*Journal of Discrete Mathematical Sciences & Cryptography*
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 26 (2023), No. 7, pp. 2025–2044
DOI : 10.47974/JDMSC-1848

# Improvement in DDoS attack detection in software defined network using ML algorithm

Saumitra Chattopadhyay *
Ashok Kumar Sahoo †
Sanjay Jasola §
*Department of Computer Science & Engineering*
*Graphic Era Hill University*
*Dehradun*
*Uttarakhand*
*India*

**Abstract**

The complexity of handling modern apps is causing older network architectures to struggle. As a result, Software Defined Networking (SDN) has developed into a paradigm-shifting network framework that enables administrators to exercise programmable control over whole network infrastructures. This development makes it easier to configure and operate large networks, but it also adds vulnerabilities, including the possibility of Distributed Denial of Service (DDoS) assaults that could take down the SDN controller. Integrating security tools like firewalls, intrusion detection and prevention systems, and anti-DDoS solutions is crucial to reducing this danger. In the context of SDN, leveraging the analytical power of Machine Learning (ML) algorithms to analyse network traffic patterns is a key tactic for fending off DDoS attacks. Effective defence systems are supported by these ML models, which can distinguish between normal network behaviour and DDoS assault traffic with ease. Decision trees (DT), support vector machines (SVM), and artificial neural networks (ANN) are examples of supervised ML techniques that have evolved into crucial tools in this field. These algorithms are developed utilising datasets with annotations that include examples of both legitimate and malicious network traffic. Once trained, the models may be used to instantly categorise incoming traffic, making it easier to identify and contain any DDoS attacks in real-time.

***Subject Classification:*** *68M25.*

***Keywords:*** *DDoS, Software defined network, Ensemble, Bagging, Boosting.*

* *E-mail:* **saumitroc@gmail.com (Corresponding author)**
† *E-mail:* **ashoksahoo2000@yahoo.com**
§ *E-mail:* **sjasola@yahoo.com**

© Taru Publications

## 1. Introduction

The traditional network architectures are facing many challenges in managing modern day applications. Recently, a new network framework known as Software Defined Networking (SDN) has given the administrators the ability to programmatically manage the entire network infrastructure. As a result it is possible to configure and manage large networks efficiently [1]. The primary concept behind this technique revolves around the separation of the data plane and the control plane[2]. One of the possible risks associated with SDN is Distributed Denial of Service (DDoS) attacks [3], which can render the controller unreachable. In order to stop such incidents, it is necessary to implement security measures such as firewalls, intrusion detection and prevention systems and anti-DDoS solutions[4]. During a DDoS attack, the attackers employ a strategy of flooding a targeted host or a cluster of hosts with a large influx of packets, often utilizing falsified source addresses [5]. While SDN architecture offers advantages like centralized controllers, global visibility, and flexible traffic forwarding rules, it also encounters various concerns and challenges, especially regarding security [6].The potential for sophisticated DDoS attacks increases when there is centralized control and communication between the controller and switches, as it may create vulnerabilities [7]. To mitigate these risks, it becomes vital to implement suitable measures to safeguard the SDN architecture, including DDoS mitigation techniques, encryption protocols, and robust authentication mechanisms.

The application of machine learning (ML) algorithms has shown noteworthy potential in the field of identifying DDoS attacks within SDN. In this context, supervised ML algorithms like decision trees (DT), Support Vector Machines (SVM), and Artificial Neural Networks (ANN) are commonly utilized. Once trained, the models can be deployed to categorize incoming traffic in real-time as either normal or malicious, based on identified patterns. This empowers the timely identification and mitigation of DDoS attacks.

In a scenario of DDoS detection in SDN, ensemble methods [8] that leverage multiple ML algorithms can be effectively utilized. These methods combine the strengths of diverse algorithms[9], improving the accuracy and resilience of DDoS detection systems. By combining various techniques, an ensemble model can achieve a wide-ranging and precise identification of DDoS attacks[10].

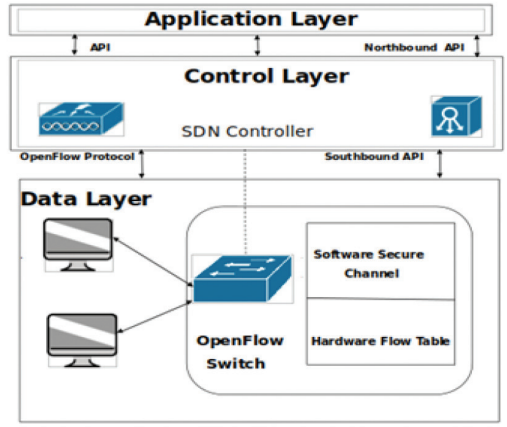



**Figure 1**

**The SDN Architecture**

However, despite its promise, SDN technology its promise, SDN (shown in Figure 1), has its own security issues[11] that need to be addressed. Each of these layers is vulnerable to ddos attacks and has their own vulnerabilities [12].

## Literature Review

For the past few years, many researchers have studied the problem of DDoS attacks in SDN and used many innovative methods for the early detection of the attack using ML approaches. The following works are reviewed stating their methods used, ML algorithm used, and results obtained.

