I.-S.H.)

Abstract: SDN has the ability to transform network design by providing increased versatility and effective regulation. Its programmable centralized controller gives network administration employees more authority, allowing for more seamless supervision. However, centralization makes it vulnerable to a variety of attack vectors, with distributed denial of service (DDoS) attacks posing a serious concern. Feature selection-based Machine Learning (ML) techniques are more effective than traditional signature-based Intrusion Detection Systems (IDS) at identifying new threats in the context of defending against distributed denial of service (DDoS) attacks. In this study, NGBoost is compared with four additional machine learning (ML) algorithms: convolutional neural network (CNN), Stochastic Gradient Descent (SGD), Decision Tree, and Random Forest, in order to assess the effectiveness of DDoS detection on the CICDDoS2019 dataset. It focuses on important measures such as F1 score, recall, accuracy, and precision. We have examined NeTBIOS, a layer-7 attack, and SYN, a layer-4 attack, in our paper. Our investigation shows that Natural Gradient Boosting and Convolutional Neural Networks, in particular, show promise with tabular data categorization. In conclusion, we go through specific study results on protecting against attacks using DDoS. These experimental findings offer a framework for making decisions.

Keywords: SDN; DDoS attack; feature selection; machine learning techniques

Citation: Raza, M.S.; Sheikh, M.N.A.; Hwang, I.-S.; Ab-Rahman, M.S. Feature-Selection-Based DDoS Attack Detection Using AI Algorithms. *Telecom* **2024**, *5*, 333–346. https://doi.org/10.3390/telecom5020017

Academic Editor: Philip Branch

Received: 12 February 2024
Revised: 31 March 2024
Accepted: 12 April 2024
Published: 17 April 2024

1. Introduction

Over the past two decades, web-based software and services have experienced a sharp rise in popularity. Currently, 57 percent of people on the planet utilize the Internet [1]. As a result, worries about internet security have greatly increased. On the Internet, several security dangers have frequently existed. Among other things, common internet outliers include Trojans, worms, port scans, and denial-of-service assaults [2]. Traditional network architectures have difficulty providing efficient solutions for big and complicated networks. SDN (software-defined networking) is an alternative strategy in which network traffic is managed with software rather than hardware such as switches and routers. A centralized controller, operating as the network's core decision maker, takes over the control plane in SDN. SDN switches typically handle the data plane and execute controller commands. This change in control architecture improves network management flexibility and manageability [3,4]. SDN provides answers to a variety of network difficulties. These include dynamic remote setup vendor-independent device selection cost savings through centralized control, low-cost network devices with simpler data operations, increased Quality of Service (QoS), and improved link failure detection [5]. Despite its advantages, SDN contains security weaknesses at several architectural layers, including unique risks. Attackers acquiring control of the central controller, allowing them to interrupt the network via distributed denial of service (DDoS) assaults, is a big problem. These assaults, which

frequently employ "botnets" of infected PCs, are difficult to identify and prevent. Their frequency and intensity are rising, offering a substantial challenge to service providers and administrators in terms of prompt identification and mitigation [6]. This study presents a decision-tree-based ensemble learning approach for detecting DDoS assaults in SDN-based Supervisory Control and Data Acquisition (SCADA) systems. DDoS assault traffic data are achieved using a particular simulated experimental network architecture. The decision tree ensemble models' performance is optimized using feature selection and hyper parameter tweaking techniques [7]. For feature selection, lots of different methods are available but we need to choose the best one to handle the detection problem. This study analyzed the performance of 15 filter-based, wrapper-based, and embedded FS approaches, as well as an ensemble feature selection (EnFS) strategy. Additionally, both supervised and unsupervised learning approaches are used to assess the quality of each feature subset and identify the best-performing one. Our experiment shows that the EnFS technique beats individual FS and offers a universally optimal feature set for AI models [8]. Traditional techniques cannot be used with SDN due to the design differences between the two networking paradigms. This encourages the authors to develop the SDN traffic dataset and work on the SDN testbed. The effort to identify DDoS attacks in SDN has already been done. However, the identification of the critical features that are crucial for attack detection is carried out in this paper. NGBoosT Classifier, Random Forest, CNN, Decision Tree, and SGD are some of the machine learning models that we employed here to detect DDoS attacks based on feature selection. SYN Flood and NetBIOS attacks are the two main types of attacks that are deployed.

Contribution

This paper's contribution is as follows:

- Detecting DDoS assaults using the 16 attributes of the public dataset;
- For efficient categorization, it employs feature selection techniques;
- Different AI algorithms were applied to classify attacks in DDoS systems;
- NGBooST Classifier discovered for tabular data.

The sections of this study are organized as follows: Section 2 provides an overview of relevant studies. Section 3 explains the whole method of data preparation and the dataset. Section 4 describes the models which we used and also the details of the environment. Section 5 summarizes this article's results and sums up and suggests future research directions.

2. Related Works and Contribution

In order to discriminate between legitimate traffic and DDoS attack traffic, a previous paper investigated various machine learning techniques for DDoS attack detection and suggests leveraging innovative features. Among the machine learning methods used to identify DDoS attacks on SDN are Logistic Regression, Support Vector Classifier, K-Nearest Neighbor, Random Forest, Ensemble Classifier, and Artificial Neural Network. Table 1 provides a list of the features currently in use and the ML models that have been deployed.

Table 1. Existing algorithms to protect SDN against DDoS attacks using AI Models.

