*International Journal of Advanced Technology and Engineering Exploration, Vol 8(82)*
*ISSN (Print): 2394-5443 ISSN (Online): 2394-7454*
*http://dx.doi.org/10.19101/IJATEE.2021.874477*

Research Article

# A DDoS defence framework in software defined network using ensemble classifier with rough set theory based feature selection

**Riyad AM***
Assistant Professor, EMEA College of Arts and Science, Malappuram, Kerala, India

Received: 22-July-2021; Revised: 24-September-2021; Accepted: 26-September-2021


## Abstract
***The network traffic is getting increased day by day with the increase in the usage of the internet and related technologies such as cloud computing, Internet of Things (IoT), and big data. However, the traditional Internet Protocol (IP) based network struggles with adopting the huge network traffic through scalability, controllability as well as manageability for which software defined network has become an alternative. It meets the requirements of modern technologies in which the control is centralized over the network. Due to the increased popularity and usage, the security of the Software Defined Networking (SDN) is often compromised. Distributed Denial of Service (DDoS) attack is a major threat that suppresses the service of the SDN network. This paper focuses on providing a defence framework for SDN against DDoS attacks with two main phases. The DDoS prevention phase implemented at the data plane is responsible for preventing attacks packets through simple flow analysis. The DDoS detection phase at the control plane extracts the features from the incoming packets on which the rough set theory-based entropy is applied to select the significant features. Later ensemble classifier categorizes the flow as normal or attack. The flow rules are updated based on the obtained results. The proposed model has experimented with two publicly available datasets and the analysis are made with the obtained results. The proposed model has better precision values in predicting the flow as benign or attack with the values 96.3% and 96.12% respectively. The result analysis proves that the proposed model outperforms various other existing models in classifying DDoS attacks.***

## Keywords
***Software defined networks, Distributed denial of attack, Flow analysis, Ensemble classifier, Rough set theory, Entropy.***

## 1.Introduction
When the technology matures, the traditional networking architecture is becoming critical in handling a large number of packets which leads to the necessity of more advanced architecture. Software Defined Networking (SDN) is one such prominent architecture that addresses advanced issues such as transmission speed, manageability, higher bandwidth and virtualization through cloud computing [1]. SDN network consists of three layers such as data, control and application. The lower layer contains the switches and routers that forward the data packets to the middle layer. The control plane controls the lower layer. The SDN controller at the middle layer is the heart of the network in which it controls the device in the lower layer and the transmission are made by the device through the flow rules set by them [2]. Also, the applications at the higher layer communicate with the devices only through the controller.

The separate control entity at the control plane decides to route and leads to several advantages in managing the network through higher bandwidth and transmission speed [3]. This separation of data and control layer increases the scalability and flexibility of the network, which are necessary for recent technologies such as cloud and fog computing. The Internet of Things (IoT) and virtualization are required to meet the requirements of ever-changing technologies in our daily life [4, 5]. According to the Cisco annual report, SDN has a higher impact of about 23% on a network over the next five years and so it is considered as one of the promising technologies to automate IT [6].

Unfortunately, when the new technology arrives and if it gains popularity, then it becomes the target and becomes vulnerable to many attacks [7]. Owing to the characteristics of SDN, it is extensively utilized as a security solution for various services. However, though the centralized framework and the controlling ability seem to be effective in providing security. The

*Author for correspondence

controller, which has complete authority and control over the SDN is still vulnerable to Distributed Denial of Service (DDoS) attacks [8]. Generally, the device in the data plane verifies the input packet with the flow rules and if there is no match, then it considers it as legitimate and forwards to the control plane. Here if the attacker sends a large number of requests from various sources, then the resources will be consumed by them and the system will be unavailable to the legitimate users [9]. This makes the SDN vulnerable and thus the detection of DDoS attacks at SDN has become an important field of research [10].

Several models with a wide range varying from conventional statistical models [11], lightweight countermeasures [12] to modern machine learning [13, 14] and artificial intelligence techniques [15] exist in detecting the DDoS attack which mainly makes use of the advantage of a centralized controller at the control plane [16]. However, despite a wide range of solutions, the DDoS attack vulnerability is still increasing rapidly due to technological development among other attacks. This situation clearly states the need for proposing a promising solution for securing SDN from DDoS attacks [17]. Thus, this indispensable requirement for identifying SDN solutions against DDoS attacks is the primary motivation behind the proposed study.

Though the modern methods offer a better solution in detecting the DDoS, it also brings the enormous overhead to the controller which takes care of the entire network. This may severely affect the performance of the SDN controller at the time of heavy traffic. Also, most of the models have complex procedures in detecting the attacks which take more time to classify the results and even creates risk of service unavailability in the SDN. Yet the obtained classification accuracy in classifying the attack packets from the benign packets still needs to be improved. Subsequently, the real challenges that exist in providing a solution to the research study are reducing the overhead of the SDN controller and the overall complexity with increased accuracy in detecting attacks. Thus, the primary objective of this proposed study is to secure the SDN by detecting the DDoS attacks by reducing the overhead of the SDN controller at the control plane.

This paper presents a DDoS attack, defence framework for SDN using machine learning techniques. It is implemented in two layers with two phases respectively. The lower infrastructure layer implements the DDoS prevention phase in which it analyses the traffic flow using statistical and count based analysis for preventing DDoS attack packets. The machine learning model is implemented at the control plane for classifying the attack packets from legitimate. It extracts the features from the network traffic and selects the significant features using the Rough Set Theory (RST) based entropy approach and then classifies the attack packets using an ensemble approach. The Ensemble Approach (EA) utilizes various high standard classifiers such as Support Vector Machines (SVM), Artificial Neural Networks (ANN) and Random Forest (RF). The obtained classification results are then combined using an accuracy-based weight assignment. Finally, if the attacks are detected, then the effect of the attacks can be mitigated by dropping the packets and retrieving the allocated resources. Also, the flow rules are updated frequently by the devices at the lower layer. Various experimental analyses have been made to analyse the performance of the proposed model with two datasets that are available for public access.