In a study conducted by Tonkal et al. [1] an ML approach for DDoS attack detection in SDN was proposed. The procedure outlined in the paper involves two steps: first, using a feature selection algorithm, specifically K-means clustering, to identify the most important features from the traffic data, and second, using an ML algorithm, specifically k-Nearest Neighbors (kNN), to categorize the traffic as either normal or attack traffic. The paper assessed the approach of using a publicly available dataset containing DDoS attack traffic. The results demonstrated a high accuracy rate of 98.85% in detecting DDoS attacks. Besides introducing a fresh ML technique that relies on K-means clustering and kNN, the paper also addresses the constraints of the suggested approach and offers recommendations for prospective avenues of research.

In a study conducted by Dong et al. [2], researchers analysed various features and traffic behaviours to propose a potential resolution for identifying DDoS attacks in SDN networks. Two distinct DDoS detection algorithms were introduced: one utilizing the Degree of Attack (DDADA) and the other employing machine learning (DDAML). To evaluate the efficacy of these algorithms in detecting DDoS attacks specifically aimed at the SDN controller, four features were employed. The findings demonstrated that, when compared to the other algorithms, the DDADA and DDAML algorithms both produced higher TPR values, with DDADA achieving 0.987 and DDAML achieving 0.994.

The researchers in [3] used a custom dataset which was created specifically for their study. They combined SVM and Random Forest (RF) classifiers to achieve an amazing accuracy rate of 97.5%. The dataset was created by using 'hping3' to launch real-time assaults against a specially designed SDN topology that was put into use on the mininet emulator, under the control of a remote controller run by 'Ryu'. Ryu is one of the SDN controllers created specifically for managing increased traffic rates and network agility. Mininet is a tool used for SDN emulation. It is used to visualize the switches, controllers and application of SDN in a virtualized environment

In the research conducted by Rahman et al [4], several machine learning techniques were assessed to detect and prevent DDoS attacks within an SDN network. A decision tree-based classifier (J48), Random Forest, SVM, and kNN were among the techniques that were tested. The researchers used mininet and a 'Ryu' controller to build the SDN network, while 'weka' was used to train and test their machine learning models. The evaluation process includes developing a number of models before selecting the best one to apply to the suggested network. Once chosen, this model was incorporated into a script created to identify and stop DDoS assaults with the intention of preventing and mitigating them. The results of the evaluation revealed that J48 outperformed the other algorithms that were assessed, particularly in terms of the time required for training and testing. These findings suggest that J48 is a superior choice in terms of efficiency and effectiveness when it comes to detecting DDoS attacks within the SDN network, taking into account the specific parameters examined in the study conducted by the researchers. The researchers demonstrated in this work that J48 outperforms the other assessed algorithms, particularly in terms of training and testing time.

In another research conducted by authors in the research as described in [5], the researchers established an SDN environment using the mininet

and floodlight simulation platforms. The researchers gathered six different characteristic values and a model for detecting DDoS attacks was developed by utilizing the switch flow table and implementing SVM classification algorithms. In traditional network architectures, there exist two main methods for detecting DDoS attacks: one based on analysing traffic characteristics and the other based on identifying traffic anomalies. By comparing and evaluating the attributes of ongoing network data packets with a pre-existing database of characteristics, it becomes possible to determine the occurrence of a DDoS attack. The experiments conducted in the study demonstrated that their proposed method achieved an average accuracy rate of 95.24% with a small amount of flow collection. This indicates that their approach successfully and accurately identified DDoS attacks within the SDN environment, highlighting the effectiveness of their methodology.

In a research conducted by authors as described in [6], a machine learning model was presented for detecting malicious traffic in an SDN environment, using the Decision Tree and SVM algorithms. The experimental results showcased the superior accuracy and detection rates of both algorithms. They trained and tested their model using the KDD99 [14] dataset and deployed the classification module within the SDN environment. In the simulation environment used for the experiment, it was shown that the SVM method performed better than the Decision Tree technique. In order to improve DDoS attack detection by looking at kernel vectors and attaining even higher detection accuracy, the authors also talked about the prospective installation of Advanced Support Vector Machine (ASVM) [15].Researchers demonstrated in this paper that, in comparison to other ML methods, the Decision Tree and SVM algorithms offers greater accuracy and detection rates

The authors in the research paper [7] proposed a combined approach that integrates ML and deep neural network classifiers with the centralized features of SDN to effectively mitigate the damaging impact of DDoS attacks on network systems. By integrating this classifier with their SDN controller, fraudulent flows from both incoming and outgoing traffic were effectively removed. Two different assault scenarios were tested by the researchers, one coming from within the network system and the other from the outside. They conducted their research using the CSE-CIC-IDS2018 dataset [16], and machine learning classifiers such the Linear Support Vector Machine (LSVM), DT, Random Forest, and Naive Bayes methods were used. They used IDS-DDoS software, which intercepted and gathered packets to produce traffic flows, to speed up the classification

process. This software was in charge of categorizing the traffic flows as malicious or genuine and informing the SDN controller when malicious traffic was present. An efficient method to recognize and reduce DDoS attacks was provided by the integration of machine learning and deep neural network classifiers with the centralized control of SDN. Researchers seek to improve the security and resilience of the network system against such hostile operations by integrating these components. Linear SVM and DNN beat Decision Tree and Random Forest in practice although in simulation it is opposite. They concluded by saying that more research is required to bridge the gap between theory and practise.

