Computers, Materials & Continua
DOI:10.32604/cmc.2022.021669
Article

Tech Science Press

DDoS Detection in SDN using Machine Learning Techniques

Muhammad Waqas Nadeem, Hock Guan Goh*, Vasaki Ponnusamy and Yichiet Aun

Faculty of Information and Communication Technology (FICT), Universiti Tunku Abdul Rahman (UTAR)
Jalan Universiti, Bandar Barat, 31900 Kampar, Perak, Malaysia
*Corresponding Author: Hock Guan Goh. Email: gohhg@utar.edu.my
Received: 10 July 2021; Accepted: 20 August 2021

Abstract: Software-defined network (SDN) becomes a new revolutionary paradigm in networks because it provides more control and network operation over a network infrastructure. The SDN controller is considered as the operating system of the SDN based network infrastructure, and it is responsible for executing the different network applications and maintaining the network services and functionalities. Despite all its tremendous capabilities, the SDN face many security issues due to the complexity of the SDN architecture. Distributed denial of services (DDoS) is a common attack on SDN due to its centralized architecture, especially at the control layer of the SDN that has a network-wide impact. Machine learning is now widely used for fast detection of these attacks. In this paper, some important feature selection methods for machine learning on DDoS detection are evaluated. The selection of optimal features reflects the classification accuracy of the machine learning techniques and the performance of the SDN controller. A comparative analysis of feature selection and machine learning classifiers is also derived to detect SDN attacks. The experimental results show that the Random forest (RF) classifier trains the more accurate model with 99.97% accuracy using features subset by the Recursive feature elimination (RFE) method.

Keywords: Machine learning; software-defined network; distributed denial of services; feature selection; protection; artificial neural network; decision trees; naïve bayes; security

1 Introduction

The software-defined network (SDN) paradigm gained the most significant interest in current days. The data centres and operator networks are shifting from traditional networks to SDN based networks because it provides more reliable, flexible and secure network environment [1–3]. Consequently, the deployment of the SDN in data centres and cloud computing environments provide reliable and flexible network architecture. The separation of control and data planes is the main innovation behind the SDN. Furthermore, the SDN provides an intelligent centralization that consists of controllers that manage the forward packet devices, and the well-designed configuration like (Open-Flow) of these devices is essential [4,5]. In the SDN, network devices like switches only forward logic, whereas the decision making and control logic ability are software at an SDN

controller. The SDN controller makes the new network flow policies and instructs the switches to follow the new policies maintained in a flow table [6]. Despite all these impressive innovations, the SDN-based network environment's several components pose some additional security threats to the SDN controller. So, the security of the SDN controller is becoming a key challenge for the development of future SDN based networks. As far as many issues to be addressed, all the security of the SDN controller is considered the highest concern. Distributed denial of service (DDoS) is a critical security issue for the SDN controller. The main aim of a DDoS attack is to make the computing resources unavailable for the users. The DDoS attack is usually caused by one or more than one bot, which is penetrated by software from malicious code. The DDoS attack also become a cause of massive damage to the network because it spreads more quickly. Furthermore, the SDN controller loses its centralized control when it is under of DDoS attack because the controller is separated from the rest of the network.

Furthermore, the detection and mitigation technologies for the DDoS attack in SDN environments is needed. In the DDoS attack, a large number of packets being forward toward the target network. If the destination and source IP addresses of the forward packets are forged, and switches do not find in its flow table entries, then unmatched flows consider as new flows. After that, the switch sends that unmatched packet to the SDN controller or directly forward the packet. The SDN controller is responsible for finding the forward paths of these packets. In legitimate traffic, many DDoS flows are hidden, which continuously consume the controller's resources, and ultimately, the controller's resources become unavailable for the upcoming new packets. As a result, the SDN controller shut down, and the whole network goes downstate. Disappointedly, this security issue still exists even in the presence of the backup controller [7]. The characteristics of the DDoS attack in an SDN environment is quietly different from the traditional networks. By analyzing the principles of the DDoS attack on the SDN controller, it can be concluded as below:

- In traditional networks, one is more network links or DDoS attackers target destination servers. In SDN, the DDoS attack is carried out on the controller. The primary purpose is to make the controller resources unavailable by failing a single point in SDN.
- In traditional networks, the IP addresses of the packets are real. That is why the DDoS attacker usually targets the terminal server. In the SDN, to launch a DDoS attack, the attacker tries to forge the IP addresses of the destination that involve the controller into continuous processing with new flows. The resources of the controller go to the unavailable state.
- In traditional networks, the server stops providing services to legitimate users when it suffers a DDoS attack. The controller lost contact with the data plane in the SDN and failed to provide services for the forwarding data packets when it suffers a DDoS attack.

In traditional networks, the detection of DDoS attacks is relatively at the mature stage. However, the detection of DDoS attacks in the SDN environment is still an open security issue because the SDN is a new paradigm in computer networks. In an SDN environment, the detection techniques, which are primarily implemented in the traditional networks, are followed to detect DDoS attacks in SDN without knowing the characteristics of the attack. The existing DDoS attack methods applied on the SDN controller do not efficiently detect the attack in SDN due to different characteristics of the SDN architecture. The SDN controller needs to continuously collect the network traffic information from the switches to determine the occurrence of the DDoS attack, which increases the controller's workload.

Consequently, to overcome these challenges, the intelligent detection of DDoS attacks in SDN is needed. In recent years, several Machine learning (ML) methods and techniques for detecting DDoS attacks in SDN have been proposed by the researchers. The machine learning techniques and approaches use historical data to learn the network behaviour and predict the upcoming packets. ML-based techniques have shown effective results for the classification of normal and attack traffic. Furthermore, the ML techniques require a particular set of features that includes source and destination IP addresses, source and destination port numbers, flow durations etc., of the network flows [6]. ML techniques also have lower computation costs than the Deep packet inspection (DPI) techniques [8–11].

Furthermore, in the SDN infrastructure, when the controller is under a DDoS attack, both forwarding and controller layers suffer from resource depletion. The existing research has shown improvement in detecting DDoS attacks at the control layer [8,12–14]. A large dimensional and large volume of network traffic data is used in machine learning-based techniques. Selecting the most relevant features from the data set for the training and testing of machine learning models is still an open issue. The optimal selection of packet features is important because it influences the design of the efficient ML-based detection model for the DDoS in SDN. Motivated by this, in this paper, a comparative analysis of feature selection based ML classifiers is present for the efficient detection of DDoS attacks in the SDN. Different feature selection techniques that include filter, wrapper, and embedded are used to find out the most optimal subset of features for the training and testing of ML algorithms. The optimal selection of features subset is significant because it reflects the overall accuracy of the machine learning classifier. The literature survey related to the SDN has concluded that the SDN controller's control plane exhibit more vulnerabilities for the DDoS attack, and most of the researchers preferred ML techniques [15–19]. Moreover, selecting the optimal features from the data set is still insignificant because the use of a large number of features for the ML models increases the cost and time complexities. Furthermore, the use of irrelevant features is not able to detect the attack more efficiently.

The main contribution of the paper is as follow:

- This work utilizes different feature selection techniques to find the most optimal features for the training and testing machine learning models. The machine learning model with optimal features gives effective results for detecting DDoS attacks in SDN controllers.
- The presented analysis combines features selection techniques with a machine learning model. Further, to reduce the training time of the machine learning models, the subset of optimal features is used.
- The presented analysis compares the results of machine learning classifiers on a different subset of features that are ranked by the feature selection methods. The experimental results have shown the effectiveness of the machine learning classifiers with feature selection methods to detect DDoS attacks in SDN.