The organization of the paper is as follows. Section 2 presents the various existing models from the literature concerning the proposed study. Section 3 explains the overall framework of the proposed model along with a detailed description of the working procedure of the two phases in appropriate subsections. The algorithm for the proposed model and the overall workflow is also presented. Extensive experimental analysis has been made and the performance analysis of the proposed method and the comparison with various existing models are presented in section 4. Finally, section 6 concludes the paper by mentioning the research findings and point outs the future work to be focused on.

## 2.Related works

Several solutions related to the security of the SDN from various vulnerabilities exist in the literature. Many of the works concentrate on providing models for detecting and mitigating DDoS attacks. The solutions vary from simple statistical based methods to machine learning algorithms and more complex deep learning models [18]. Machine learning algorithms are widely adopted for various solutions related to various fields. Numerous strategies related to detection and mitigation of the DDoS attack in SDN were surveyed [19–21]. Most of the models incorporate either supervised machine learning such as clustering, unsupervised machine learning such as classification or semi-supervised machine learning which is the combination of both supervised and unsupervised machine learning algorithms [22].

Entropy, a statistical approach, is considered to be the most common significant approach that measures the randomness which is then used to analyze the traffic flow. Maximum entropy estimation was suggested to identify the benign and attack traffic in the SDN network [23, 24]. Similar approaches were proposed to detect DDoS attacks using a statistically based entropy model [5, 25, 26]. A classification framework that detects the DDoS attack based on statistical features at flow, level and packet level was suggested [27]. A statistical model using entropy computation and ensemble-based machine learning technique was proposed to improve the efficiency in detecting attacks [16].

A lightweight attack detection approach was offered that makes use of tables to store header fields and their hash values to detect the attack. The model was implemented at the data plane; however, it cannot identify attacks when all the fields are changed instantaneously [28]. A similar lightweight based framework was proposed to mitigate DDoS attacks [29]. A DDoS defence framework termed ArOMA was proposed to identify the attacks automatically without any human intervention for the centralized SDN networks [30]. A mitigation framework for various security breaches by detecting the attacks on the networks integrated with SDN and the cloud that is suitable only for an IoT environment was introduced [31].

A survey of various techniques was presented and the models are categorized into four main groups based on the base idea as information theory, machine learning, Artificial Neural Networks (ANN) and other models. The author also identified various research challenges that exist in the field of study and future directions of research [32]. A clustering approach was recommended that uses agglomerative and K-means with feature extraction utilizing Principal Component Analysis (PCA) for which voting is applied to identify the class label. Once the clustering is done, then the unsupervised machine learning algorithms of K-Nearest Neighbours (KNN), SVM and RF classifiers are applied and trained for classifying the future network traffic [33].

An unsupervised model that makes use of SVM and Neural Network (NN) classifiers for categorizing the flow as legitimate and illegitimate was proposed [34]. A similar model was proposed that makes use of machine learning techniques such as KNN and XGBoost classifiers for detecting and mitigating specific DDoS attacks such as TCP-SYN and ICMP flooding. The model was evaluated based on the testbed deployment [35]. A similar model with SVM, ANN, KNN and Naive Bayes (NB) classifiers was made in which the KNN model has better results, however, the accuracy is low compared with the other models [1].

A hybrid flow-based framework was introduced that utilizes a combination of SVMs-Self-Organizing Map (SOM), in detecting DDoS attacks. The model also takes care of various network components from resource enervation in SDN [36]. A novel model was also suggested that utilizes the concept of a NB classifier on intrusion detection systems which are implemented in the form of multi-agents in the network to listen to the traffic and to classify irregular traffic from the normal one [37]. Another mechanism termed Learning Driven Detection Mitigation (LEDEM) for detecting DDoS was proposed that utilizes a semi-supervised machine learning method [38]. A secure self-adaptive framework was anticipated that extracts the network traffic attributes based on statistics and applies machine learning algorithms for DDoS attacks [39].

From the literature survey made, it is found that many solutions make use of several machine learning algorithms and statistical approaches. Though the models protect the SDN networks with some efficient solutions, the accuracy of the models is still needed to be revised. Some of the models, create a huge overhead for the centralized SDN controller. By analyzing the existing solutions with their drawbacks and the need to overcome the issues, this paper proposes the defence framework in the software defined network for DDoS attacks using an ensemble classifier with an RST based feature selection.

## 3.Proposed DDoS defence framework in SDN

The overall framework of the proposed DDoS defence framework using an ensemble classifier with an RST based feature selection is shown in *Figure 1*. The model is designed for software defined networks in which it has two modules. The first phase is the DDoS prevention implemented in the lower-level infrastructure layer specifically, the data plane in which the simple flow analysis of the input packets is used for identifying the attacks. It also prevents the attack from entering the control layer of the target system. The second phase is the DDoS attack detection implemented in the control layer where a packet level analysis is carried out in which the features of the network traffic will be extracted.

As the network traffic contains a large number of attributes in which all of them may not be important, RST based feature selection has been employed to select the significant attributes that increase the classification accuracy. Then the ensemble classifier comprising of SVM, ANN and RF are applied over the selected feature set to classify the traffic as an attack or legitimate. Finally, based on the classification result, necessary action can be carried out to mitigate the attack.