Paper	AI Techniques	Dataset	Features
[9]	NB, SVM, and NN	Real-time dataset from TCP traffic	Number of hosts connected per second
[10]	DT	Self-generated traffic	Protocol and service type, flag, TTL, and source/destination IP
[11]	KNN, DT, and NN	CAIDA 2007 and self-generated traffic	Number of ports per IP, the entropy of ports per IP, and number of ICAMP packets per IP
[12]	SVM, KNN, and RF	NSL-KDD and self-generated	Extracted features 27 and 40

Table 1. *Cont.*

Paper	AI Techniques	Dataset	Features
[13]	DA, SVM, KNN, NB, and DT	ISCX data	Number of bytes sent by source and destination number of packet sent by source, flow duration, and number of bytes divided by number of packets sent by source and destination
[14]	SVM, KNN, NN, and NB	Self-generated traffic	12 features including the number of packets received on the control plane
[15]	Apache spark	DDOS_DNS_AMPL, DDOS_CHARGEN AND RADB_DDOS	Source/destination IP, source/destination port number, protocol, packet length, number of bytes, and timestamp
[16]	DPMM	Dataset generated by other	Number of packets transmitted, ratio of source and destination bytes, and connection duration time
[17]	KNN, NB, and SVM	Self-generated traffic	Flow length, flow duration, flow size, and flow rate
[18]	NB	NSL-KDD	25 features in total including protocol and duration
[19]	Soft-max, NN and, Stacked Auto encoder	Self-generated traffic	34 from TCP, 20 UDP, and 14 features from ICMP flows
[20]	SVM	KDD1999, KDD CUP 1999	30 features including protocol and flag
[21]	RF, SVM, XGBoost, DT, and k-NN	CIC-IDS2018	26 features selected
[22]	DNN, LSTM, and GRU	CICDDoS2019 and DDoS-AT-2022	No features selected
[23]	CNN	KDD cup 99	Features extracted automatically (not mentioned)
[24]	LSTM-CNN and RNN	Self-generated	No features selection
[25]	Cybernet	CICDDOS2019	Not Given
Our paper	RF, DT, CNN, SGD, and NGBooST Classifier	CICDDOS2019 (SYN and NeTBIOS)	16 features selected including source port and destination port

2.1. Performance of ML/DL in DDoS Attack Detection

In their dataset, Meti et al. [9] only include TCP traffic from actual networks and include the number of connected devices per second and peak/off-peaktime indicators. In terms of accuracy, precision, and recall, the results of the comparison demonstrate that NN has the best accuracy and precision. A decision tree (DT) technique is also suggested by Zekri et al. [10] to be used to identify DDoS attacks in the cloud network. They divide traffic into four groups and use self-generated traffic to validate the suggested approach. Entropy and logarithm values are calculated by Tuan et al. [11] to identify TCP-SYN flood and ICMP flood assaults in SDN, respectively. In order to identify DDoS attacks, Sahoo et al. [12] offer an enhanced SVM model that uses kernel principal component analysis (KPCA) and genetic algorithms (GA). In order to determine the additional costs associated with using ML for DDoS attack detection in SDN, Bakker et al. [13] compared the initialization times and accuracy of seven classifiers. The effectiveness of four ML approaches in the detection of DDoS attacks with and without feature selection is also compared by Polat et al. [14]. The most accurate method is KNN employing wrapper-based selection; the authors trained with 6 and 12 key features.

2.2. ML Deployment for DDoS Attack Detection

For the purpose of creating and optimizing models, Huyu et al. [15] suggest sending real-time traffic data to an off-line learning pipeline. Data are gathered via routers and transmitted to the pipeline for feature engineering and data transformation. They can be used in conjunction with an existing model to defend the network from DDoS assaults. For the purpose of detecting DDoS attacks using DNS queries, Ahmed et al. [16] provide an SDN-based Dirichlet Process Mixture Model (DPMM) clustering technique. This implies that many projections will be wrong. According to Dong et al. [17], a flow is assumed to be a vector with values for the length, duration, size, and rate of the flow. Four internet service providers were connected to one another in a set-up scenario by Mohammed et al. [18]. They take into account the possibility of an ML server using NB classification

operating outside of this network. To analyze network traffic, Niyaz et al. [19] suggest running three modules—Traffic Collector and Flow Installer (TCFI), Feature Extractor (FE), and Traffic Classifier (TC)—over the controller. SVM is used by Wang et al. [20] in conjunction with the sFlow toolset to detect threats to SDN. The model pulls behavioral variables, such as protocol type, for SVM training from traffic statistics it gathers from switches via the controller. SVM can identify an attack by contrasting normal and malicious behavioral profiles. This research provides a feature-engineering- and machine-learning-based technique for detecting DDoS assaults in SDN. First, the CSE-CIC-IDS2018 dataset was cleaned and normalized, and the best feature subset was identified using an enhanced binary grey wolf optimization approach [21]. This research proposes DL-2P-DDoSADF, a deep-learning-based two-phase DoS attack detection system. The suggested approach has been validated using the CICDDoS2019 and DDoS-AT-2022 datasets. The performance and efficacy of several deep learning approaches, including Deep Neural Networks (DNN), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU), are compared [22]. A two-phase technique for DDoS attack detection and mitigation is introduced by the framework. In the detection step, a Deep Convolutional Neural Network (CNN) is used for classification and an Improved-Update-oriented Rider Optimization Algorithm (IU-ROA) for feature selection. A bait detection approach is used in the mitigation step to neutralize malicious nodes. The KDD Cup 99 dataset is used for testing, and the results demonstrate efficient DDoS attack detection and mitigation [23]. The goal of this paper is to use a variety of deep learning algorithms to categorize the traffic into normal and harmful classes based on features extracted from the dataset [24]. The authors of this study [25] have created a novel deep learning architecture called the Cybernet model, concentrating on 1D CNN and LSTM architectures. Learning the essential behaviors of cyberattacks in the realm of cybersecurity and accurately identifying and detecting various forms of DDoS attacks were the objectives.

2.3. DDoS

DDoS attacks are often launched from a single computer or resource in order to limit or completely disable access by overloading the targeted system or resource. The amount of damage caused by a DDoS assault depends on the attacker's resource strength. A DDoS assault is a sort of cyberattack that attempts to overload a target system by flooding it with traffic from many computers or devices. DDoS assaults are often carried out utilizing zombie computers or botnets. The attackers launch a DDoS assault by sending coordinated traffic through zombie machines to the target systems. These assaults can make targeted systems unreachable by overloading resources on a massive scale [7].