In the research as described in [8], a comprehensive mechanism consisting of four modules to address DDoS attacks: a trigger for detecting attacks, the detection of attacks, tracing back the source of attacks, and implementing measures to mitigate the impact of attacks was proposed. The development of the attack detection trigger module, which aims to promptly respond to DDoS attacks and lessen the burden on controllers and switches, is one important contribution of their work. With the use of this trigger system, attacks can be quickly detected and efficiently mitigated. The researchers used a neural network-based approach to detect DDoS attacks, relying on neural networks for precise and effective attack detection. The researchers unveiled a strategy for fending off DDoS assaults that includes countermeasures and flow table cleaning. By using this approach, attacks can be stopped at the source, and occupied switch resources can then be released. The suggested methodology helps to reduce DDoS assaults and ensures the effective use of network resources by efficiently blocking malicious traffic and clearing out flow tables. The researcher's experimental findings demonstrated that the proposed mechanism may accurately pinpoint the assault source and immediately begin attack detection in less than one second. It can also stop an attack at the source and free up resources that switches have been using.

In another work carried out by Abou El Houda, et al [9] proposed a highly effective and scalable remedy known as WisdomSDN has been developed to address DNS amplification attacks within a SDN. WisdomSDN uses the proactive and stateful mapping strategy to defend SDN controllers from Domain Name Service amplification attacks [17]. Illegal DNS requests and responses are addressed from both a detection and mitigation perspective in the proposed approach. By actively sending only valid responses and preventing the amplification of fraudulent DNS responses, the Proactive Amplification Suppression system makes sure that every DNS request and response have a direct and exclusive

correlation. With this proactive strategy, the SDN infrastructure is protected against malicious traffic. A machine learning-based module is used in WisdomSDN to quickly detect illegitimate DNS requests in real-time. This module consists of three parts: Flow Statistical Characteristics, which efficiently gathers flow features using the sFlow protocol in a scalable manner; Entropy-based Characterization Scheme, which determines the amount of disorder or randomness in network traffic; and Bayesian Network Fusion, which automatically detects unauthorized DNS requests based on the information gathered. WisdomSDN offers a powerful method for preventing DNS amplification assaults in SDN systems by integrating the proactive and stateful mapping strategy with the machine learning-based DDoS detection module. By precisely identifying and reducing unauthorized DNS traffic, the suggested method improves the security and robustness of SDN networks.

Aladaileh, Mohammad A., et al [10] provides a comprehensive review of state-of-the-art techniques for detecting DDoS attacks on SDN controllers. It investigates a number of topics, such as the analysis of detection methods, the investigation of various DDoS attack types that target SDN controllers, and a qualitative comparison with similar studies. The need of using many packet header features rather than depending on a single feature is one significant finding from the survey that is highlighted. This method improves DDoS attack detection and identification, particularly in conditions with low network flow rates. The detection technique becomes more reliable and efficient in precisely recognizing DDoS assaults on SDN controllers by taking into account several aspects. The research provided a fresh categorization strategy for DDoS attack detection techniques already in use. The classification is based on the features and deployed locations of various techniques inside the SDN environment, as well as the utilized technique. The diverse detection methodologies are presented in a systematic and ordered manner by this classification framework, allowing for easier comprehension and comparison. The experimental results presented in this paper demonstrated that WisdomSDN can quickly detect and mitigate DNS amplification attacks with a high detection rate, a low FPR, and little overhead, making it a promising option for SDN environment.

In a work as described in [11], the researchers examined and categorized DDoS flooding attacks on SDN through the utilization of diverse ML techniques. Numerous well-known ML algorithms and classifiers, including Gaussian Nave Bayes (GNB), Regression Tree (CART), Classification, and Quadratic Discriminant Analysis (QDA), are

examined in the paper. The researchers use Mininet to simulate the SDN model and Low Orbit Ion Cannon (LOIC) to create DDoS flooding attacks. The CART algorithm, comes out with the best all-around performance. It achieves robustness, a quick training time of 12.4 ms, a fast prediction speed of $5.3 \times 10^5$ observations per second, and a high prediction accuracy of 98%.

In a research as explained in [12], the authors suggest a method to categorize SDN traffic using ML algorithms incorporating neighborhood component analysis (NCA). This is done using a publicly accessible dataset called "DDoS attack SDN Dataset."[3] This dataset includes 23 different features from both normal traffic and attack-related traffic. UDP, TCP, and Internet Control Message Protocol (ICMP) are just a few of the several traffic kinds that are covered by the dataset. With the exception of source and target machine information, it consists of over 100,000 recordings with statistical data like byte_count, duration_sec, packet rate, and packet per flow. The authors use a number of machine learning methods, including DT, kNN, ANN, and SVM, on the collected dataset after preprocessing and feature selection phases. The accuracy rates of the algorithms are used to assess the categorization results. The accuracy rate for the ANN approach is 97.35%, followed by kNN's 95.41%, DT's 94.14%, and SVM's 80.56%. The paper presents evidence demonstrating the effectiveness of using Network Traffic Classification (NCA) and machine learning methods to classify Software-Defined Networking (SDN) traffic as either benign or malicious. The ANN technique outperforms the other studied algorithms in terms of accuracy, demonstrating its applicability for identifying and differentiating between legitimate and malicious data in SDN systems.