The rest of the paper is organized as follows. Section 2 briefly describes the related work in this field. In Section 3, the mechanism of DDoS attacks in SDN is elaborated. Section 4 describes the material and method of this work. In Section 5, the experimental results of the machine learning classifiers for DDoS attack detection in SDN. Finally, the conclusion of the paper is discussed in Section 6.

2 Related Research

Machine learning techniques give a compelling performance for the detection of DDoS attacks in SDN. The ML techniques effectively detect the attack against the control plane of the SDN controller. This section briefly discusses the current works to detect DDoS attacks in SDN using machine learning techniques. Furthermore, features selection based ML models and techniques presented by the researchers in recent years are analyzed in the section. In [15], a statistical and machine learning-based method is proposed. K-mean and K nearest neighbors (KNN) based hybrid model is proposed in [16]. Support vector machine (SVM) based DDoS detection in SDN was performed in [17]. Kernel principal component analysis (KPCA), Genetic algorithm (GA) and SVM based method is presented in [18].

In [20], an entropy-based technique that uses Flow samples are presented for traffic classification, and it just focuses on a standard distribution of the traffic. A COFFEE model that extracts the features from the flow for the detection of attack is present in [21]. For the extraction of more features, the suspected flow sends to the controller. In [22], many features are utilized by the machine learning techniques to detect the attack.

Furthermore, in [23], traffic features based on a lightweight DDoS attack detection algorithm is presented. The Self-organizing map (SOM) is used for the analysis and extraction of traffic information. After the extraction of features Artificial Neural Network is used for the detection of DDoS attacks. In [24], the researchers proposed a k-nearest neighbor-based algorithm that uses the abstract distance between the traffic features to detect the attack. This algorithm gives effective results for the detection of abnormal flow and also reduce the false alarm rate. Although the researchers proposed various machine learning-based solutions for detecting DDoS attacks, these solutions have some limitations in terms of optimal feature selection, low accuracy and efficiency.

In [25], Naïve bayes (NB) and K-mean clustering-based method was proposed to detect DDoS attacks. The K-mean cluster method clusters the traffic data that show similar behaviours and the Naïve Bayes algorithm classifies the cluster data into standard and attacks traffic. In [26], Artificial Neural Network-based methods are proposed to detect known and unknown DDoS attacks. The researcher in the controller applies a dynamic Multilayer perceptron (MLP) that works with a feedback mechanism to detect DDoS attacks [27]. They use some selected features that cannot distinguish between standard and attack traffic flows.

In [28], the authors introduced the trigger mechanism that detects the DDoS attack faster and reduces the switches' workload. The trigger mechanism applied on the controller's control plane effectively attacks but increases the controller's workload. Zang et al. [29] proposed a finer-grained method that uses the flow features to detect an attack. It extracts the 39 different traffic features from the flow and improves the detection accuracy.

3 DDoS Attack in SDN

In a DDoS attack, the rate of incoming packets towards the network becomes high. Hence, the collection of spoofed and legitimate packets binds the network resources that make the resources unavailable. If this process continues, the server starts to drop the packets, and it becomes unreachable for the new incoming legitimate packets. The DDoS attack is categorized into three types: application-layer attack, protocol-exploitation attack, and volumetric attack. The TCP flooding and UDP flooding attacks are considered volumetric attacks, whereas the DNS and HTTP flood are referenced as application-layer attacks [30]. The control plane of the SDN

controller has centralized network intelligence. In a single SDN controller-based architecture, there is a high chance of a Single point of failure (SPF).

When the attacker gets access to the controller, it gives massive damage to the network's infrastructure [31]. At the top of the control plane, the applications that include routing, firewall and load balancing are operated. If the attacker accesses the firewall application, the controller forms a different Access control list (ACL) [32]. A secure connection is created between the OF switch and controller using TLS/SSL; if the TLS connection goes on downstate, it needs a backup controller for the switch. In such a case, the OF switch will use the flow table according to his choices. A DDoS attack may create onto the controller when a malicious flow can be rule into the flow table [33–35]. Besides this, in SDN, the format of the flow has some essential properties. The SDN controller uses the southbound protocol that includes OpenFlow to take action against the flow entries. The same flow in SDN maybe has more than one rule for it. The flow has different fields that include timeout, counter, action field, priority, etc. Each field has its specific task. For example, the timeout field represents the expiration time for a flow. The counter field keeps the information regarding bytes per flow, and the instruction field identifies the needed action for a flow entry. Fig. 1 describes the discussed scenario.

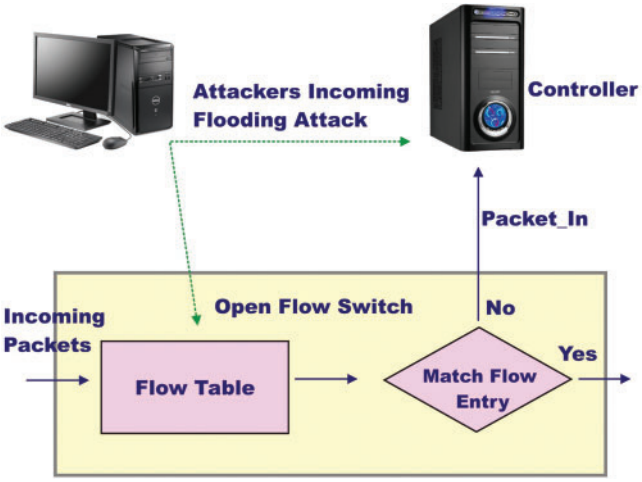

Figure 1: Systematic diagram of attack in SDN controller

4 Material and Method

This section describes the material and method of the paper. The systematic diagram of the process steps for the implementation of feature selection methods and machine learning classifiers is presented in Fig. 2. Additionally, this section also describes the description about dataset, feature selection methods and machine learning classifiers. This process consists of different phases that includes data selection, feature selection, data partitioning and modeling.

4.1 Data Set

This section briefly describes the description of the data set that has been used in this work. NSL-KDD [18] dataset is used to for the training and testing of machine learning classifiers for the detection of DDoS attack. The NSL-KDD is the refined version of the KDD'99 dataset. The data set has 41 features and 52,800 records that have been considered for the simulation.

The NSL-KDD has the data of different type of attacks that includes DDoS, Probe, R2L, U2R and normal traffic. The data related to DDoS attack is extracted from the original data set and it used to evaluate the performance of different machine learning classifiers for the detection of DDoS attack in SDN. The ML classifiers perform efficiently and evaluate accurately. The features of the NSL-KDD dataset are listed in Tab. 1.