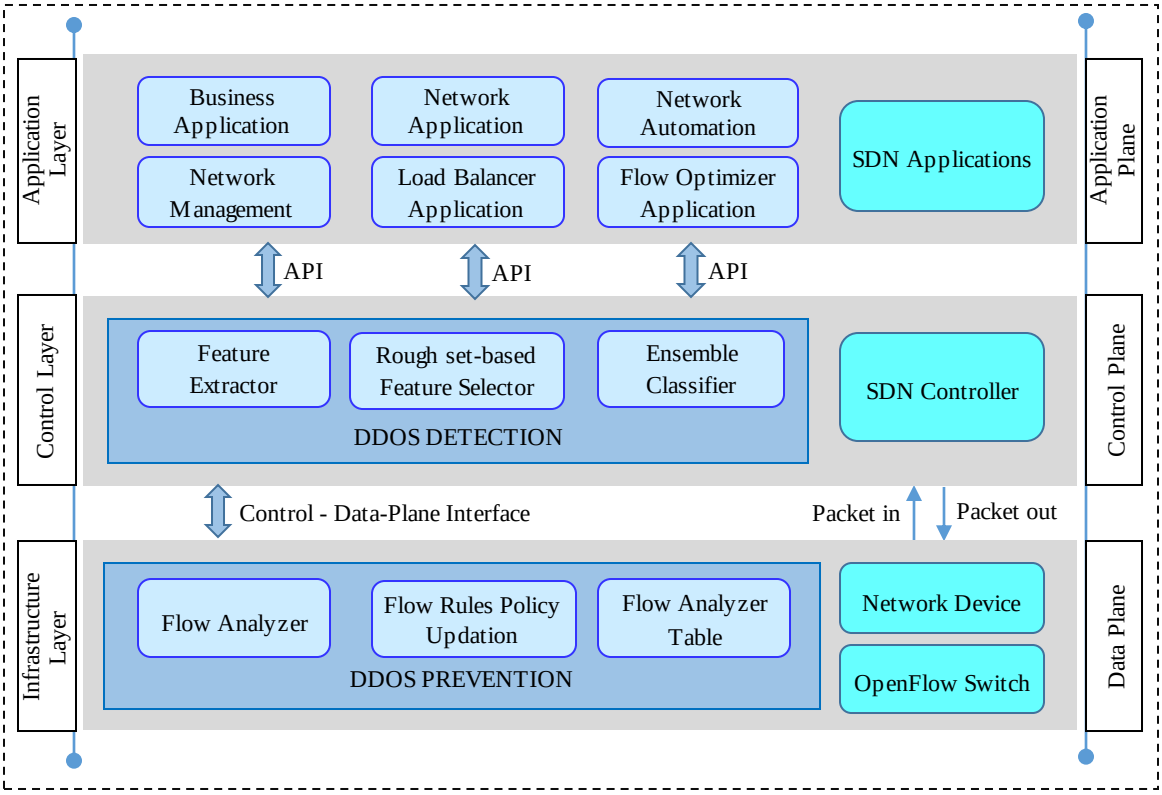


**Figure 1** Overall Design of the proposed DDoS defence framework for SDN

### 3.1DDoS attack prevention phase

In this phase, flow analysis has been carried out to find whether the traffic flow satisfies the given constraints. Initially, the incoming traffic flow is compared with the flow rules that contain the details about the flow that can be considered as normal or attack which is created from traffic flow history. However, the flow rules are updated frequently based on the flow analysis.

If the given traffic flow is not satisfied with the given constraints for the particular time window, then the traffic is updated as suspicious and are updated in the flow rules. Each incoming traffic flow is recorded in the flow table from which the analysis of the traffic flow can be made. The header fields of each packet are extracted and are stored in the flow analyzer table which can be defined as a tuple with 6 elements such a {src_ip, dest_ip, src_port, dest_port, protocol and bytes}.

#### 3.1.1Flow analyser

The flow analyzer is responsible for performing the statistical analysis on the traffic flow for the particular time window. Flow analyzer extracts the details from the OpenFlow switches about the number of bytes ($BC_{t_n}$) and the number of packets ($PC_{t_n}$) at a time $t_n$ and connection duration *(d)* of each specific flow [40].

**Byte count at time $t_n$ ($B_{t_n}$)**

It depicts the average byte rate of a flow between a time frame represented as $t_n$ and $t_{n-1}$. The formula to compute $B_{t_n}$ is given in Equation 1.

$$B_{t_n} = \frac{BC_{t_n} - BC_{t_{n-1}}}{t_n - t_{n-1}} \quad (1)$$

**Packet count at time $t_n$ ($P_{t_n}$)**

It depicts the average packet rate of a flow between a time frame represented as $t_n$ and $t_{n-1}$. The formula to compute $P_{t_n}$ is given in Equation 2.

$$P_{t_n} = \frac{PC_{t_n} - PC_{t_{n-1}}}{t_n - t_{n-1}} \quad (2)$$

**Connection duration ($d_f$)**

It defines the duration of a connection of the traffic flow which can be computed by subtracting the relative end time with the relative start time and the formula is given in Equation 3.

$$d_f = time_{con_{end}} - time_{con_{start}} \quad (3)$$

These values are actual flow parameter values. The vector of the extracted actual value of the volume-based metrics $av(B_{t_n}, P_{t_n}, d_f)$ is then compared with the predicted value $pv(B_{t_n}, P_{t_n}, d_f)$ which is computed from the previous actual traffic flow using Exponential Moving Average (EMA) as given in Equation 4.

$$pv_{t_n} = av_{t_{n-1}} \times \alpha + pv_{t_{n-2}} \times (\alpha - 1), \alpha = {}^{2}/_{n} \quad (4)$$

Here the value 2 refers to the smoothing factor and the EMA instead of a weighted moving average is utilized since it represents the value based on the recent traffic flow that occurred in the network. However, to identify the difference between the actual and the predicted values, the confidence interval that specifies the upper and lower bounds is computed using the z score as in Equation 5.

$$confi\_interval = \begin{cases} \overline{av} - 1.96 \times \left(\frac{\sigma_{av}}{\sqrt{n}}\right), \ for\ LB \\ \overline{av} + 1.96 \times \left(\frac{\sigma_{av}}{\sqrt{n}}\right), \ for\ UB \end{cases} \quad (5)$$

Here, $\overline{av}$ represent the mean of all the $n$ records of the vector $av$ containing actual values, 1.96 represents the z-score value at 95% confidence level, $\sigma_{av}$ represent the standard deviation of the vector $av$.