3. Methodology

The flowchart shown in Figure 1 depicts how machine learning may be used to identify attacks using DDoS. The first step is to choose two datasets, CICDDOS 2019 and KDD CUP-1999, which are then analyzed using exploratory data analysis (EDA) in order to find trends, connections, and important features. Correlation analysis, feature engineering, and data visualization are all part of the EDA phase and help to improve the quality of the data used for training models and comprehending the dataset. Following EDA, the characteristics that were processed are preserved and several machine learning techniques are used. If the model's predictions are suitable it is determined with an accuracy check. If not, the method reflects the standard cycle of continual improvement in machine learning model building, progressively suggesting changes to the features and algorithms until the required accuracy is obtained.

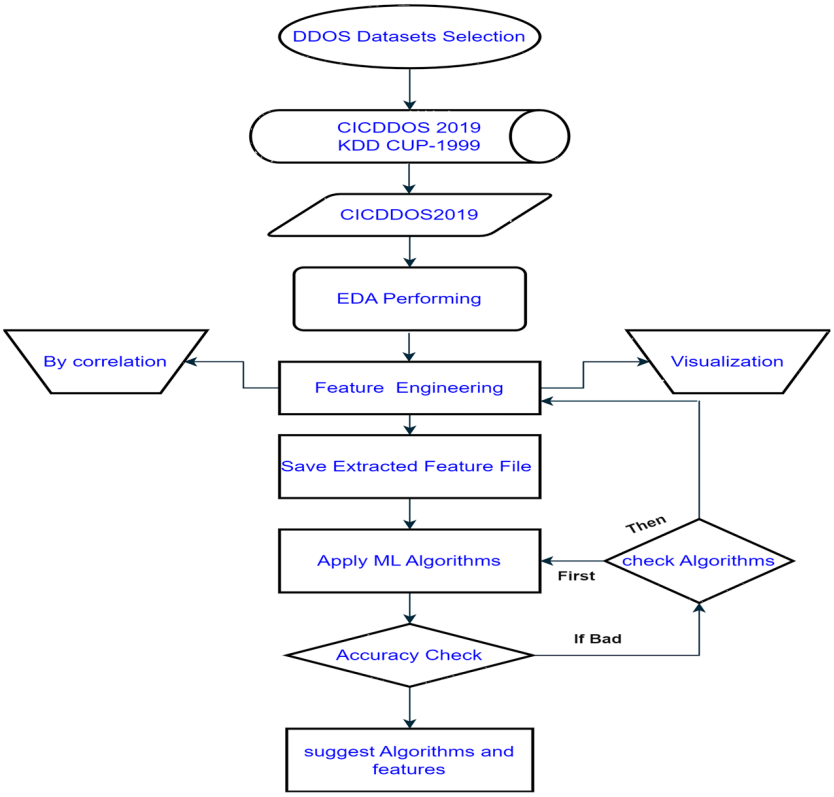

Figure 1. Architecture of the proposed model.

3.1. DDoS Dataset

Every AI model requires effective datasets for training and testing in order to calculate the percentage of authenticity. Accurate data collection is a major problem worldwide; thus, in this area, researchers and institutions are making a contribution. We use CICDDOS2019 [26] datasets, as well as having separate training and testing files, which have 88 columns and millions of rows. They consist of 12 different DDOS attack types in total (NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDP-Lag, WebDDoS, SYN, and TFTP), as well as 7 attack types for testing (NetBIOS, LDAP, MSSQL, UDP, UDPLag, and SYN). Basically, SYN and NetBIOS are the two types chosen for this experiment. The size of these two types of data are five Giga bytes (5 GB). The features that are considered are shown in Table 2.

Table 2. Extracted features.

Features	Descriptions
Source Port	Number on sender's side of a communication
Destination Port	Number on the receiver's side
Flow Duration,	Sequence of packets between source and destination
Total Fwd Packets	Total number of packets sent
Total Backward Packets	From destination to source
Bwd Packet Length Min	The minimum length of packets
Flow IAT Min	Minimum time between two consecutive packets in a flow
Bwd IAT Min	Interarrival time of packets in the backward
Fwd PSH Flags	The sender has finished sending data
Fwd Header Length	Header in the forward direction
Bwd Header Length	Header in the backward direction
Bwd Packets/s	Backward packets per second
Init_Win_bytes_backward	The size of the receiving window during the initial phase of the connection in the backward direction

Table 2. *Cont.*

Features	Descriptions
Active Mean	Duration of active network connections
SYN Flag Count	SYN flags in the TCP packets
Inbound	Distinguishing between normal and potentially malicious traffic

3.1.1. NeTBIOS

NetBIOS DDoS attacks are reflection-based. NetBIOS is a TCP/IP protocol that runs at Layer 5 of the OSI/ISO model. It connects system applications on multiple computers to interact in a local network. NetBIOS offers three services: Name Service (NS), Datagram Distribution Service (DGM), and Session Service (SSN). By default, NetBIOS-NS operates on port 137. DDoS assaults occur exclusively with the NetBIOS naming service [27].

3.1.2. SYN

SYN Flood (attack) is a major part of the evolving DDoS landscape. This attack exploits the widely employed TCP protocol and especially the 3-way handshake, flooding targeted end-hosts, i.e., victims, with SYN packets. These exhaust their memory and processing resources, failing to serve legitimate requests. SYN Flood attacks are difficult to counter via commonly used IP-based mitigation schemas. IP-based rules, required to block the attack traffic, increase proportionally to the number of malicious sources. This demands network devices/firewalls to store thousands/millions of filtering rules, which is unattainable due to memory resource limitations. Notably, when spoofing is employed, IP-based filtering may be ineffective. An alternative mitigation method for SYN Floods relies on the SYN cookies technique. This approach, instead of blocking malicious SYN packets, generates appropriately crafted SYN-ACK packets. Although this method protects the victim from the launched attack, it consumes significant processing resources and introduces large rates of backscatter traffic [28].