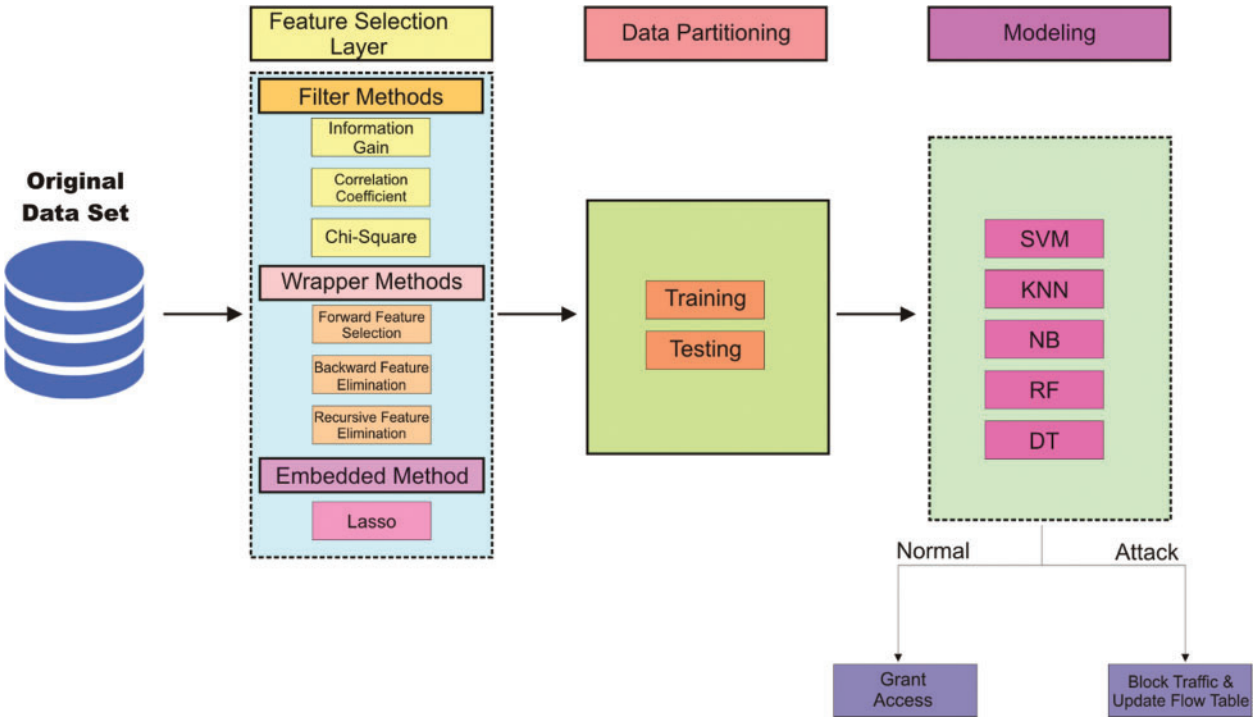

Figure 2: Systematic diagram for the implementation of features selection and machine learning methods

Table 1: Features of the dataset [18]

No.	Feature name	No.	Feature name
1	Duration	22	Srv-count
2	Source bytes	23	Serror-rate
3	Destination bytes	24	Srv-serror-rate
4	Land	25	Rerror-rate
5	Wrong fragment	26	Srv-rerror-rate
6	Urgent	27	Same-srv-rate

(Continued)

Table 1: Continued

No.	Feature name	No.	Feature name
7	Hot	28	Diff-srv-rate
8	Number-of-failed-login logins	29	Srv-diff-host-rate
9	Logged-in	30	Dst-host-count
10	Num-compromised	31	Dst-host-srv-count
11	Root-shell	32	Dst-host-same-srv-rate
12	Su-attempted	33	Dst-host-diff-srv-rate
13	Num-rootNum-file-creations	34	Dst_host_same_src_port_rate
14	Num-shells	35	Dst_host_srv_diff_host_rate
15	Num-access-files	36	Dst_host_serror_rate
16	Num-outbound-cmds	37	Dst_host_srv_serror_rate
17	Is-host-login	38	Dst_host_rerror_rate
18	Is-guest-login	39	Dst_host_srv_rerror_rate
19	Count	40	Protocol_type
20	Num_file_creations	41	Flag
21	Service		

4.2 Feature Selection in Machine Learning

In machine learning, feature selection task is important because the use of irrelevant features by the ML model can increase the running time and the cost of the system. It also affects the performance of the models and makes the generalization performance of the models much poorer. In general form, the feature selection method for a learning problem from data can be formulated as: for a given set of labeled data samples $(x_1, y_1), \ldots\ldots (x_i, y_i)$ where $x_i \varepsilon R^n$ and $y_i \varepsilon R (or\ y_i \varepsilon \{\pm 1\}$ for a classification problem), the variable y_i achieved the lowest prediction error for a subset of m features (m < n). There are several algorithms that can be used for the selection of optimal features. In general, they can be structured into three different categories that includes filter, wrapper and embedded.

4.2.1 Filter Methods

In filter methods, the variable ranking technique is used as a principle for the selection of features. Ranking methods are well known due to their simplicity and also have good success for practical applications. For the selection of the features, a suitable ranking criteria and threshold is used. The features that are below from the threshold value will be discarded from the feature subset. Furthermore, ranking methods, also known as filter methods, are applied on the dataset before classification to filter out less relevant features. The basic property of the optimal features is that they contain the information related to different classes that are present in the data. This property describes the relevance of the features and provides a way for the measurement of features usefulness for different classes. Furthermore, the relevant feature does not depend on input data, but they depend on the class label i.e., if the feature has no influence on the class labels than it can be discarded. The correlation is used in filter methods for the determinations of the unique or optimal features. The correlation and mutual information both methods are used in filter methods for the selection of optimal feature subset. These methods are formulated as:

$$R_i = \frac{\mathrm{cov}\,(x_i, Y)}{\sqrt{\mathrm{var}\,(x_i) * \mathrm{var}\,(Y)}} \tag{1}$$

where in Eq. (1) x_i is the input variable, Y represents the output, *cov* is the covariance, and *var* is the variance.

$$H(Y) = -\sum_{y} p(y) \log(p(y)) \tag{2}$$

In Eq. (2) Y is the uncertainty information output, and $p(y)$ is the probability for each input. So, in this paper, three different filter methods that include Information gain (IG), Correlation coefficient (CC), and Chi-Square have been used for the selection of optimal features.

4.2.2 Wrapper Methods

The wrapper methods use itself as a regression or classification model and search the good subset of features through evaluation function. The classification model with different subsets of features is run on the training dataset. The features which produce lowest estimated error will be chosen as optimal features. The mathematical formulation of the wrapper methods as follows: The main goal of the feature selection method is to find the optimum n-column vector σ where $\sigma \varepsilon \{0, 1\}$, that is defined as the optimal subset of selected features, which can be further defined as

$$\sigma^{\theta} = \arg\min \left(\int \mathrm{V}\left(\mathrm{y}, \mathrm{f}\left(\mathrm{x}^{*}\sigma, \alpha\right)\right) \mathrm{dP}(\mathrm{x}, \mathrm{y}) \right) \tag{3}$$

In the above Eq. (3), V represents the loss function, $P(x, y)$ is the unknown probability function that was defined for the sampled data $x^{*}\sigma = (x_1\sigma_{1,\ldots\ldots}, x_n\sigma_n)$. The function $y = f(x, \alpha)$ represents the classification engine that is evaluated for each selected feature subset σ and for each subset of its hyper-parameter α. This paper uses three different wrapper methods that include Forward feature selection (FFS), Backward feature elimination (BFE), and Recursive feature elimination (RFE) for the raking of most optimal features.