In a study conducted by Khalid Mohamed Hosny et al.[13] proposed a detection mechanism for DDoS attacks is put forward, comprising of three primary phases. Various attack rates are used in the initial step to calculate the mean rate of flow start, which is subsequently used to define a threshold. To establish if the flow rate is within the anticipated range, suggesting typical traffic, or whether it signals abnormal activity, the estimated average flow initiation rate is compared to the predetermined threshold in the following step. The flow specification is examined in the third step using several attack scenarios to ascertain whether the incoming communication is legitimate or malicious, confirming or denying the existence of an attack. The outcomes show the usefulness of the suggested mechanism and its capacity to identify DDoS attacks, whether they are directed at a single victim or several victims. The system achieves an average detection accuracy of 99.26% and a minimum assault detection

rate of 20%. With the quickest detection time thus far, this accuracy rate is the greatest one.

In the research carried out by P. Maity et al. [14], a method was proposed based on the central limit theorem is implemented. This method, which primarily focuses on spotting resource depletion assaults, takes a probabilistic approach. The DARPA dataset is used to train the probabilistic model for this assignment. In terms of false negative rate (FNR), the suggested probabilistic strategy outperforms the entropy-based method in terms of identifying various assault scenarios. The proposed method's FNR is lowered to 98% at a 50% attack intensity, whereas the entropy-based mechanism displays a FNR of 78%. Finally the researchers proved and highlighted the advantages of probabilistic techniques over entropy-based methods

In a research described in [15], they suggested a controller scheduling mechanism for SDN to identify DDoS attack precisely. The method includes selecting the appropriate request for the controller while considering other factors, such as the normalized waiting time, the size of the queue, and the seriousness of the intended switch.While the polling-based multi-queue scheduling approach outperforms the single queue scheduling method in terms of performance, it also causes imbalance between different queues when subjected to Distributed Denial of Service (DDoS) assaults. In contrast, the suggested approach offers an effective way for SDN controllers to lessen the effects of DDoS attacks.

## 1. Proposed Work

The proposed work is depicted in the Figure 3 below. Firstly, the training data will be provided to the inference engine (Figure 2). After successful training of the proposed system, the testing data will be provided to the system. If the prediction is to accept, the incoming data packet will be only forwarded to the SDN controller/ OpenFlow switch for further processing in the SDN [31].

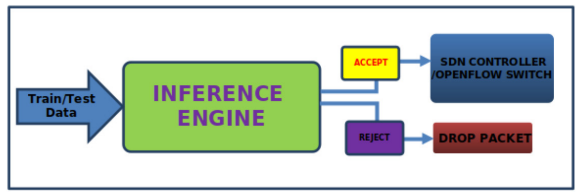



**Figure 2**

**The Proposed Work**

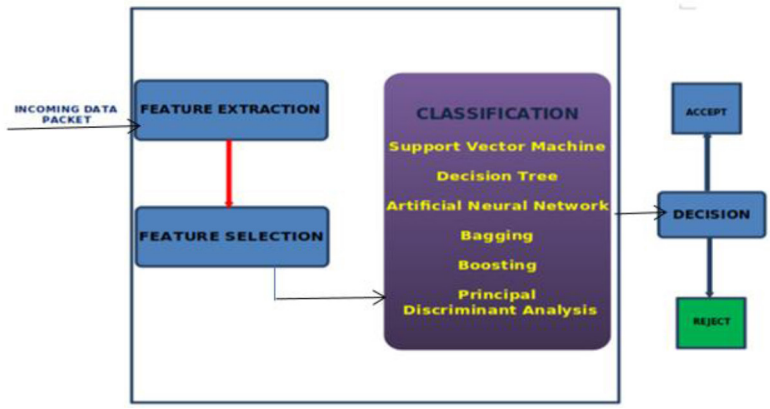



**Figure 3**

**The Proposed System Architecture**

The proposed model of the work is depicted clearly in Figure 3 above. We will assess and create the Dataset which will be relevant for SDN environment only [28]. This data set will be divided into two groups training and testing data sets. The data set will be fed to the proposed Inference Engine which will classify the packet entering as normal or attack type based on the different machine learning algorithms [22].

In the first phase of this Inference Engine will do feature extraction to take only the most relevant features from the data set. In the second phase, we will perform the feature selection to qualify the best parameters for classification of the incoming packets as normal or attack type [23].

In the next phase, we will use the features selected using machine learning algorithms[24] like SVM, Decision Tree, Artificial Neural Network, bagging, boosting and Principal Component Analysis for training our model. Once trained, we will use the model for testing.

The SDN Dataset on Mendeley Data by Nisha Ahuja is a comprehensive collection of network traffic data captured during a simulated DDoS attack. The dataset contains over 25,000 network flow records captured from a SDN environment [3]. The dataset was recorded when a botnet was employed to simulate a DDoS attack and overwhelm the network with traffic. The dataset includes information on the transport protocol, source and destination IP addresses[25], number of packets and bytes transmitted and received, and other relevant network traffic information.

Attacks that are protocol-based, such as those against TCP, UDP, or ICMP, target a specific protocol[26], whereas volumetric attacks involve

saturating a network with traffic. Application-based attacks, on the other hand, concentrate on specific applications, such as HTTP or DNS servers[27]. This SDN Dataset is a useful tool for researchers and security professionals because it includes information from all three types of DDoS attacks, which will help them understand and mitigate the effects of these attacks. Additionally, the information contains flow duration and inter-arrival time, which are crucial characteristics for identifying DDoS attacks.