Then the comparison will be made for the predicted vector through EMA with the lower and upper bounds of the actual values. Thus, if none of the values in the predicted vector lies between the lower and upper bounds, then it specifies that the flow is an attack in which case, the packets will be discarded and are not forwarded to the SDN controller and is updated in the flow rule policy. On the other hand, if more than one element in the predicted vector does not fit in the confidence interval, then the flow is marked as suspicious in the flow rule policy and will be forwarded to the SDN controller with an alert. Otherwise, the traffic flow is considered to be legitimate and is then forwarded to the SDN controller. The algorithm steps for the DDoS attack prevention phase using flow analysis are presented below.

**Algorithm1: FlowAnalysis_Prevention**
**Input:** Input traffic flow
**Output:** Identifying attacks and normal packets
**Procedure** flow_analyzer(traffic flow)
**Begin**
Extract features from incoming traffic flow
Store the traffic flow details in the flow_analyzer table
**For** each specific time window between $t_{n-1}$ and $t_n$
//Compute actual analysis parameter values
Compute the actual parameter values $av$ as $av(BC_{t_n}, PC_{t_n}, d)$ as in Equation 1, 2 and 3
**For** each previous actual parameter value (APV) ranges $t_1$ to $t_{n-1}$
Compute EMA
**End For**
//Compute predicted analysis parameter values
Compute predicted $pv(B_{t_n}, P_{t_n}, d_f)$ as in Equation 4
//Evaluate the lower and upper bound
Identify Lower Bound (LB) and Upper Bound (UB) from previous $av$s using Z-score confidence interval as in Equation 5
**If** $pv$ lies between LB and UB **then**
Drop the packets and update the flow rules
**Elseif** any two elements in the $pv$ do lie between LB and UB
Mark as suspicious and forward to a controller to next phase
**Else**
Mark as legitimate and forward for processing the request
**End If**
**End For**
**End Procedure**

### 3.2DDoS attack detection phase

This phase is responsible for identifying the DDoS attack that is identified as suspicious and other attack flows that are missed by the prevention phase. Thus, the prevention phase will reduce the workload of the control plane by preventing most of the attacks at an early stage. This phase extracts the features from the network traffic, identifies the significant features and finally classifies the traffic using a trained ensemble classifier.

### 3.2.1Feature extractor

This component helps in extracting various attributes from the network traffic such as count based [41] and statistical based analysis [42, 43]. In count-based analysis, the count of incoming packets is analysed for each specific time frame. If the number of packets from the same source is greater than the threshold value, then the packets are considered as attack packets else the number of packets from different sources is analysed for the particular time frame. If the packets from the number of different sources exceed the threshold value, then the packets are recognised as attack packets that can be dropped. The threshold values for the analysis are used as given by [41].

The statistical based packet analysis based on the ratio of various protocols and their entropy is computed as given below.

**Protocol proportion**

The packets are initially classified based on the protocols. Then the ratio of each protocol is computed for the given time frame as the ratio of the number of packets of a specific protocol to the total number of packets.

**Entropy computation**

The entropy of various fields in the tuple is computed for the analysis. The entropy value of the particular source IP *(sip)* can be computed using information entropy as in Equation 6.

$$E(sip_n) = \sum_{i=1}^{k} -p(sip_n^i)\log_2 p(sip_n^i) \quad (6)$$

Here k is the number of IP addresses from different sources. The equation can be defined with the number of packets with the specific source address ($n(sip)$) to the total number of packets ($p_n$) at a particular time frame $n$ can be computed as in Equation 7.

$$E(sip_n) = \sum_{i=1}^{k} -\frac{n(sip)_n^i}{p_n}\log_2\frac{n(sip)_n^i}{p_n} \quad (7)$$

This entropy computation can also be extended to other elements *dest_ip*, *src_port*, *dest_port*, *protocol* as *E(dip)*, *E(sport)*, *E(dport)*, and for different protocols such as Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Internet Control Message Protocol (ICMP) as $E(p_{tcp})$, $E(p_{udp})$, $E(p_{icmp})$. Here, if the network receives any attacks, then the number of the source IP address will increase, that also leads to the increase in the entropy values. Thus, the maximum entropy for normal traffic can be computed and the results can be compared with the current entropy values. Similarly, the entropy of protocols can be computed which may lead to zero during the attack traffic. The statistical analysis of the network traffic can be highly influential in detecting the attack packets.

If the traffic flow is legitimate at both count and statistical based analysis, then the flow is legitimate for which the request can be processed. If the analysed traffic flow is recognized as an attack by both analyses, then it is an attack for which the packets can be dropped and the resources can be revoked. On the other hand, if one analysis identifies the traffic as an attack and the other identifies it as normal, the computed values are considered as extracted features along with other information and are used for classifying the traffic as normal and attack.

### 3.2.2RST based entropy for feature selection

The rough set theory is a significant field that mainly evaluates the dependency between various data. Thus, it is used in data mining for knowledge extraction as well as for classification problems. Thus, combining the RST with information entropy provides higher accuracy in classification as well as in identifying the significant features. In the proposed work, an RST based entropy approach is used for selecting the important features related to the study [44].

Initially, the most important CORE attributes are identified by evaluating the probability of a positive region concerning the target feature (T). With the selected attributes, the dependency of other attributes with that of the CORE attribute concerning the target feature is made with the help of a heuristic based evaluation criterion as in Equation 8.

$$F(S,a) = Card\left(POS_{S+\{a\}}(D)\right) \times E(S,a) \quad (8)$$

Here, *Card()* represents the cardinality of the positive region and *E(S, a)* represents the entropy concerning the attribute $a$. The candidate feature having higher dependencies concerning the CORE attributes are selected further. This process continues until certain stopping condition $POS_{S\subseteq}(D)=POS_C(D)$ is met.