4.2.3 Embedded Method

The embedded feature selection methods are included in the classification and regression models. In these algorithms, the features that best contribute for the accuracy of the model will be selected as optimal subset of features. The wrapper based and filter based methods have been combined for the development of the embedded methods. Furthermore, they take the advantage of the feature selection process and performs the features selection and classification simultaneously [36]. The Lasso embedded feature selection algorithm has been used for the selection of optimal features because it eliminates the weights of the least important features and gives a reduced set of features. The Lasso method is formulated is as:

$$\sum_{\mathrm{i}=1}^{\mathrm{N}} \left(\mathrm{y_i} - \sum_{\mathrm{j}} \mathrm{x_{ij}} \beta_{\mathrm{j}} \right)^{2} + \lambda \sum_{\mathrm{j}=1}^{\mathrm{p}} |\beta_{\mathrm{j}}| \tag{4}$$

In the above Eq. (4) the β_j is the regular coefficient of the classifier model, x_{ij} is the standardized set of features and y_i is the centered output, λ control the shrinking of the coefficient, for i in [1, N] where N represents the number of observation, and j in [1, P] where P denotes the number of features. The goal of the Lasso function is to minimize the number of features.

4.3 Machine Learning Classifiers

The set of optimal features which were selected by different feature selection methods are used as an input for different machine learning classifiers. The brief description of these classifiers is as below:

4.3.1 Random Forest

This section describes the general framework of the Random forest (RF) model. The RF classifier model consists of 1000 trees, and minimum number leaf node is 1. Furthermore, in the RF model every weak learner was grown to its maximum, unpruned, and 63% observations of the feature subset $\sqrt{m}$ was provided for the bootstrap, where m represents the number of features, and all optimal features are used by the RF model.

4.3.2 Support Vector Machines

The support vector machine model is implemented with radial basis kernel function where the optimal hyperplane is computed as:

$$\text{argmin}\ \left(\mathrm{w}\frac{1}{2}\mathrm{w}^{\mathrm{t}}\mathrm{w}+\mathrm{C}\sum_{\mathrm{i}=1}^{\mathrm{N}}\varphi_{\mathrm{i}}\right) \tag{5}$$

Subject to $y_i(w^t \varnothing (x_i) + b \geq 1 - \varphi_i)$ where φ_i is ≥ 0, $i = 1, \ldots, N$, where C represents the cost factor that penalizes the miss-classification in the training data, w is the vector coefficient, b is the constant intercept term, and φ_i is used to control the margins on each side of hyperplanes, and $\varnothing$ represents the radial basis kernel function. In SVM, the radial basis kernel function gives better accuracy as compared to the linear kernel function.

4.3.3 K-Nearest Neighbors

K-nearest neighbors is a supervised machine learning classifier that is simple and can be easily used to solve regression and classification problems. The nearest k neighbors mechanism is used to determine the class for the new upcoming data. The Euclidean and Manhattan distance functions are used for the measurement of the distance between two data. In this paper, the Euclidean distance function is used. The similarity between the data samples that to be classified and the sample that were found in the classes was distinguished. The Euclidean distance function calculates the distance between the new encountered data and the data which is present in the training set individually. After that, the classification set is created by selecting the k dataset which has the smallest distance. The number of KNN neighbors is based on the value of classification. The mathematical formulation is presented as below:

$$\mathrm{d}\,(\mathrm{q},\mathrm{p}) = \sqrt{\sum_{\mathrm{i}=1}^{\mathrm{n}}(\mathrm{q_i}-\mathrm{p_i})^2} \tag{6}$$

where in Eq. (6) q_i is the data points in the data set and p_i represents the probability of each data point.

4.3.4 Naïve Bayes

The Naïve Bayes algorithm is based on the bayes rules that uses conditional independences of the features $X, [X_1, X_2, X_3, \ldots\ldots X_n]$ and corresponds to the output $Y \varepsilon [0, 1]$. Furthermore, the

Gaussian NB model uses probability for the estimation class of continuous predictive features. The probability of features for the different classes is computed as:

$$P(\mathrm{Y}=1|\mathrm{X}) = \frac{1}{1+\exp(\mathrm{w}_\circ + \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{x_i w_i})} \tag{7}$$

where w_i represents the weights that are $w_1, w_2, \ldots\ldots w_n$ and computed with $w_i = \frac{\mu_{i0}-\mu_{i1}}{\sigma_i^2}$ and

$$\mathrm{w}_\circ = \ln \frac{\mathrm{P(Y=0)}}{\mathrm{P(Y=1)}} + \sum_{\mathrm{i}} \frac{\mu_{\mathrm{i0}} - \mu_{\mathrm{i1}}}{\sigma_{\mathrm{i}}^2} \tag{8}$$

where σ_i represents the standard deviation and μ_i is the mean of the feature x_i.

4.3.5 Decision Trees

The Decision tree (DT) consists of nodes and branches. It arranges the knowledge extracted from the data in the recursive hierarchical structure. Each internal node is represented as an attribute, and it is associated with the relevant data for classification. The classes in the data set are corresponding by the leaf nodes of the tree, and branches represent the possible results. The new input can be classified continuously by the nodes and branches until the lead node is reached. The main aim of the induction process in the DT is to maximize the correct classification for all the training data.

Furthermore, the pruned process is applied to the trained tree to avoid overfitting. The decision tree also generates a comprehensive structure of the classification. For each new data, the final classification is determined through the verification of different attributes. In DT, the Gini index is used for the classification which as formulated as:

$$\mathrm{Gini}\,(\mathrm{L_k}) = 1 \sum_{\mathrm{c}=1}^{\mathrm{c}} -(\mathrm{p(c|L_k)})^2 \tag{9}$$

where in Eq. (9) p is the probability, c is the class and L_k represents the leaf node for each class.

4.4 DDoS Detection Model

The machine learning-based model for the detection of DDoS attacks in SDN is presented in Fig. 3. The model monitors the OpenFlow (OF) switches for time intervals, and the SDN controller sends the flow results to all the switches present in the network. Furthermore, the SDN controller receives the statistics of the flow, and after that, it fed these flows into the statistics monitor for the extraction of features. The extracted features send to the feature selection method to select the optimal features that are important for the detection of DDoS attacks. After feature selection, these features are sent to the machine learning classifiers that predict the normal and attack traffic flow. If the ML classifier detects the DDoS attack, then the mitigation module present in the SDN controller immediately sends new flow rules to the switches to drop the upcoming packets.

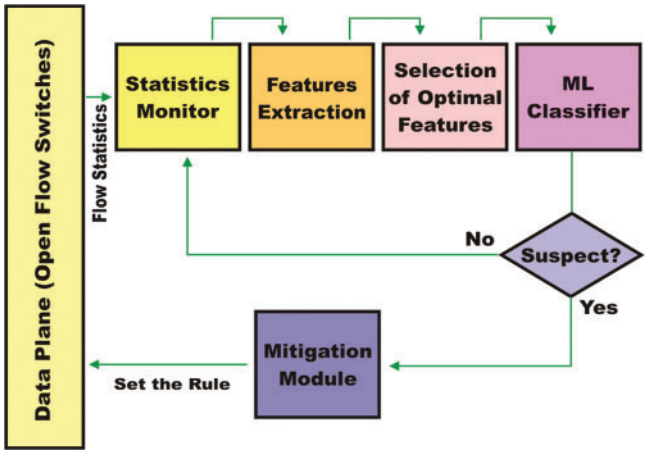

Figure 3: Machine learning based DDoS detection model

5 Experimental Results

To evaluate the performance of the machine learning classifiers, different evaluation metrics that includes accuracy, precision and recall have been selected. Confusion matrix is used to compute these evaluation metrics. The performance evaluation of the machine learning classifiers is important for the accurately detection of attack in the SDN controller. The mathematical formulation of these metrics is as follow:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

Among these formulas, TP (True positive) is the probability of the attack traffic which is recognized as attack; TN (True negative) is the probability of normal traffic which is known as normal traffic; FP (False positive) is referred as the probability of the normal traffic which is recognized as attack traffic and FN (False negative) is the probability of the attack traffic which is recognized as normal traffic. The classifier which achieved the higher recall rate and accuracy gives better detection performance for the attack.