The dataset's richness and variety make it a useful tool for building DDoS detection algorithms and conducting machine learning-based analysis.

The various features present in the dataset are explained below:

- APPF: The average packet count per flow is a measurement that is used to compare the volume of packets inside a flow to the total number of flow entries in a switch.

  The formula is as written in equation 1 below:

$$APPF = \frac{\Sigma_{i=1}^{i=n} c_i}{f} \quad ...(1)$$

The packet count in this case, denoted by ci, ranges from flow 1 to flow n. The letter f stands for the total number of flow entries in the switch.

DDoS assaults can be discovered by keeping an eye on the APPF value and comparing it to a specified threshold. The presence of an attack can be determined by a large drop in the APPF below the predicted range since the attacker's packets are dispersed across several flow entries with lower packet counts.

ABPF: The Average Byte Count per Flow (ABPF), which compares the total number of bytes transferred during a flow to the total number of flow entries present in a switch, is another statistic used to identify DDoS attacks. The formula is as written in equation 2 below:

$$ABPF = \sum_{i=1}^{i=n} b_i / f \quad ...(2)$$

Here, 'i' range from flow 1 to flow n, and bi denotes the total bytes throughout each flow. The letter n stands for the total number of flow entries in the switch.

DDoS assaults can be discovered by tracking the ABPF value and contrasting it with a predetermined threshold. An attack is present when the ABPF significantly drops below the expected range since the attacker's traffic often has lower bytes per flow than legitimate traffic.

The overall count of flow descriptions saved in a switch's flow table at any given time defines the total number of flows in the switch. Each flow description contains details about a particular flow, such as the source and destination IP addresses, source and destination ports, protocol type, and other pertinent attributes. The notation 'f' is used to denote the total number of flow entries in a switch, which can be determined by calculating the length of the flow table. The formula is as written in equation 3 below:

$$f = \text{length}(\text{flowtable}) \quad ...(3)$$

Gaining valuable insights into network activity and potential security risks can be achieved by keeping track of the overall count of flow entries in a switch.

Duration: A flow entry's duration in a switch's flow table is measured using a metric denoted as duration. It offers details on how long a specific flow has been present in the network. The formula for calculating the total duration is as stated in equation 4 below:

$$\text{Total duration} = ((d * 10^9) + (d_{nano})) \quad ...(4)$$

Shorter durations could represent regular network traffic, while longer durations might be a sign of an active attack. Recognizing the duration of flows can help distinguish between good and bad hosts or flows within the network and shed light on the temporal behavior of traffic.

The quantity of Packet_in messages reported to the controller can significantly increase when attackers use spoofed IP addresses or other malicious tactics. This increase is the result of the switch sending more packets to the controller that lack matching flow entries. It is customary to maintain and modify a counter within the event handler routine connected to the occurrence of the Packet_in message in order to keep track of the number of Packet_in messages generated. The counter is increased in accordance with each Packet_in message that the controller receives and processes.

Packet rate: The term "packet rate" refers to a calculation that expresses how many packets are transmitted in a second. This metric provides insightful data on how quickly data travels through the network. For determining the packet rate, we can use the formula as stated in equation 5 below:

$$\text{Packet Rate} = P_{t+1} - \frac{P_t}{1} \quad ...(5)$$

In the equation above, "$p_{t+1}$" stands for the number of packets sent within a flow at time "t+1," while "$p_t$" is the quantity of packets sent at time "t." Furthermore, "i" stands for the monitoring interval.

We compute the difference between the subsequent packet counts "$p_{t+1} - p_t$" and divide that result by the monitoring interval "I" to determine the packet rate.

**Table 1**
**Result from Bagged Tree**

| | | | |
|---|---|---|---|
| True Class | DDoS | 10196 | 0 |
| | Non DDoS | 1 | 15889 |
| | | DDoS | Non DDoS |
| Predicted Class | | | |

Port Bandwidth: The total amount of data sent and received by a switch port over a specific time period is measured by the port bandwidth. It gives information about how quickly data is moving through the port.

To calculate the port bandwidth, we consider the transmitted bytes (tx_bytes) and received bytes (rx_bytes) collected periodically as port statistics from the switch. The formula for calculating the bandwidth is as follows:

$$\text{Bandwidth} = (t * 8) / 1000 + (r * 8) / 1000 \quad ...(6)$$

In this formula, t represents the transmitted bytes and r represents the received bytes. The values are multiplied by 8 to convert them from bytes to bits. Then, they are divided by 1000 to convert the result from bits to kilobits. In the given equation, the variable "tx_bytes" represents the count of bytes sent from the port, "rx_bytes" denotes the count of bytes received at the port, and "interval" indicates the duration during which the statistics are gathered (usually measured in seconds).

**Feature Extraction and Selection**

Here we have explored the important characteristics found in the dataset and how they are utilized by a machine learning algorithm to achieve accurate classification of the traffic. We have used 16 features from the dataset. In our work firstly we have achieved high accuracy in 16 features using ensemble machine learning model and secondly we have even reduced the number of features by removing some irrelevant features

with regard to software defined networking topology. This approach aims to use ensemble bagging machine learning for DDoS attack detection. This machine learning method for intrusion detection[28] is efficient and particularly helpful for exposing multiple attacks.