### 3.2.3Ensemble classifier

Ensemble classifiers have gained attention in machine learning due to their improved accuracy for various types of datasets. Ensemble models classify the test samples based on various base classifiers instead of depending on a single classifier. In the proposed model, the most powerful classifiers such as SVM, ANN and RF are used for evaluation.

Here the final classification result is evaluated by providing various weights to the classifiers based on their accuracy. If the classifiers have the same result, then the classification is done with a single result, whereas if they disagree with the results, the highest of the computed weight for the classification result will be considered. During the training phase, the classification accuracy of the classifiers is evaluated and are considered as the weights of the classifiers *w(c)*. Then the final result can be obtained by computing the class weight from the resultant label of different classifiers as in Equation 9.

$$Class\_weight_i = \frac{\sum w(c)}{n} \quad (9)$$

Here n is the number of classifiers that predicts the test data with the same class label. Once the traffic flow is considered as an attack, then the details are updated in the flow rules. Additionally, the packets are dropped and the allocated resources are also retrieved.

The algorithm for the attack detection phase using the machine learning technique is presented below.

**Algorithm2: ML_attack_detection**
**Input:** Suspicious traffic flow
**Output:** Identifying attacks and normal packets
**Procedure** attack_detector (traffic flow)
**Begin**
**For** each specific time window between $t_{n-1}$ and $t_n$
//Count based analysis
Compute the number of packets from the same source and different sources
**If** the number of packets from the same source < threshold **then**
**If** the number of packets from different sources < threshold **then**
Mark as legitimate
**Else** mark as an attack
**End If**
**Else** mark as an attack
//Statistical based analysis
**If** the ratio of incoming & outgoing packets is normal **then**
**If** E(sip), *E(dip)*, *E(sport)*, *E(dport)* is similar to the entropy of normal packets **then**
**If** $E(p_{tcp})$, $E(p_{udp})$, $E(p_{icmp})$ is not zero **then**
Mark as legitimate
**Else** mark as an attack
**End If**
**Else** mark as an attack
**End If**
**Else** mark as an attack
**End IF**
**End For**
**If** traffic flow is legitimate in count and statistical based analysis **then**
Process the request
**Else if** traffic flow seems to be an attack on the count and statistical based analysis
Drop the packets as an attack
**Else** extract the features
**End If**
//Feature selection using RST based entropy
Identify CORE attrib. using probability concerning class
**For** each attribute other than the CORE attribute
Evaluate dependency of other attributes with CORE attributes using heuristic based evaluation criterion
Select the candidate having higher dependency
Continue until the stop criterion is satisfied
**End For**
//Classification of normal and attacks
Apply SVM, ANN and RF for selected features
Evaluate classification accuracy as weights for classifiers
Compute the class weight as in Eq. (9)
Predict results based on the class label having the highest weight.
**If** an attack is detected **then**
Update flow rules and revoke resources allocated
**End If**
**End Procedure**

## 4.Experimental analysis

This section presents the experimental analysis and the summary of results obtained for the proposed model.

### 4.1Experimental setup

Experiments are done on the system configured with i7 3.4Ghz with 8GB RAM and 64-bit Windows 8.1 operating system. The programs for implementing the RST based feature selection and ensemble classifiers are written in Java and are executed for the different datasets. Additionally, a simple simulation has been made to identify the feasibility of the study in which the host system is connected to the outside world with network of systems with 15 clients using a router and firewall [45]. The clients send both legitimate and attack requests to the host system. The attack packets specifically smurf and TCP-Synchronize (SYN) attacks are generated using the Netwag tool [46] with which the packets are examined and the features are extracted. Then the proposed RST based selection and ensemble classifiers are applied after the data to identify the

attacks. The experimental investigation has been performed with 200 packets in which 20 packets are attacks.

The workflow of the proposed overall DDoS attack, defence model with the prevention and detection phase is presented in *Figure 2*.