In this section, the performance of the various classifiers has been evaluated on different optimal features subsets that is selected by feature selection methods. The classifiers that include SVM, KNN, NB, RF and DT are selected. Furthermore, the filter, wrapper and embedded methods are used to rank the most optimal features. These methods include Information Gain, Correlation Coefficient, Chi-Square, Forward Feature Selection, Backward Feature Selection, Recursive Feature Elimination and Lasso. The accuracy curves for these methods are shown subsequently in Figs. 4–10.

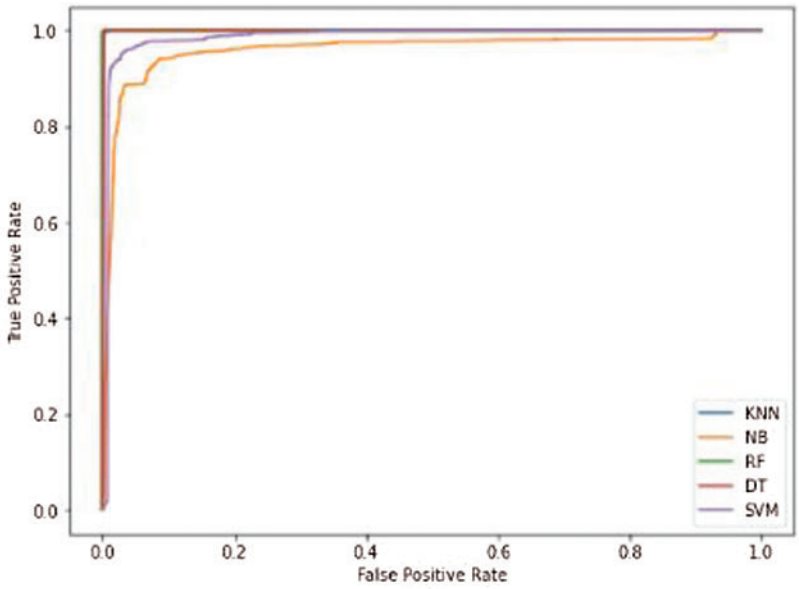

Figure 4: Accuracy curve for information gain

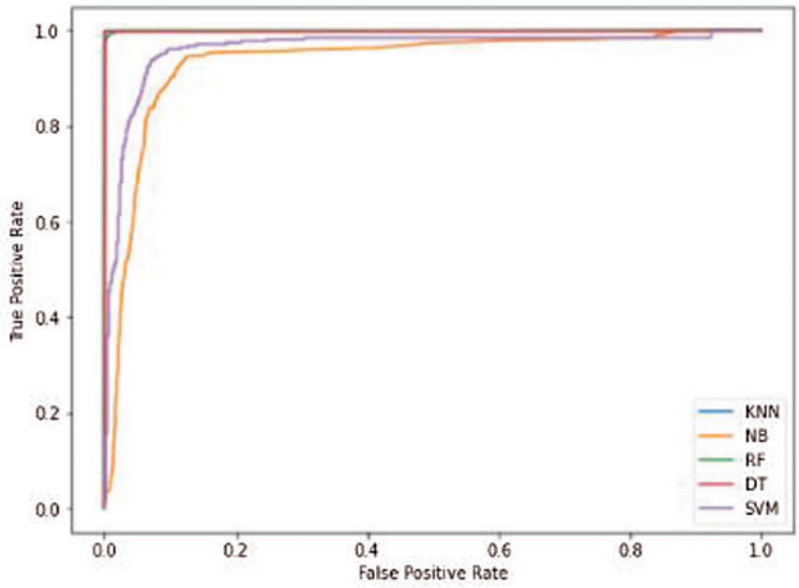

Figure 5: Accuracy curve for correlation coefficient

Based on the results, the classification accuracy of the different classifiers goes on the stable state on different subsets of the features. Here, stable accuracy is defined as some classifiers achieving maximum accuracy at the same set of features. Therefore, it is imperative to find the top-optimal features subsets on which classifiers offer better results for detecting the attack. Compering the accuracy of the different classifiers, the RF classifiers consistently achieved 99.97% accuracy on the feature's subset, which is ranked by the RFE method. Therefore, our finding is that the RF classifier gives better results than the other classifiers for detecting DDoS attacks.

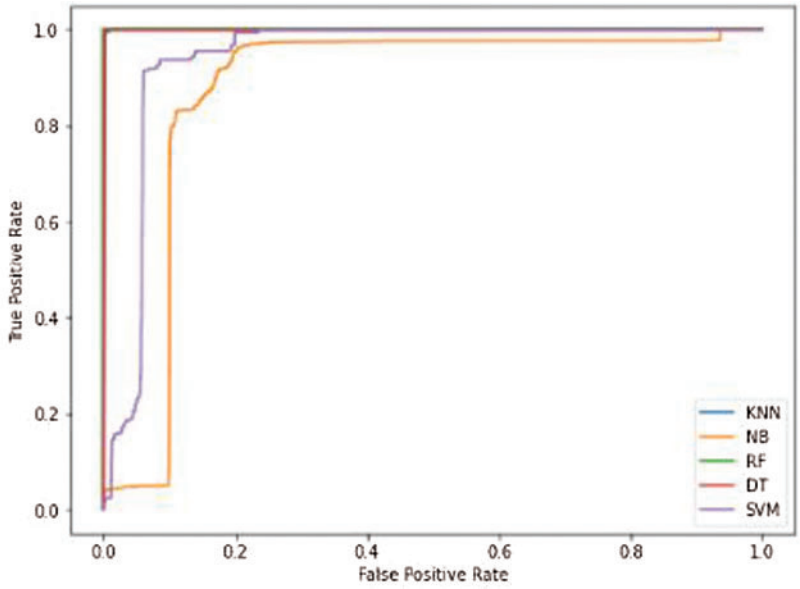

Figure 6: Accuracy curve for chi-square

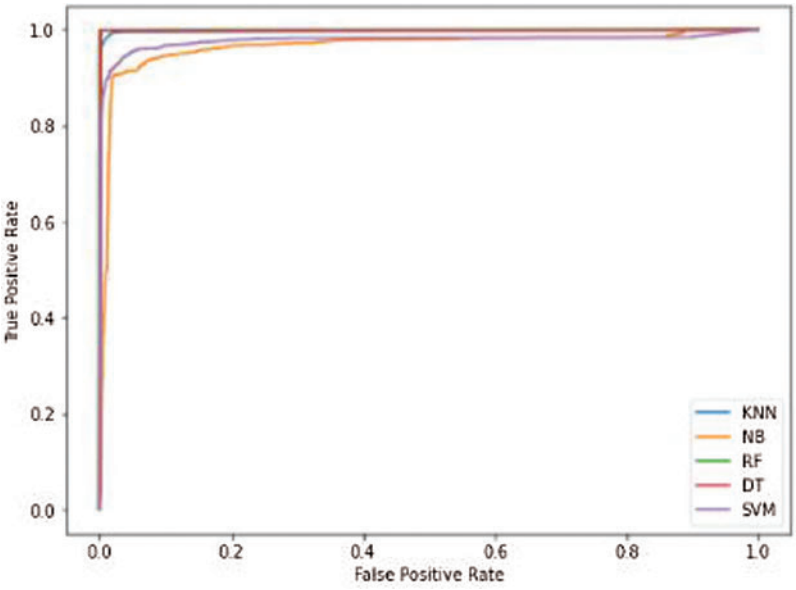

Figure 7: Accuracy curve for FFS

Furthermore, a comparative analysis of the different classifiers on different subsets of the optimal features is presented in Tab. 2. The table shows the number of features selected by each method and the accuracy of the classifiers on this subset of features. It is observed in the table that the maximum accuracy, which is 99.97%, is achieved by the RF classifier on the RFE feature subset. The favourable results of the RFE methods cloud are attributed to the fact that the