The Ensemble Classifier[29] is a machine learning technique that combines multiple models to improve the precision and robustness of the classification process. By combining several weak models, this technique aims to produce a strong model. The Ensemble Classifier creates a final output by combining the predictions of several base models that have been trained on the same dataset. The predictions can be combined in a variety of ways, including voting, averaging, and stacking. Each base model predicts a class label in the voting method, and the final result is determined by the outcome that receives the most votes. The final output of the averaging method is created by averaging the outputs of all the base models.

The approach entails the following steps:

- Generating multiple subsets from the original dataset by randomly selecting samples with replacement.
- Constructing a base model for each subset.
- Executing all the models simultaneously.
- Combining the predictions from all models to derive the final predictions.

## 2. Results and Discussions

The dataset shown in Table 2 was used to evaluate the model's performance, and metrics like Accuracy, Precision, Recall, and F1-score were computed. The Ensemble Bagged Trees machine learning model's accuracy was compared to that of the other models shown in Table 1 for comparison. Data exploration was done before the analysis to better understand the dataset, including the distribution of legitimate and malicious traffic[30] and the number of instances in each traffic class.

A machine learning method known as an ensemble classifier (EC) combines different classifiers to reach a conclusion. KNN, Decision Tree, Random Forest, and SVC are a few potential classifiers. On this dataset, it has been demonstrated that this method yields exceptional results, outperforming other machine learning models with an accuracy of 99.99%.

**Table 2**

**Results obtained from Data Set**

| Performance Metrics | Percentage |
|---|---|
| Accuracy | 99.98% |
| Precision | 99.95% |
| Recall | 99.99% |
| F1-Score | 99.95% |
| Specifity | 99.95% |

We used 16 features (Figure 4), which were also used in a Ahuja et al.[3], to assess the performance of the ensemble bagging machine learning model. On these 16 features, our bagging trees model produced exceptional results, and on 13 features, it produced noticeably better results. The results for the 16 features are shown in the table below, along with an evaluation of the results against those of other models in (Table 3).

The effectiveness of different classification models was evaluated within a particular scenario. Logistic Regression[31] achieved a classification accuracy of 70.03%, indicating that it has shortcomings in accurately capturing the linear relationship between the features and generating precise predictions for the class labels. This implies that LR may struggle to effectively model complex patterns and dependencies within the data, leading to suboptimal performance in certain scenarios. Likewise, the K-nearest neighbors (KNN) algorithm[32] exhibited a classification accuracy of only 70.01%. Despite its simplicity and reliance on proximity calculations, KNN[33] may not yield satisfactory outcomes in intricate situations that demand more sophisticated approaches.

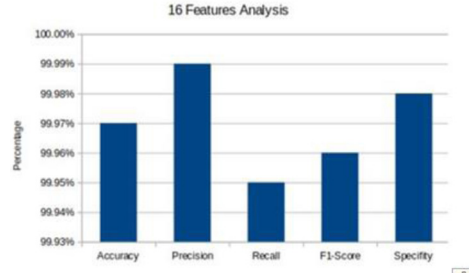



**Figure 4**

**Result Analysis**

SVM[34], which aims to find an optimal decision boundary, also fell short in delivering satisfactory results. The highly correlated features and potential non-linearity of the data could have hindered SVM's ability to find an appropriate decision boundary without overfitting.

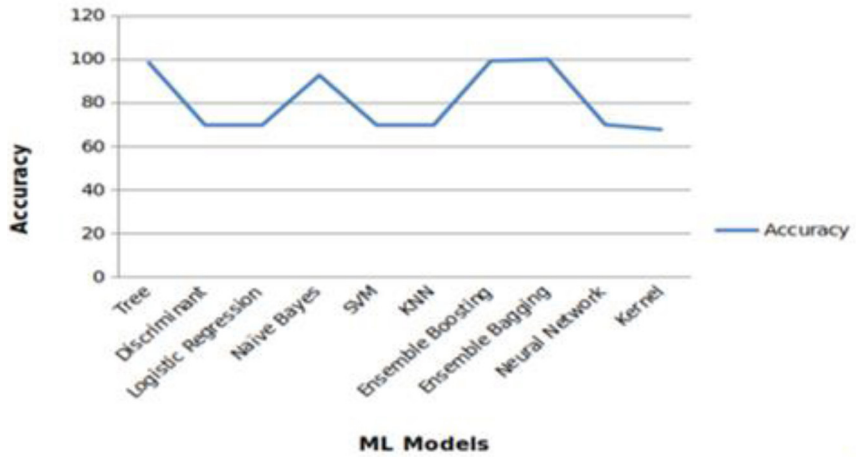



**Figure 5**

**Comparisons of results obtained from experiments.**

**Table 3**

**Comparative Analysis of Results Obtained**

| Model | Preset | Accuracy |
|---|---|---|
| Tree | Fine Tree | 98.99 % |
| Discriminant | Quadratic Discriminant | 69.90 % |
| Logistic Regression | Logistic Regression | 70.03 % |
| Naive Bayes | Kernel Naive Bayes | 92.70 % |
| SVM | Quadratic / Cubic / Fine Gaussian / Medium Gaussian/ Coarse Gaussian | 70.03 % |
| KNN | Fine KNN/ Weighted KNN | 70.01 % |
| Ensemble | Boosted Trees | 99.33 % |
| Ensemble | Bagged Trees | 99.99 % |
| Neural Network | Narrow / Medium / Wide / Bilayered / Trilayered Neural Network | 70.03 |
| Kernel | SVM Kernel | 67.76 |

Random Forest (RF)[35], known for its ensemble of decision trees, yielded an accuracy of only 92.02%. RF's strength lies in compensating for individual decision trees' mistakes[36], but in this case, it did not perform as well as desired (shown in Figure 5).