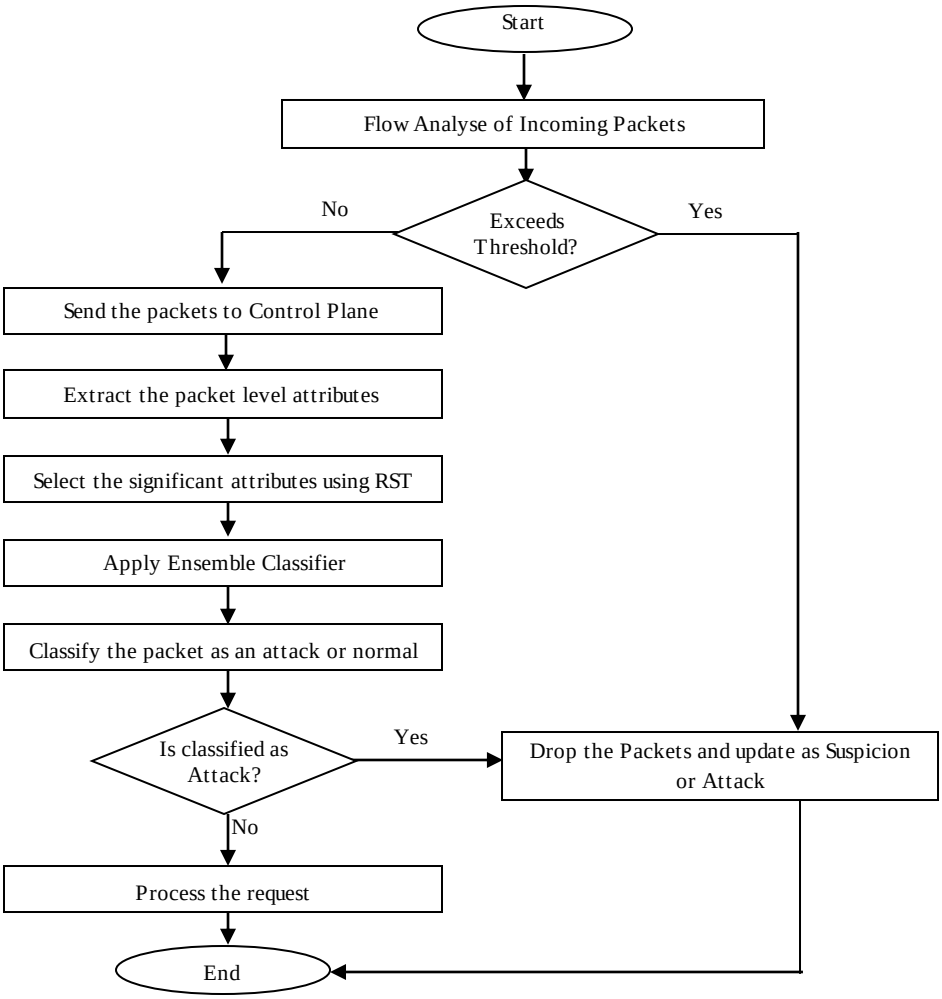


**Figure 2** Workflow of the proposed DDoS defence model

### 4.2Dataset used

The proposed classifier is trained using two datasets such as the NSL dataset and the UNB-ISCX dataset for evaluation. The NSL dataset [47] has 41 attributes that are extracted from the network traffic and the target attribute has 5 classes comprising benign traffic and attack of various types including DoS, Probe, Remote to Local (R2L), and User to Root (U2R). For the evaluation, about 50% of the instances is used as training samples and the remaining 50% of the instances as a test set. Another UNB ISCX dataset [37] was used to evaluate the performance of the proposed models in detecting attacks. It contains the set of various labelled data that indicates the DDoS attack and normal data arrived in the network traffic. The dataset is partitioned and 75,248 samples are used as training sets. A detailed description of the dataset including the number of features, the number of classes and the number of samples used in the proposed study is given in *Table 1*.

**Table 1** Dataset description

| Dataset | No. of features | No. of samples | No. of classes |
|---|---|---|---|
| NSL | 41 | Train data: 1,25,973<br>Test data: 20064 | 5 (4 attacks and 1 normal) |
| UNB ISCX | 21 | Total: 2,25,745<br>Training: 75,248 | 2 (1 attack and 1 normal) |

With the experiments performed with 200 packets, a dataset has been generated for the proposed study that contains 200 samples containing two classes identified as an attack and normal. Here 25 features are extracted and are listed in *Table 2*. The features are extracted by collecting basic details of the packets (flow_id, src_ip, dest_ip, src_port, dest_port, protocol, service, duration, type) as well as by performing entropy calculations (E(sip), E(dip), E(sport), E(dport), E($p_{tcp}$), E($p_{udp}$), E($p_{icmp}$)) as discussed in Section 3.2.1 and other features are collected based on the inspiration from NSL dataset.

**Table 2** List of features in the generated dataset

| Feature | Description |
|---|---|
| flow_id | Packet flow id |
| src_ip | Source address |
| dest_ip | Destination address |
| src_port | Source port |
| dest_port | Destination port |
| Protocol | Protocol used |
| Duration | Duration of connection |
| E(sip) | Entropy of source address in past 2s |
| E(dip) | Entropy of destination address in past 2s |
| E(sport) | Entropy of source port in past 2s |
| E(dport) | Entropy of destination port in past 2s |
| E($p_{tcp}$) | Entropy of TCP protocol in past 2s |
| E($p_{udp}$) | Entropy of UDP protocol in past 2s |
| E($p_{icmp}$) | Entropy of ICMP protocol in past 2s |
| Service | Type of service |
| Count | Sum of connection to the same system in past 2s |
| Type | Type of data |
| src_byte | Total bytes from source to destination |
| dest_byte | Total bytes from destination to source |
| same_srv_rate | Total connection to the same service |
| diff_srv_rate | Total connection to different service |
| service_count | Total connection to the service in past 2s |
| dst_host_count | Total connection to the same host |
| dst_host_srv_count | Total connection to the same host and same service |
| srv_diff_host_count | Total of different connections to host |

## 4.3Result analysis

A handful of performance metrics is available to evaluate the proposed model and to compare the results with the existing models [47]. This section evaluates the performance of the proposed model using various key evaluation metrics.

### 4.3.1Analysis using NSL dataset

To evaluate the performance of the proposed feature selection approach, the approach is evaluated with various standards and existing feature selection techniques such as genetic, ranker and greedy algorithms and best subset selection algorithm [39] using the NSL dataset. The selected features with various feature selection techniques used for the analysis of the NSL dataset are listed in *Table 3*.

**Table 3** List of selected features with different techniques using NSL dataset

| FS techniques | Selected features | Count |
|---|---|---|
| Genetic | 3, 4, 5, 6, 7, 9, 12, 24, 25, 26, 27, 29, 30, 32, 34, 39 | 16 |
| Ranker | 3, 4, 5, 6, 12, 23, 25, 26, 28, 29, 30, 31, 33, 34, 35, 36, 38, 39 | 17 |
| Greedy | 3, 4, 5, 6, 12, 14, 26, 29, 30, 37, 38 | 11 |
| Best Subset Selection | 3, 4, 5, 6, 7, 9, 12, 14, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39 | 25 |
| RST based Entropy | 3, 4, 5, 6, 7, 9, 12, 14, 23, 25, 26, 27, 29, 30, 32, 34, 36, 38 | 18 |

The feature selection techniques such as genetic, ranker and greedy algorithms and best subset selection, then compare with the proposed model with respect to the detection rate with NSL dataset using various classifiers such as SVM, J48 and NB classifiers. Here, genetic algorithm and greedy algorithm are evaluated with Correlation-based Feature Selection (CFS) which selects the attributes based on the correlation between them. The ranker algorithm is evaluated using the infogain that selects significant features based on the information gain concerning the target feature. The methods are analysed using detection rate, where it represents the number of attack samples identified among the total samples. The detection rate of the proposed RST based feature selection, and existing feature selection methods such as genetic, ranker, greedy, best subset selection methods used in the analysis are presented in *Table 4*.