RF classifier and RFE are both founded on similar information metrics. Furthermore, the other classifiers that include SVM, NB, KNN and DT also give better results on the RFE feature subset compared to the features subset that is ranked by the other feature selection methods. So, it is also concluded that the 28 features selected by the RFE method are most optimal for detecting DDoS attacks in SDN.

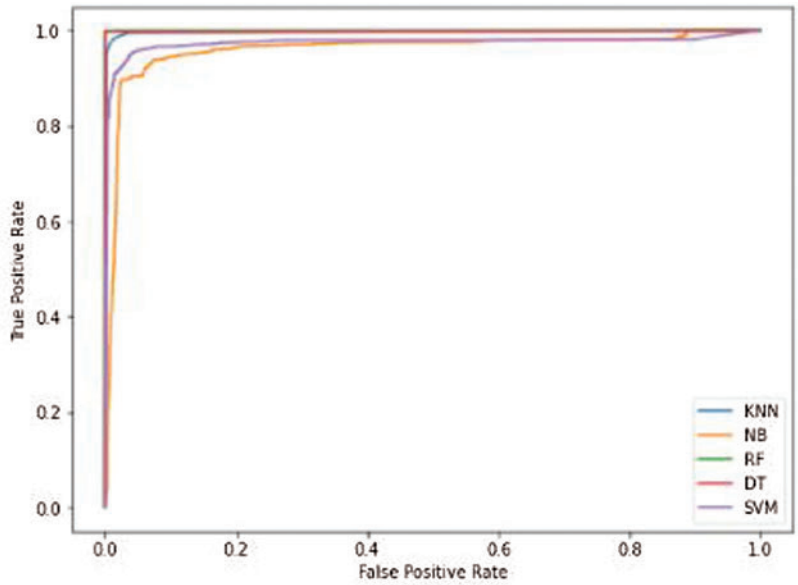

Figure 8: Accuracy curve for BFE

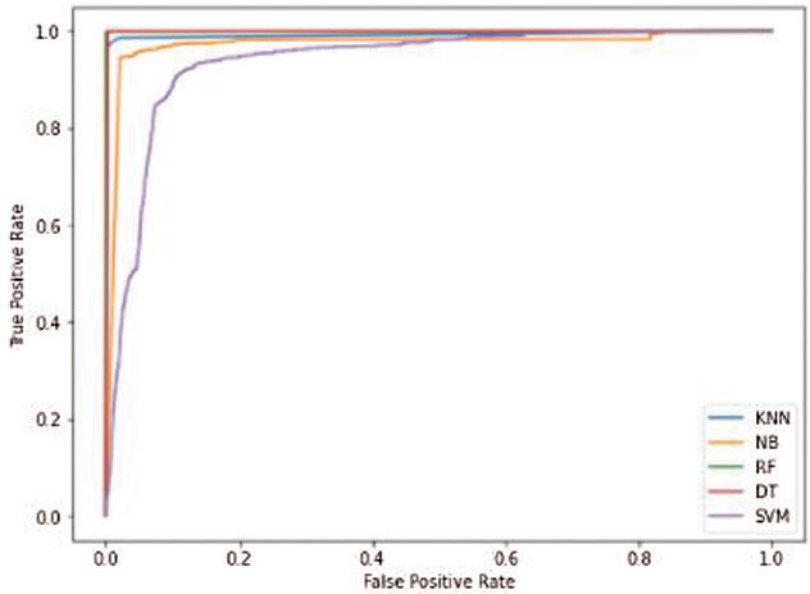

Figure 9: Accuracy curve for RFE

On the other hand, the Chi-Square method relatively achieved low initial accuracy of 48.77% by the NB classifier. It is also observed that the accuracy of the second highest, which is 99.94%, is achieved by the RF classifiers on the Backward feature elimination feature subset. However, it selects the 34 number features that increase the computational time and complexity of the classifier. Therefore, based on the analysis of results, it is reasonable to conclude that the RFE method selects the top-fittest features for the classifiers. Furthermore, the other evaluation metrics that include precision, recall and specificity of the classifiers were also computed using a confusion matrix. The Figs. 11a–11c show the comparison of the different classifiers in term of precision, recall and specificity respectively.

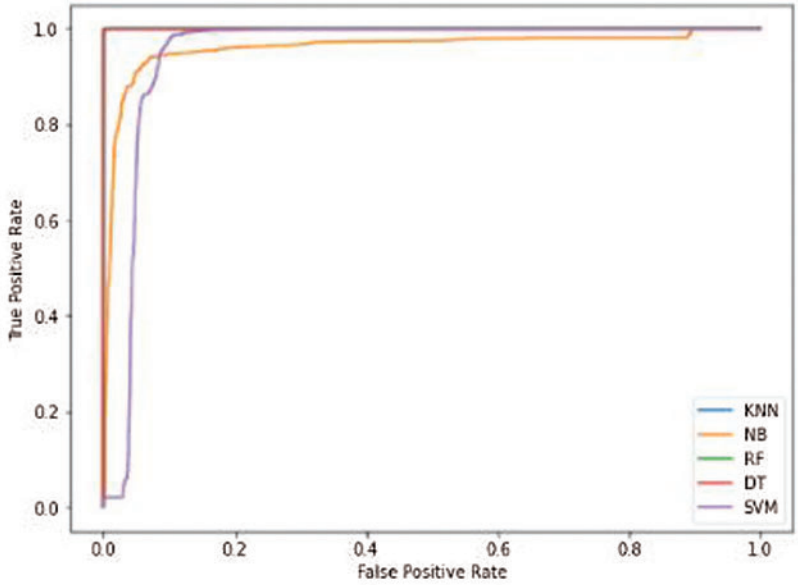

Figure 10: Accuracy curve for Lasso

Table 2: Comparison of experimental results

ML Classifiers	Filter method			Wrapper method			Embedded
	Information gain	Correlation coefficient	Chie-square	Forward selection	Backward elimination	Recursive elimination	Lasso
Number of selected features							
	20	23	24	34	34	28	24
Accuracy							
SVM	60.19%	80.61%	59.45%	58.60%	59.62%	89.18%	59.79%
KNN	99.57%	99.19%	99.41%	98.66%	98.50%	98.57%	99.54%
NB	85.08%	63.79%	48.77%	84.79%	83.42%	92.12%	69.43%
RF	99.92%	99.91%	99.91%	99.91%	99.94%	**99.97%**	99.90%
DT	99.88%	99.83%	99.85%	99.80%	99.86%	99.80%	99.83%