To assess the efficiency of the suggested approach, a comparison was carried out between the proposed method and existing works in the area of DDoS attack detection. An emulated dataset was utilized for this purpose. It was discovered that the highest performance achieved by prior research in this field was 98.8%. However, the results obtained from our proposed model surpassed this benchmark, achieving the highest accuracy rate of 99.9%. This significant improvement in accuracy demonstrates the superiority of our proposed method in effectively detecting DDoS attacks. The obtained results highlight the potential of our model as a reliable and efficient solution for DDoS attack detection.

## 5. Conclusion and future work

In this paper we have selected a SDN relevant dataset and used various ML algorithms to solve the problem of DDoS detection in SDN. We have found that ensemble methods gives better results compared to individual algorithms. In future we are going to test this model on other datasets and also going to test it on our own datasets .

## Conflicts of Interest

The authors declare that they have no competing interests.

## References

[1] Tonkal, Ö., Polat, H., Başaran, E., Cömert, Z., & Kocaoğlu, R., Machine learning approach equipped with neighbourhood component analysis for DDoS attack detection in software-defined networking. Electronics, 10(11), 1227 (2021).

[2] Dong, Shuo, and Mohsen Sarem. "DDoS Attack Detection Method Based on Improved KNN With the Degree of DDoS Attack in Software-Defined Networks." IEEE Access, vol. 8, pp. 5039-5048 (2020). doi:10.1109/ACCESS.2019.2963077.

[3] Ahuja, Nisha, et al. "Automated DDOS Attack Detection in Software Defined Networking." *Journal of Network and Computer Applications*, vol. 187, p. 103108 (2021).

[4] Rahman, Osman, Muhammad Adnan Ghafoor Quraishi, and Chi-Hung Lung. "DDoS Attacks Detection and Mitigation in SDN Using Machine Learning." 2019 IEEE World Congress on Services (SERVICES), pp. 184-189 (2019). doi:10.1109/SERVICES.2019.00051.

[5] Abou El Houda, Zakarya, Leila Khoukhi, and Abdellah Senhaji Hafid. "Bringing Intelligence to Software Defined Networks: Mitigating DDoS Attacks." IEEE Transactions on Network and Service Management, vol. 17, no. 4, pp. 2523-2535 (2020). doi:10.1109/TNSM.2020.3014870.

[6] Cui, Yunhe, et al. "SD-Anti-DDoS: Fast and Efficient DDoS Defense in Software-Defined Networks." *Journal of Network and Computer Applications*, vol. 68, pp. 65-79 (2016). doi:10.1016/j.jnca.2016.04.005.

[7] Ye, Jin, et al. "A Ddos Attack Detection Method Based on SVM in Software Defined Network." Security and Communication Networks, vol. 2018 (2018), pp. 1-8. doi:10.1155/2018/9804061.

[8] Sudar, K. Muthamil, et al. "Detection of Distributed Denial of Service Attacks in SDN using Machine learning techniques." 2021 International Conference on Computer Communication and Informatics (ICCCI), pp. 1-5 (2021).

[9] Luong, Thanh-Khoa, Tien-Dung Tran, and Gia-Tuan Le. "DDoS attack detection and defense in SDN based on machine learning." 2020 7th NAFOSTED Conference on Information and Computer Science (NICS), pp. 31-35 (2020). doi:10.1109/NICS51282.2020.9335867.

[10] Aladaileh, Mohammad A., et al. "Detection Techniques of Distributed Denial of Service Attacks on Software-Defined Networking Controller–A Review." IEEE Access, vol. 8, pp. 143985-143995 (2020). doi:10.1109/ACCESS.2020.3013998.

[11] Sangodoyin, A.O., Akinsolu, M.O., Pillai, P., and Grout, V. "Detection and Classification of DDoS Flooding Attacks on Software-Defined Networks: A Case Study for the Application of Machine Learning." IEEE Access, vol. 9, pp. 122495-122508 (2021), doi: 10.1109/ACCESS.2021.3109490.

[12] Tonkal, Ö., Polat, H., Başaran, E., Cömert, Z., and Kocaoğlu, R. "Machine Learning Approach Equipped with Neighbourhood Component Analysis for DDoS Attack Detection in Software-Defined Networking." Electronics, vol. 10, p. 1227 (2021), doi: 10.3390/electronics10111227.

[13] Hosny, K.M., Gouda, A.E-S., and Mohamed, E.R. "New Detection Mechanism for Distributed Denial of Service Attacks in Software Defined Networks." *International Journal of Sociotechnology and Knowledge Development*, vol. 12, no. 2, pp. 1-30 (2020).