3.2. Data Preprocessing

Data preparation is critical for machine learning performance. It converts raw data into useable formats, allowing for insights and forecasts. Addressing missing data is critical in the CICDDoS2019 dataset. Error detection and correction are critical, and deleting columns with large data gaps might improve model performance. The CICDDoS2019 dataset contains many properties (columns) with zero values, as we have discovered. Such as some columns (Unnamed: 0, Bwd PSH Flags, Fwd URG Flags, Bwd URG Flags, FIN Flag Count, PSH Flag Count, ECE Flag Count, Fwd Avg Bytes/Bulk, Fwd Avg Packets/Bulk, Fwd Avg Bulk Rate, Bwd Avg Bytes/Bulk, Bwd Avg Packets/Bulk, and Bwd Avg Bulk Rate) which were dropped because these columns contain zero values in most of the records.

Feature Engineering: After choosing a dataset, the key question is how to organize the data according to machine learning models which we selected before. Having fewer features may lead to better data visualization, faster learning, more accuracy, and less overfitting, among other benefits. There are several options for feature selection, and the best ones are given below. We used filtering techniques that relied on correlation for the end outcome.

Filter Methods: Based on statistical metrics (such as correlation and chi-squared), these strategies rank or score features independently of the machine learning algorithm.

Correlations: The statistical relationship between two variables is measured through correlations. Negative correlation means one variable rises as the other declines, and positive correlation says both variables rise or fall together. Correlation coefficients like Pearson's (linear) or Spearman's (rank-based) are used to measure it. Strong connection is indicated by a coefficient close to +1 or −1; weak or no correlation is indicated by a value close to 0. Correlation highlights associations rather than implying causation. To analyze

connections and forecast behavior, it is utilized in a variety of disciplines like economics, science, and data analysis.

Almost three different approaches were used in our study to pick features. First, we utilized the scikit-learn library's "SelectBest" method. The purpose of this method is to automatically choose features according to defined standards, such as having to pick 10 features from the dataset. But the results fell short of our expectations. Next, we looked at univariate selection strategies, which are useful in locating characteristics that make a substantial contribution to the predictive ability of the model. Finally, we chose features that depend on one another using a correlation-based method, which greatly improved our model's accuracy. By quantifying the linear correlations between the variables, this strategy helped to clarify the links between the variables. However, it is crucial to remember that this method relies on linear interactions and may miss nonlinear relationships, scale inconsistencies, and connections between variables, all of which call for careful analysis. In Figure 2 the heat map depicts the correlations between several network traffic factors. Each square in this type of depiction represents the relationship between the x- and y-axis features. The color intensity and sign of the numbers indicate the correlation's strength and direction (red for positive and black for negative). For example, darker red squares indicate a stronger positive link, whereas darker black squares indicate a stronger negative correlation. Lighter squares or white indicate little to no connectivity. This graphic is widely used to quickly assess which pairs of attributes are most strongly associated. In our study, we nearly applied five algorithms across the entire dataset (88 columns).

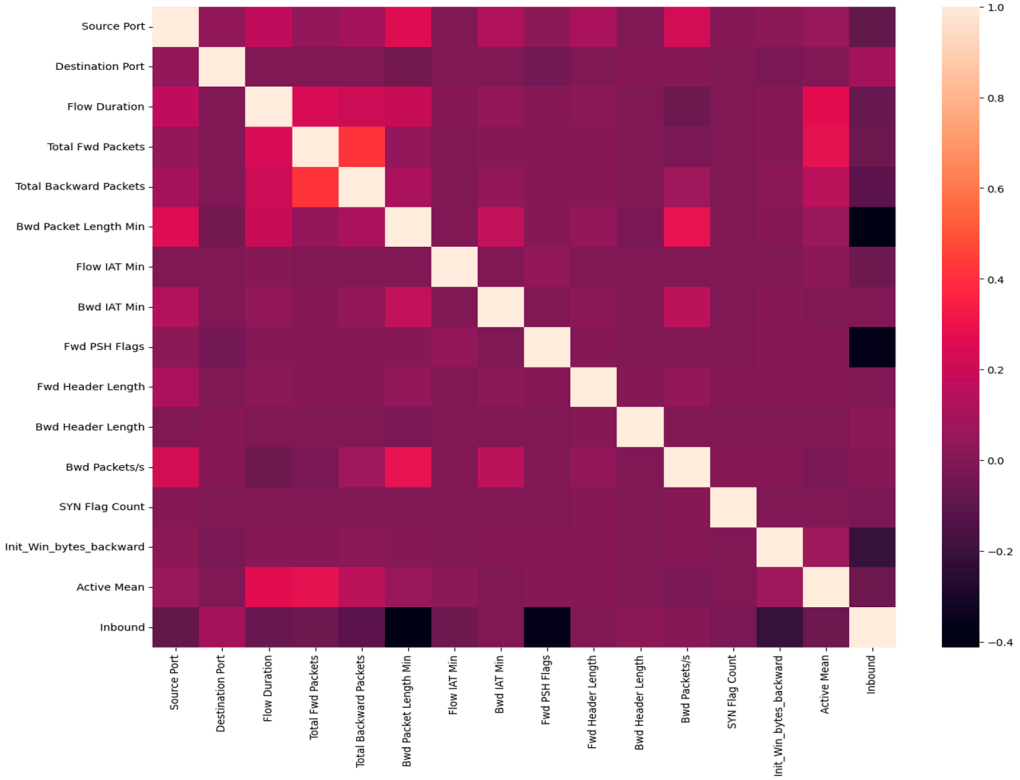

Figure 2. Correlations of Attributes.

4. DDoS Detection Model

In our paper, five machine learning models—Random Forest, Decision Tree, CNN, Stochastic gradient descent, and NGBoosT Classifier—are examined, put into practice, and tested.

4.1. Random Forest

A Random Forest is a method of ensemble learning that combines forecasts from various decision trees. It is used for both classification and regression. The predictions from all of the decision trees in a Random Forest are combined to form the final forecast. Each decision tree in a Random Forest is trained using a different subset of the data. Through the use of ensemble techniques, RF increases the accuracy of individual DTs, making it more dependable for DDoS assault detection. The necessity to train many trees makes RFs more computationally demanding than a single DT [21].