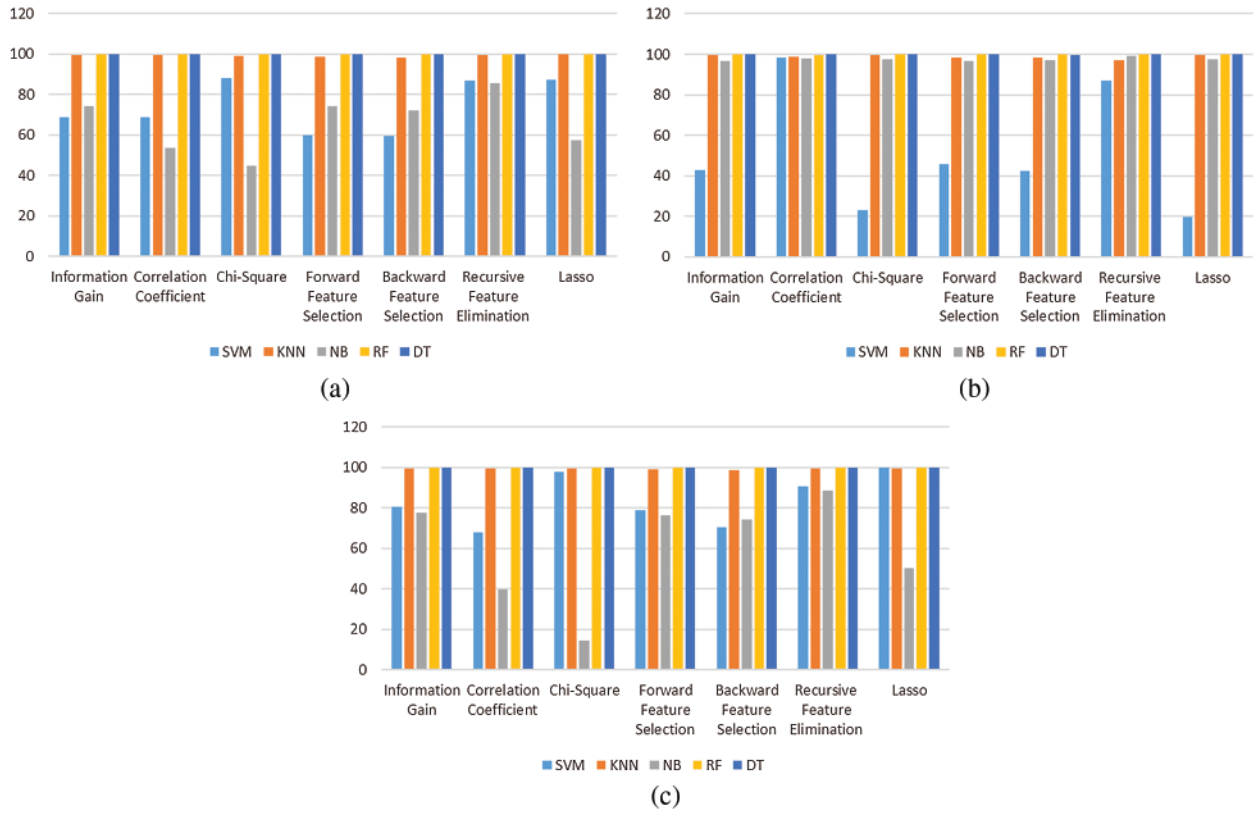

Figure 11: (a): Classifiers result in term of precision (b): Classifiers result in term of recall (c): Classifiers result in term of specificity

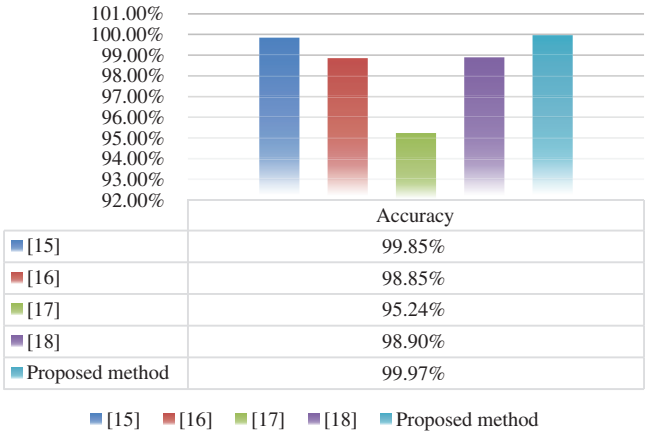

Figure 12: Performance comparison of the proposed method with existing solutions in term of accuracy

A comparative analysis of the proposed method with other recent machine learning development for DDoS attack detection in SDN is shown in Fig. 12. The accuracy evaluation matrix is used for the comparison purpose. It is clearly observed in Fig. 12 that the proposed method gives better accuracy as compared to the existing research for the DDoS attack detection in SDN.

6 Conclusion

Although the SDN makes considerable advancements in networking, it faces several security issues, where the most common security issue for the SDN is DDoS attacks. The SDN controller is the centralized control of the whole network and it becomes more vulnerable to DDoS attacks can reach there. Hence, the intelligent detection of DDoS attacks in the SDN is needed. In response to this problem, this paper presents a comparative analysis of different machine learning classifiers based on the optimal subset of features for the early and accurate detection of DDoS attacks over the SDN. The combination of the machine learning classifiers and the advantages of the SDN protects the SDN controller from DDoS attacks.

Furthermore, the extraction and selection of optimal features for the machine learning-based models are also crucial for accurately detecting an attack. The experimental results prove that the RF classifier on recursive feature elimination ranked features subset achieves good results for detecting attacks in the SDN controller. However, the resources consumption of the SDN controller increases and the detection accuracy of the DDoS attack decreases when the network is under a larger-scale network traffic. Furthermore, the use of an irrelevant and many features also increase the SDN control's workload, which may affect the controller's efficiency. In the future, these machine learning classifiers and feature selection techniques would also be used to detect the other attack classes such as smurf, Probe, R2L and U2R, in the SDN. Although in the environment of a single controller, this work is well for detecting DDoS, it may fail to detect the attack traffic in a multi-controller environment. So, in the future, these models are also evaluated to detect attacks in a multi-controller context.

Acknowledgement: The authors thank their families and colleagues for their continued support.

Funding Statement: No funding was received to support any stage of this research study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Y. Zhang, L. Cui, W. Wang and Y. Zhang, "A survey on software defined networking with multiple controllers," *Journal of Network and Computer Applications*, vol. 103, no. 3, pp. 101–118, 2018.

[2] P. Visu, L. Lakshmanan, V. Murugananthan and M. V. Cruz, "Software-defined forensic framework for malware disaster management in internet of thing devices for extreme surveillance," *Computer Communications*, vol. 147, no. 5, pp. 14–20, 2019.

[3] E. Molina and E. Jacob, "Software-defined networking in cyber-physical systems: A survey," *Computers & Electrical Engineering*, vol. 66, no. 11, pp. 407–419, 2018.

[4] A. Mondal, S. Misra and I. Maity, "AMOPE: Performance analysis of openflow systems in software-defined networks," *IEEE Systems Journal*, vol. 14, no. 1, pp. 124–131, 2019.

[5] M. Conti, C. Lal, R. Mohammadi and U. Rawat, "Lightweight solutions to counter DDoS attacks in software defined networking," *Wireless Networks*, vol. 25, no. 5, pp. 2751–2768, 2019.