[14] Maity, P., Saxena, S., Srivastava, S., Sahoo, K.S., Pradhan, A.K., and Kumar, N. "An Effective Probabilistic Technique for DDoS Detection in OpenFlow Controller." *IEEE Systems Journal*, vol. 16, no. 1, pp. 1345-1354 (2022), doi: 10.1109/JSYST.2021.3110948.

[15] Li, S., Cui, Y., Ni, Y., and Yan, L. "An Effective SDN Controller Scheduling Method to Defence DDoS Attacks." *Chinese Journal of Electronics*, vol. 28, pp. 404-407 (2019), doi: 10.1049/cje.2019.01.017.

[16] Zhang, Y. and Liu, Z. "Software-Defined Networking: Overview, Applications, and Challenges." IEEE Communications Magazine, vol. 58, no. 1, pp. 78-84 (2020).

[17] Wang, H., Zhang, Y., and Li, H. "A Survey on Software-Defined Networking-Based Security Solutions." IEEE Communications Surveys and Tutorials, vol. 22, no. 1, pp. 246-266 (2020).

[18] Lee, J., Park, S., and Kim, H. "Secure and Efficient Software-Defined Networking Architecture for Virtualized Cloud Environments." IEEE Transactions on Network and Service Management, vol. 17, no. 1, pp. 288-301 (2020).

[19] Lin, J. and Xu, G. "SDN-Based DDoS Detection and Mitigation: A Survey." IEEE Communications Surveys and Tutorials, vol. 22, no. 1, pp. 361-380 (2020).

[20] Yu, M., Li, K., & Wu, Y., SDN-based security services for cloud computing: a survey. IEEE Transactions on Cloud Computing, 8(1), 1-15 (2020).

[21] Kim, J., Jeong, S., & Choi, Y., Security analysis of software-defined networking: a survey. IEEE Communications Surveys and Tutorials, 23(2), 1145-1177 (2021).

[22] Bhattacharjee, S., Bhatnagar, S., & Maitra, S., SDN-based architecture for secure and scalable IoT networks. *IEEE Internet of Things Journal*, 8(6), 4719-4733 (2021).

[23] Gao, J., Wang, J., & Ren, K., Software-defined networking security: a comprehensive survey. IEEE Communications Surveys and Tutorials, 23(1), 117-147 (2021).

[24] Kumar, A., Singh, K., & Khan, T., L-RTAM: Logarithm based reliable trust assessment model for WBSNs. *Journal of Discrete Mathematical Sciences and Cryptography*, 24(6), 1701-1716 (2021).

[25] Zhang, Y., Li, X., & Wu, J., A survey on the security challenges and solutions of software-defined networking. IEEE Network, 35(2), 28-33 (2021).

[26] Khan, T., & Singh, K., Resource management based secure trust model for WSN. *Journal of Discrete Mathematical Sciences and Cryptography*, 22(8), 1453-1462 (2019).

[27] Qiu, K., Wang, W., Zhu, X., Wu, M., & Li, H., A survey on software-defined networking security: threats, vulnerabilities, and countermeasures. IEEE Access, 8, 14723-14745 (2020).

[28] Sood, S., Sood, S., & Sood, J. K., A comprehensive review on security challenges and mitigation techniques in software-defined networks. *Journal of Network and Computer Applications*, 161, 102832 (2020).

[29] Khan, T., & Singh, K., Resource management based secure trust model for WSN. *Journal of Discrete Mathematical Sciences and Cryptography*, 22(8), 1453-1462 (2019).

[30] Chakraborty, M., Das, A. K., Samanta, A., Maity, S., & Goswami, D., Security challenges and mitigation techniques in software-defined networking. 2020 International Conference on Recent Advances in Electronics and Communication Technology (ICRAECT), 1-5 (2020).

[31] Alenazi, M. R., Al-Salman, R., & Al-Turjman, F. "Security Issues and Challenges in Software Defined Networking: A Survey." IEEE Communications Surveys & Tutorials, vol. 23, no. 1, pp. 359-388 (2021).

[32] Islam, S. M. R., Alam, M. M., Hasan, M. N., & Al-Fuqaha, A. "A Survey on Security and Privacy Issues in Software-Defined Networking (SDN)." *Journal of Network and Computer Applications,* vol. 183, pp. 102-124 (2021).

[33] Umamaheswaran, S., John, R., Deepthi, S. S., & Dharmarajlu, S. M., Caption positioning structure for hard of hearing people using deep learning method. *Journal of Discrete Mathematical Sciences and Cryptography*, 25(3), 623-633 (2022).

[34] Shah, S. P., Panchal, P. B., & Jain, R. K. "Security Issues and Challenges in Software-Defined Networking: A Review." *International Journal of Computer Science and Mobile Computing*, vol. 10, no. 1, pp. 41-51 (2021).

[35] Ali, A. S., Jawad, M. A., Hussain, S. F., Al-Sammarraie, A., & Ali, H. "A Survey on Security Challenges and Solutions for Software-Defined Networking." *Future Computing and Informatics Journal*, vol. 6, no. 2, pp. 337-348 (2021).

[36] Gupta, V., Juyal, S., Singh, G. P., Killa, C., & Gupta, N., Emotion recognition of audio/speech data using deep learning approaches. *Journal of Information and Optimization Sciences*, 41(6), 1309-1317 (2020).

*Received September 2023*