4.2. Decision Tree

A decision tree is a graphical representation of a decision-making process that iteratively separates data into subgroups based on the values of input attributes. Each split produces a branching node that may lead to other splits or other outcomes. In classification and regression tasks, decision trees are used to generate predictions or choices based on patterns in the feature values of the data. Since DTs are simple to understand and analyze, figuring out the criteria that lead to DDoS attack detection is an easy task. Consistency in detection can be impacted by minor modifications in the data that lead to noticeably different tree architectures [10].

4.3. Convolutional Neural Network

A convolutional neural network is a form of neural network that is commonly used for image identification and classification. CNNs have a variety of applications, including object detection, image processing, computer vision, and face recognition. Images are used to provide input to convolutional neural networks. Convolutional neural networks, as opposed to manually developing features, are used to automatically learn a hierarchy of features that may subsequently be used for categorization. Algorithms are able to adjust to the changing nature of network traffic in SDN, making them appropriate for environments where attack patterns might emerge. They often require an enormous amount of labeled data to achieve excellent performance, which might not always be available in the case of detecting DDoS attacks [25].

4.4. NGBoosT Classifier

An NGBoost Classifier can be used as a classifier for classification tasks. In this case, NGBoosT gathers a collection of weak learners (typically decision trees) and then combines their predictions to get a final prediction. As it is designed to optimize the negative log-likelihood loss, it is well suited for probabilistic predictions in classification. Utilizing the natural gradient can lead to more dependable and effective convergence, which is the main advantage of NGBoosT over traditional gradient boosting. When detecting DDoS attacks in unclear situations, NGBoost's probabilistic predictions—which include both the predictions and their uncertainty—are helpful. Some teams may find it difficult to grasp the underlying concepts of probabilistic prediction and gradient boosting [29].

4.5. Stochastic Gradient Descent

Stochastic Gradient Descent is a straightforward yet highly efficient method for fitting linear classifiers and regressors to convex loss functions such as (linear) SVMs and Logistic Regression. SGD has been used effectively for large-scale and sparse machine learning issues that are frequently encountered in text categorization and natural language processing. It may be applied with a variety of linear models and is flexible enough to handle different DDoS detection scenarios. Data pretreatment needs to be done properly because feature scaling is so important to SGD's performance [29].

Experimental Environment

In our experimental setup, we utilized a robust computing environment to conduct our research effectively. The host system ran on Windows 10, equipped with 8 GB of RAM and

powered by an Intel Core i7 processor clocked at 2.50 GHz. This high-performance hardware configuration provided the necessary computational resources for our experiments.

For software tools, we leveraged Visual Studio Code (VSCode) as our integrated development environment (IDE) for coding and experimentation. Additionally, we employed python version 3.7.10 as our primary programming language. To ensure a well-organized and isolated development environment, we created a dedicated anaconda environment. Within this environment, we installed the Tensor Flow library, a powerful deep learning framework, which played a central role in our experiments.

5. Experiment Result

This section presents the findings of numerous tests to evaluate the precision of various machine learning models. When assessing the effectiveness of the installed DDoS detection system, a number of parameters are taken into consideration which are described below. We used two different types of DDOS attack data (SYN and NeTBIOS). Furthermore, we divided the data into the most famous percentages for training and testing, which was 0.8 into training and 0.2 into testing. Overall, the combined size of data are 4.85 GB.

5.1. Performance Parameters

We use a variety of metrics, including F1, recall, accuracy, precision, and a confusion matrix, to determine how well AI models perform. Accuracy (shown in Figure 3, precision, recall, and F1 scores are shown in Figure 4 and results are in Table 3.

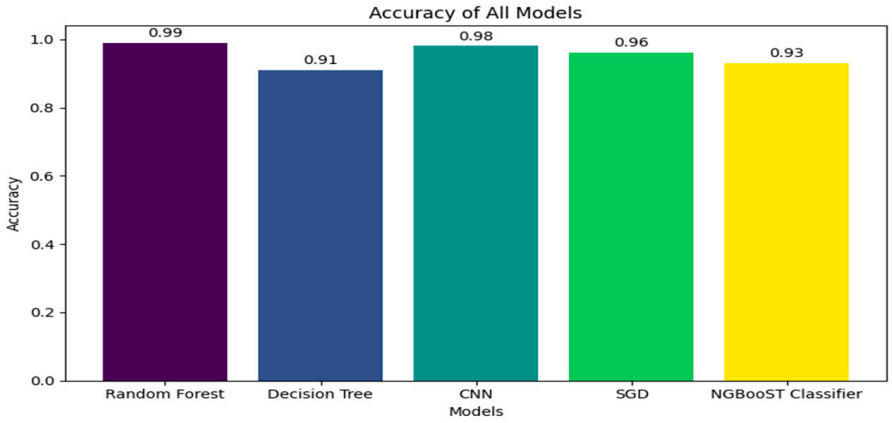

Figure 3. Accuracy of all models.