From the evaluated detection rate, the proposed RST based entropy approach offers a good result with SVM classifier as 99.51% and NB classifiers as 96.82% among other models which have been used for comparison. On the other hand, J48 still provides good results with a 99.72% of detection rate. With the J48 classifier, the existing best feature selection

model offers better results of 99.75% than other models.

**Table 4** Detection accuracy of features with different techniques

| S. No. | Detection rate | | |
|---|---|---|---|
| | SVM | J48 | Naïve Bayes |
| All features | 94.31 | 95.76 | 80.12 |
| Genetic Alg. + CFS | 96.85 | 98.33 | 82.62 |
| Ranker Alg.+InfoGain | 97.14 | 97.77 | 82.03 |
| Greedy Alg. +CFS | 95.71 | 92.74 | 83.92 |
| Best Subset Selection | 99.40 | 99.75 | 95.87 |
| RST based Entropy | 99.51 | 99.72 | 96.82 |

The average increased rate in attack detection is evaluated for the proposed model which is computed using $\Delta d/D_R$, where $D_R$ is the detection rate of the proposed model and $\Delta d$ is the difference in the detection rate with other models. The average increase in attack detection for the proposed model with SVM, J48 and NB classifiers are 2.24%, 2.58% and 11.06% respectively, and that of the best subset selection is 2.17%, 2.60% and 10.43% respectively.

The values obtained for the detection rate presented in *Table 4* is depicted as a graph in *Figure 3* in which the horizontal axis represents the different feature selection algorithms with different classifiers and the vertical axis represents the detection rate in percentage. The proposed RST based entropy model offers best performance similar to the subset selection model than other models. This is because the RST based feature selection accepts the advantage of both RST and information entropy of evaluating the dependency between the attributes. Thus, though the results obtained for the proposed model and best feature selection seems to be similar, however, the proposed method offers good results in two out of three cases.

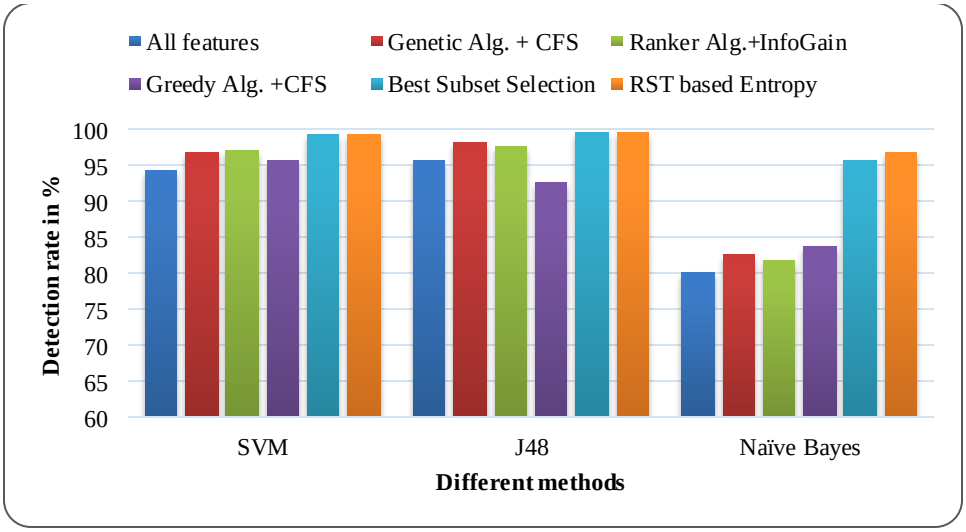


**Figure 3** Detection rate comparison with different feature selection techniques

The proposed model has been evaluated using different performance metrics such as accuracy, detection rate, the precision, false alarm rate, and F1-measure for different class samples specified in the NSL dataset with 10-fold cross-validation. The evaluation results obtained for the proposed model for the NSL dataset with different classes are presented in *Table 5*. The table also presents the weighted average computed for all the classes in the NSL dataset with respect to all the metrics used for the analysis. The attack class having good results for different performance metrics are highlighted in bold letters.

Among the different classes in the NSL dataset such as normal and attack classes, including Probe, DoS, U2R, R2L, the proposed model offers better accuracy, detection rate and precision for probe attack class as 99.91%, 99.87%, 99.85% respectively and the model offers a minimum false rate of 0.01% for U2R and R2L attack classes. The weighted average of the performance metrics which have been used in the study, including accuracy, detection rate, the precision, false alarm rate and F1-measure for the proposed model is 99.58%, 99.57%, 97.89%, 0.18% and 98.71% respectively.

To evaluate the results, the accuracy of the proposed model with ensemble classifiers and other individual classifiers such as SVM, ANN and RF are analyzed and the accuracy of the classifiers after selecting the significant features using the proposed RST based feature selection are presented in *Table 6*. The obtained results show that the proposed model has a better accuracy rate for most of the attack classes and the normal class among other individual classifiers used.