[6] C. B. Zerbini, L. F. Carvalho, T. Abrao and M. L. Proenca Jr, "Wavelet against random forest for anomaly mitigation in software-defined networking," *Applied Soft Computing*, vol. 80, pp. 138–153, 2019.

[7] Y. Xu and Y. Liu, "DDoS attack detection under SDN context," in *35th Annual IEEE Int. Conf. on Computer Communications*, San Francisco, CA, USA, pp. 1–9, 2016.

[8] R. T. Kokila, S. T. Selvi and K. Govindarajan, "DDoS detection and analysis in SDN-based environment using support vector machine classifier," in *Sixth Int. Conf. on Advanced Computing*, Chennai, India, pp. 205–210, 2014.

[9] H. Wang, L. Xu and G. Gu, "Floodguard: A dos attack prevention extension in software-defined networks," in *45th Annual IEEE/IFIP Int. Conf. on Dependable Systems and Networks*, Rio de Janeiro, Brazil, pp. 239–250, 2015.

[10] K. S. Sahoo, A. Iqbal, P. Maiti and B. Sahoo, *A Machine Learning Approach for Predicting DDoS Traffic in Software Defined Networks*. Bhubaneswar, India: International Conference on Information Technology, pp. 199–203, 2018.

[11] N. Meti, D. G. Narayan and V. P. Baligar, "Detection of distributed denial of service attacks using machine learning algorithms in software defined networks," in *Int. Conf. on Advances in Computing, Communications and Informatics*, Udupi, India, pp. 1366–1371, 2017.

[12] S. Shin, V. Yegneswaran, P. Porras and G. Gu, "Avant-guard: Scalable and vigilant switch flow management in software-defined networks," in *Proc. of the 2013 ACM SIGSAC Conf. on Computer & Communications Security*, New York, United States, pp. 413–424, 2013.

[13] Z. Chen, F. Jiang, Y. Cheng, X. Gu, W. Liu *et al.,* "XGBoost classifier for DDoS attack detection and analysis in SDN-based cloud," in *IEEE Int. Conf. on Big Data and Smart Computing*, Shanghai, China, pp. 251–256, 2018.

[14] M. Latah and L. Toker, "Towards an efficient anomaly-based intrusion detection for software-defined networks," *IET Networks*, vol. 7, no. 6, pp. 453–459, 2018.

[15] A. B. Dehkordi, M. Soltanaghaei and F. Z. Boroujeni, "The DDoS attacks detection through machine learning and statistical methods in SDN," *The Journal of Supercomputing*, vol. 77, no. 3, pp. 2383–2415, 2021.

[16] L. Tan, Y. Pan, J. Wu, J. Zhou, H. Jiang *et al.,* "A new framework for DDoS attack detection and defense in SDN environment," *IEEE Access*, vol. 8, pp. 161908–161919, 2020.

[17] J. Ye, X. Cheng, J. Zhu, L. Feng and L. Song, "A DDoS attack detection method based on SVM in software defined network," *Security and Communication Networks*, vol. 2018, no. 4, pp. 11–23, 2018.

[18] K. S. Sahoo, B. K. Tripathy, K. Naik, S. Ramasubbareddy, B. Balusamy *et al.,* "An evolutionary SVM model for DDOS attack detection in software defined networks," *IEEE Access*, vol. 8, pp. 132502–132513, 2020.

[19] J. A. P. Díaz, I. A. Valdovinos, K. K. R. Choo and D. Zhu, "A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning," *IEEE Access*, vol. 8, pp. 155859–155872, 2020.

[20] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras and V. Maglaris, "Combining openflow and sflow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments," *Computer Networks*, vol. 62, no. 2, pp. 122–136, 2014.

[21] L. Schehlmann and H. Baier, "COFFEE: A concept based on openflow to filter and erase events of botnet activity at high-speed nodes, information. 2013-informatik angepasst an mensch," *Organization und Umwelt*, vol. 220, pp. 2225–2239, 2013.

[22] J. Ashraf and S. Latif, "Handling intrusion and DDoS attacks in software defined networks using machine learning techniques," in *National Software Engineering Conf.*, Rawalpindi, Pakistan, pp. 55–60, 2014.

[23] R. Braga, E. Mota and A. Passito, "Lightweight DDoS flooding attack detection using NOX/openFlow," in *IEEE Local Computer Network Conf.*, Denver, USA, pp. 408–415, 2010.

[24] Y. Wang, T. Hu, G. Tang, J. Xie and J. Lu, "SGS: Safe-guard scheme for protecting control plane against DDoS attacks in software-defined networking," *IEEE Access*, vol. 7, pp. 34699–34710, 2019.

[25] W. Yassin, N. I. Udzir, Z. Muda and M. N. Sulaiman, "Anomaly-based intrusion detection through k-means clustering and naives bayes classification," in *4th Int. Conf. Computer Informatics*, Sarawak, Malaysia, vol. 49, pp. 298–303, 2013.

[26] A. Saied, R. E. Overill and T. Radzik, "Detection of known and unknown DDoS attacks using artificial neural networks," *Neurocomputing*, vol. 172, no. 7, pp. 385–393, 2016.

[27] M. Wang, Y. Lu and J. Qin, "A dynamic MLP-based DDoS attack detection method using feature selection and feedback," *Computer Security*, vol. 88, no. 7, pp. 101645–101658, 2020.

[28] M. P. Singh and A. Bhandari, "New-flow based DDoS attacks in SDN: Taxonomy, rationales, and research challenges," *Computer Communications*, vol. 154, no. 2, pp. 509–527, 2020.

[29] X. D. Zang, J. Gong and X. Y. Hu, "An adaptive profile-based approach for detecting anomalous traffic in backbone," *IEEE Access*, vol. 7, pp. 56920–56934, 2019.

[30] G. Somani, M. S. Gaur, D. Sanghi, M. Conti and R. Buyya, "DDoS attacks in cloud computing: Issues, taxonomy, and future directions," *Computer Communications*, vol. 107, no. 8, pp. 30–48, 2017.

[31] K. S. Sahoo, S. K. Panda, S. Sahoo, B. Sahoo and R. Dash, "Toward secure software-defined networks against distributed denial of service attack," *The Journal of Supercomputing*, vol. 75, no. 8, pp. 4829–4874, 2019.

[32] N. Sultana, N. Chilamkurti, W. Peng and R. Alhadad, "Survey on SDN based network intrusion detection system using machine learning approaches," *Peer-to-Peer Networking and Applications*, vol. 12, no. 2, pp. 493–501, 2019.

[33] A. Mishra, N. Gupta and B. B. Gupta, "Defense mechanisms against DDoS attack based on entropy in SDN-cloud using POX controller," *Telecommunication Systems*, vol. 77, no. 1, pp. 47–62, 2021.

[34] M. P. Novaes, L. F. Carvalho, J. Lloret and M. L. Proença, "Adversarial deep learning approach detection and defense against DDoS attacks in SDN environments," *Future Generation Computer Systems*, vol. 125, no. 1, pp. 156–167, 2021.

[35] N. Agrawal and S. Tapaswi, "An SDN-assisted defense mechduanism for the shrew DDoS Attack in a cloud computing environment," *Journal of Network and Systems Management*, vol. 29, pp. 1–28, 2021.

[36] Z. M. Hira and D. F. Gillies, "A review of feature selection and feature extraction methods applied on microarray data," *Advance Bioinformatics*, vol. 2015, no. 5, pp. 125–139, 2015.