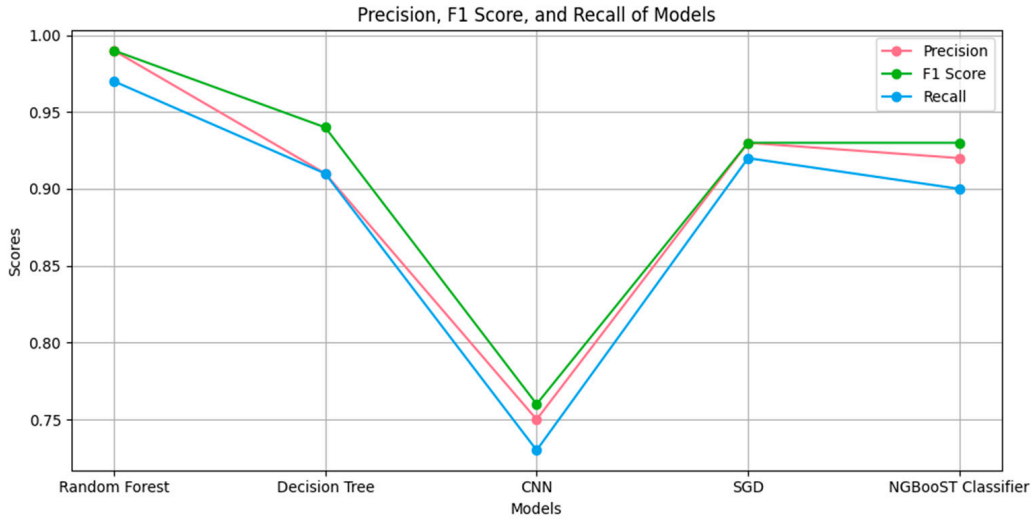

Figure 4. Recall, precision, and F1_Score.

Table 3. Performance parameter results.

Model Name	Precision	F1 Score	Recall
Random Forest	0.99	0.99	0.97
Decision Tree	0.91	0.94	0.91
CNN	0.75	0.76	0.73
SGD	0.93	0.93	0.92
NGBooST Classifier	0.92	0.93	0.9

Accuracy: One of the performance indicators is accuracy, which is mathematically defined as a fraction where the denominator specifies the sum of false positives and negatives as well as the terms present in the numerator, and the numerator specifies the sum of true positives and true negatives. Equation (4) defines it as follows:

$$\text{Accuracy} = ((TP + TN)/(FP + FN)) * 100, \quad (1)$$

where TP = True Positive, TN = True Negative, FP = False Positive, and FN = False Negative.

Precision: Precision is the total number of examples that were correctly identified as being in a certain class. Precision is defined as the ratio of correctly predicted cases to all expected cases. The formula below can be used to compute it.

$$\text{Precision} = (TP/(TP + FP)) * 100, \quad (2)$$

where TP = True Positive and FP = False Positive.

Recall: Recall is the ability of the classifier to correctly recognize each positive case. Recall is defined as the proportion of true positives to the total of both true positives and false negatives.

$$\text{Recall} = (TP/(TP + FN)) * 100, \quad (3)$$

where TP = True Positive and FN = False Negative.

F1 Score: The F1 score may be regarded as a harmonic mean of accuracy and recall, with the greatest value being 1 and the worst value being 0. Precision (P) and recall (R) both contribute equally to the F1 score.

$$\text{F1 score} = 2 * (P * R)/(P + R), \quad (4)$$

where P = precision and R = recall.

Confusion Matrix: A confusion matrix is a table that lists the effectiveness of a classification model. It provides details about the model's accuracy and flaws by showing the number of true positives, true negatives, false positives, and false negatives. The confusion matrix can be used to calculate many metrics, including accuracy, precision, recall, and F1 score, which aid in assessing the efficacy of the model.

5.2. Evaluation of ML

CNNs can adjust to the dynamic nature of SDN network traffic, they are appropriate for situations where attack patterns are subject to change. As DTs are simple to read and comprehend, figuring out the criteria that lead to DDoS attack detection is a basic process. By employing ensemble techniques, RF increases the accuracy of individual DTs and increases its dependability for DDoS attack detection. SGD is flexible for a range of DDoS detection scenarios since it can be applied to multiple linear models. When detecting DDoS attacks in unclear situations, NGBoost's probabilistic predictions—which include both the predictions and their uncertainty—are helpful.

5.2.1. Accuracy

Figure 3 shows the accuracy scores that are used to measure the effectiveness of different machine learning models. The Random Forest algorithm has an impressive

0.99 accuracy at the top of the chart, shown by a purple bar, which shows how strong its prediction powers are. The CNN models, which have an accuracy of 0.98 and are represented by a teal bar, are strong competitors in image and pattern recognition tasks. At 0.96 accuracy, the Stochastic Gradient Descent (SGD) classifier, indicated by a green bar, likewise exhibits excellent performance. The NGBoost Classifier, indicated by a yellow bar, has a slightly lower score of 0.93 but still exhibits an excellent degree of accuracy. With a blue bar at 0.91, the decision tree algorithm has the lowest accuracy rate in this grouping, but it still shows a good accuracy rate.

5.2.2. Precision, Recall, and F1 Score

Figure 4 shows the F1 score and recall for a collection of five machine learning models—Random Forest, Decision Tree, CNN, SGD, and NGBoost Classifier—are shown side by side in this line graph. These metrics are essential for assessing each model's accuracy in data classification. The three metrics for the Random Forest and NGBoost Classifier models are in the range of 0.95 to 1.00, which indicates a high degree of accuracy with a minimal number of false positives or negatives. The CNN and Decision Tree models, on the other hand, exhibit a sharp decline in all measures, with a maximum at 0.85, suggesting a greater percentage of incorrect classifications. While recall and F1 score are comparable to the best models, the SGD model's precision is slightly below 0.90, indicating that it is still fairly good at recognizing true positives despite its reduced accuracy. Furthermore, we have also included data in Table 3 for ease of understanding.

5.2.3. Confusion Matrix

The performance of classification models, detailing the correct and incorrect predictions for three classes, is depicted in the confusion matrix. In Figure 5, diagonal values represent correct predictions with 18,856 for Class 0, 3,056,171 for Class 1, and 2,849,811 for Class 2, indicating a high number of true positives, especially for Classes 1 and 2. The off-diagonal numbers show misclassifications, where, notably, Class 0 was often confused with Class 2 (6720 instances), and Class 2 with Class 1 (194,197 instances). The heat levels correlate with the frequency of predictions, with the darker shades representing higher frequencies of predictions, and thus, the matrix visually emphasizes where the model predictions are concentrated.

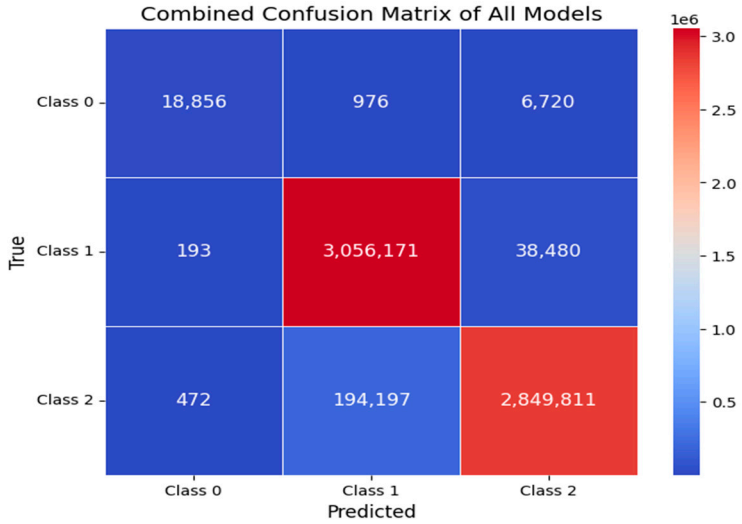

Figure 5. Confusion matrix of all models.

5.3. Comparision with Already Used Models and Our Models' Results

Table 4 shows an overview of the studies conducted in the last few years and analyzes the accuracy of the tested models. Our major goals were choosing features and applying two distinct models on one-dimensional data. We obtained accuracy values of 0.96 and 0.93 in

both of the tested models on 1D data; on SYN and NeTBIOS DDOS data, none of the authors of previous studies used convolution neural networks and natural gradient boosting.

Table 4. Comparison with already used models.

Papers	Models	Result (Accuracy)
[9]	NB, SVM, and NN	0.7, 0.8, and 0.8
[11]	KNN, DT, and NN	0.98, 0.98, and 0.98
[12]	SVM, KNN, and RF	0.95,0.92, and 0.94
[14]	SVM, KNN, ANN, and NB	Average of 0.95
[21]	RF, SVM, XGBoost, DT, and kNN	0.99, 0.98, 0.99, 0.9
[22]	DNN, LSTM, and GRU	0.97, 0.96, and 0.96
[29]	Cybernet	0.99
Our work	RF, DT, SGD, CNN, and NGBooST	0.99, 0.91, 0.98, 0.96, and 0.93

6. Conclusions

The DDoS attack, which is a very strong technique that has been launched to attack network devices and services, has recently been regarded as one of the most significant attacks. As a result, in this research, we examine the DDoS assault, analyze it, and construct a machine learning model to detect such attacks. In this research, we explored multiple feature selection approaches to identify the most important features that can be used to anticipate DDoS attacks in an effective manner. Sixteen features were chosen from the dataset and were used with five machine learning models. According to the results, the RF, CNN, and SGD models with 16 features provide the best precision, accuracy, recall, and F1 score.

In the future, we will build our own dataset of distributed denial of service assaults in a variety of contexts, such as software-defined networks and the Internet of Things, using these 16 attributes. Consequently, by utilizing the CICDDoS2019 dataset in our research, we were able to greatly improve the way in which DDoS attacks were classified based on these attributes.

Author Contributions: Conceptualization, M.S.R. and M.N.A.S.; methodology, M.N.A.S. and M.S.R.; software, M.N.A.S. and M.S.R.; validation, M.N.A.S. and M.S.R.; formal analysis, M.N.A.S. and M.S.R.; investigation, M.S.R.; resources, M.S.R.; data curation, M.S.R.; writing—original draft preparation, M.N.A.S. and M.S.R.; writing—review and editing, M.S.A.-R.; visualization, M.N.A.S. and M.S.R.; supervision, I.-S.H. and M.S.A.-R.; project administration, I.-S.H. and M.S.A.-R.; funding acquisition, I.-S.H. and M.S.A.-R. All authors have read and agreed to the published version of the manuscript.

Funding: This project was supported by NSTC 112-2221-E-155-009, Taiwan.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: We wish to acknowledge the anonymous referees who gave precious suggestions to improve the work.

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

References

1. Internet Growth Usage Statistics. 2019. Available online: https://www.clickz.com/internetgrowthusage-stats-2019-time-online-devices-users/235102/ (accessed on 10 January 2024).
2. Singh, J.; Behal, S. Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges and future directions. *Comput. Sci. Rev.* **2020**, *37*, 100279. [CrossRef]
3. Hossain, M.A.; Sheikh, M.N.A.; Rahman, S.S.; Biswas, S.; Arman, M.A.I. Enhancing and measuring the performance in software defined networking. *Int. J. Comput. Netw. Commun. (IJCNC)* **2018**, *10*, 27–39. [CrossRef]
4. Sheikh, M.N.A.; Hwang, I.S.; Ganesan, E.; Kharga, R. Performance Assessment for different SDN-Based Controllers. In Proceedings of the 2021 30th Wireless and Optical Communications Conference (WOCC), Taipei, Taiwan, 7–8 October 2021; pp. 24–25. [CrossRef]

5. Ahuja, N.; Singal, G.; Mukhopadhyay, D.; Kumar, N. Automated DDOS attack detection in software defined networking. *J. Netw. Comput. Appl.* **2021**, *187*, 103108. [CrossRef]
6. Wang, Y.; Wang, X.; Ariffin, M.M.; Abolfathi, M.; Alqhatani, A.; Almutairi, L. Attack detection analysis in software-defined networks using various machine learning method. *Comput. Electr. Eng.* **2023**, *108*, 108655. [CrossRef]
7. Oyucu, S.; Polat, O.; Türkoğlu, M.; Polat, H.; Aksöz, A.; Ağdaş, M.T. Ensemble learning framework for DDoS detection in SDN-based SCADA systems. *Sensors* **2024**, *24*, 155. [CrossRef] [PubMed]
8. Saha, S.; Priyoti, A.T.; Sharma, A.; Haque, A. Towards an Optimized Ensemble Feature Selection for DDoS Detection Using Both Supervised and Unsupervised Method. *Sensors* **2022**, *22*, 9144. [CrossRef] [PubMed]
9. Meti, N.; Narayan, D.G.; Baligar, V.P. Detection of distributed denial of service attacks using machine learning algorithms in software defined networks. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Manipal, India, 13–16 September 2017; pp. 1366–1371. [CrossRef]
10. Zekri, M.; El Kafhali, S.; Aboutabit, N.; Saadi, Y. DDoS attack detection using machine learning techniques in cloud computing environments. In Proceedings of the 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech), Rabat, Morocco, 24–26 October 2017; pp. 1–7. [CrossRef]
11. Tuan, N.N.; Hung, P.H.; Nghia, N.D.; Tho, N.V.; Phan, T.V.; Thanh, N.H. A DDoS attack mitigation scheme in ISP networks using machine learning based on SDN. *Electronics* **2020**, *9*, 413. [CrossRef]
12. Sahoo, K.S.; Tripathy, B.K.; Naik, K.; Ramasubbareddy, S.; Balusamy, B.; Khari, M.; Burgos, D. An evolutionary SVM model for DDOS attack detection in software defined networks. *IEEE Access* **2020**, *8*, 132502–132513. [CrossRef]
13. Bakker, J.N.; Ng, B.; Seah, W.K. Can machine learning techniques be effectively used in real networks against DDoS attacks? In Proceedings of the 2018 27th International Conference on Computer Communication and Networks (ICCCN), Hangzhou, China, 11 October 2018; pp. 1–6. [CrossRef]
14. Polat, H.; Polat, O.; Cetin, A. Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models. *Sustainability* **2020**, *12*, 1035. [CrossRef]
15. Huyn, J. A scalable real-time framework for DDoS traffic monitoring and characterization. In Proceedings of the Fourth IEEE/ACM International Conference on Big Data Computing, Applications and Technologies, Austin, TX, USA, 5–8 December 2017; pp. 265–266. [CrossRef]
16. Ahmed, M.E.; Kim, H.; Park, M. Mitigating DNS query-based DDoS attacks with machine learning on software-defined networking. In Proceedings of the MILCOM 2017–2017 IEEE Military Communications Conference (MILCOM), Baltimore, MD, USA, 23–25 October 2017; pp. 11–16. [CrossRef]
17. Dong, S.; Sarem, M. DDoS attack detection method based on improved KNN with the degree of DDoS attack in software-defined networks. *IEEE Access* **2019**, *8*, 5039–5048. [CrossRef]
18. Mohammed, S.S.; Hussain, R.; Senko, O.; Bimaganbetov, B.; Lee, J.; Hussain, F.; Kerrache, C.A.; Barka, E.; Bhuiyan, M.Z.A. A new machine learning-based collaborative DDoS mitigation mechanism in software-defined network. In Proceedings of the 2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Limassol, Cyprus, 15–17 October 2018; pp. 1–8. [CrossRef]
19. Niyaz, Q.; Sun, W.; Javaid, A.Y. A deep learning based DDoS detection system in software-defined networking (SDN). *EAI Endorsed Trans. Secur. Saf.* **2016**, *4*, e2. [CrossRef]
20. Wang, P.; Chao, K.M.; Lin, H.C.; Lin, W.H.; Lo, C.C. An efficient flow control approach for SDN-based network threat detection and migration using support vector machine. In Proceedings of the 2016 IEEE 13th International Conference on E-Business Engineering (ICEBE), Macau, China, 4–6 November 2016; pp. 56–63. [CrossRef]
21. Liu, Z.; Wang, Y.; Feng, F.; Liu, Y.; Li, Z.; Shan, Y. A DDoS detection method based on feature engineering and machine learning in software-defined networks. *Sensors* **2023**, *23*, 6176. [CrossRef] [PubMed]
22. Mittal, M.; Kumar, K.; Behal, S. DL-2P-DDoSADF: Deep learning-based two-phase DDoS attack detection framework. *J. Inf. Secur. Appl.* **2023**, *78*, 103609. [CrossRef]
23. Singh, S.; Jayakumar, S.K.V. DDoS Attack Detection in SDN: Optimized Deep Convolutional Neural Network with Optimal Feature Set. *Wirel. Pers. Commun.* **2022**, *125*, 2781–2797. [CrossRef]
24. Ahuja, N.; Singal, G.; Mukhopadhyay, D. DLSDN: Deep Learning for DDOS attack detection in Software Defined Networking. In Proceedings of the 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 28–29 January 2021; pp. 683–688. [CrossRef]
25. Salih, A.A.; Abdulrazaq, M.B. Cybernet Model: A New Deep Learning Model for Cyber DDoS Attacks Detection and Recognition. *Comput. Mater. Contin.* **2024**, *78*, 1275–1295. [CrossRef]
26. Sharafaldin, I.; Lashkari, A.H.; Hakak, S.; Ghorbani, A.A. Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In Proceedings of the 2019 International Carnahan Conference on Security Technology (ICCST), Chennai, India, 1–3 October 2019; pp. 1–8. [CrossRef]
27. Mekala, S.; Dasari, K.B. NetBIOS DDoS attacks detection with machine learning classification algorithms. In Proceedings of the 2023 International Conference on Advancement in Computation & Computer Technologies (InCACCT), Gharuan, India, 5–6 May 2023; pp. 176–179. [CrossRef]

28. Dimolianis, M.; Pavlidis, A.; Maglaris, V. SYN flood attack detection and mitigation using machine learning traffic classification and programmable data plane filtering. In Proceedings of the 2021 24th Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN), Paris, France, 1–4 March 2021; pp. 126–133. [CrossRef]
29. Duan, T.; Anand, A.; Ding, D.Y.; Thai, K.K.; Basu, S.; Ng, A.; Schuler, A. Ngboost: Natural gradient boosting for probabilistic prediction. In Proceedings of the International Conference on Machine Learning, Virtual, 13–18 July 2020; pp. 2690–2700. [CrossRef]