**Table 5** Results for proposed DDoS attack detection model using NSL dataset

| Attack Class | Accuracy | Detection rate | Precision | False alarm rate | F1-mesure |
|---|---|---|---|---|---|
| Normal | 99.23 | 99.82 | 98.98 | 0.50 | 99.40 |
| Probe | **99.91** | **99.87** | **99.85** | 0.05 | **99.86** |
| DoS | 99.67 | 99.76 | 97.52 | 0.10 | 98.63 |
| U2R | 99.66 | 87.63 | 90.14 | **0.01** | 88.87 |
| R2L | 99.72 | 90.11 | 99.67 | **0.01** | 94.65 |
| *Wt. Avg. value* | *99.58* | *99.57* | *97.89* | *0.18* | *98.71* |

**Table 6** Accuracy comparison on different classifiers using NSL dataset

| Attack class | Accuracy | | | |
|---|---|---|---|---|
| | SVM | ANN | RF | Ensemble |
| Normal | 99.03 | 87.32 | 93.16 | 99.23 |
| Probe | 90.91 | 99.85 | 95.76 | 99.91 |
| DoS | 92.36 | 98.63 | 89.74 | 99.67 |
| U2R | 99.71 | 72.32 | 98.64 | 99.66 |
| R2L | 65.77 | 97.21 | 96.71 | 99.72 |

The accuracy rate of various classifiers is represented as a graph in *Figure 4*. In the graph, the horizontal axis represents the classifiers implemented on different classes in the NSL dataset whereas the vertical axis represents the accuracy rate of the classifiers for different classes. From the graph, it is clear that the accuracy of individual classifiers such as SVM and ANN and RF is highly influenced by the attack classes clandestinely. However, the proposed model is not subjective to any of the classes and shows a steady rate in classifying attacks from the normal one.

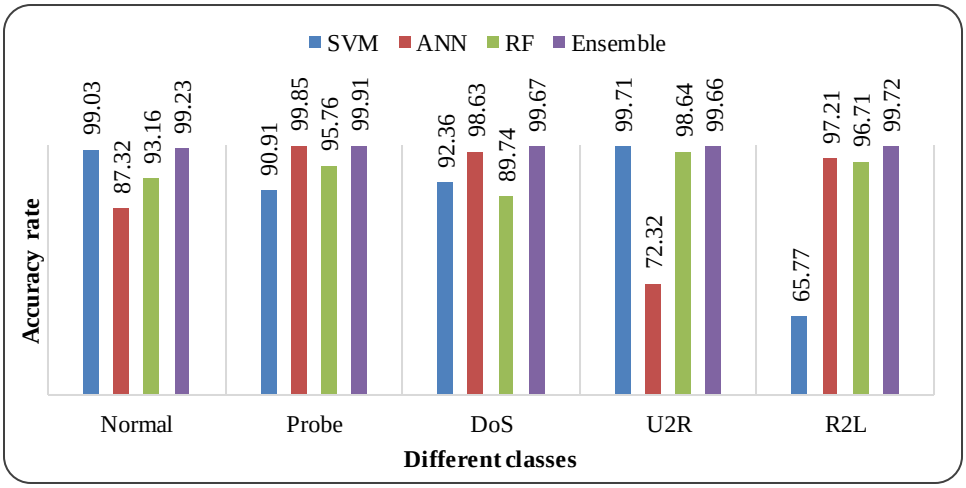


**Figure 4** Accuracy comparison of different classifiers using NSL dataset

### 4.3.2Analysis using UNB-ISCX dataset

A performance analysis is also made using the UNB-ISCX dataset for a fair comparison of the proposed model with the existing models that were evaluated using the same dataset in the literature. The evaluation metrics used by the existing works [38] such as F-measure (F), Recall of Benign Detection (RBD), the Precision of Benign Detection (PBD), Recall of DDoS Detection (RDD), Precision of DDoS Detection (PDD), Network Accuracy (NA), Average Detection Time (ADT) measured in milliseconds (ms) and Average Throughput (AT) are utilized in this study. The existing models used for the evaluation of performance comparison are Deep

Belief Network (DBN) [31], Extreme Learning Machines (ELM), NB [37] and Semi-supervised Deep Extreme Learning Machine (SDLEM) [38]. The assessment is carried out with 75,248 samples for training the model to which the 10-fold cross-validation is applied to evaluate the performance. The results obtained for the proposed model and the existing models are presented in *Table 7* where the model having best results is marked as bold.

**Table 7** Performance comparison using UNB-ISCX dataset

| Metrics | DBN | ELM | Naïve Bayes | SDLEM | Proposed |
|---|---|---|---|---|---|
| F-Measure | 96.2 | 86.33 | 88.92 | 92.92 | **97.11** |
| RBD | **97.38** | 84.95 | 87.80 | 91.9 | 97.21 |
| PBD | 95.35 | 91.63 | 92.67 | 94.69 | **96.3** |
| RDD | 97.27 | 82.71 | 86.29 | 91.12 | **97.69** |
| PDD | 95.16 | 90.27 | 91.71 | 94.15 | **96.12** |
| NA | **96.28** | 87.31 | 89.58 | 92.96 | 94.5 |
| ADT | 4.8 | 3.2 | **2.1** | 2.3 | 2.4 |
| AT | 145 | 151 | 138 | 175 | **163** |

The proposed model has better precision values in predicting benign and DDoS attack detection with 96.3% and 96.12%, respectively whereas the recall values in predicting benign and DDoS attack detection are 97.21% and 97.69% respectively. Though the proposed model has a minimum network accuracy and maximum average detection time as 94.5% and 2.4 ms respectively, than the DBN model that has better results, the difference between the values is very minimum. Thus, from the analysis made on the computed results, it can be seen that the proposed model and DBN show good results than ELM, NB and SDLEM. Specifically, SDLEM maintains good recall for benign detection and network accuracy. On the other hand, the proposed model offers better precision for benign detection, recall and precision for DDoS detection. This shows that the model works well in detecting attacks. This is because the ensemble model uses three main powerful classifiers such as SVM, ANN and RF which is suitable for large datasets with high dimensional space with the increased accuracy which is not affected by overfitting. The proposed ensemble model increases its merits such as making decisions by analyzing similar events and producing the appropriate result, even with the incomplete data from the models used in the ensemble classifier. The recall and precision for benign and DDoS attack detection along with the network accuracy of different methods such as DBN, ELM, NB, SDLEM and proposed model are compared using a graph representation and is presented in *Figure 5*.

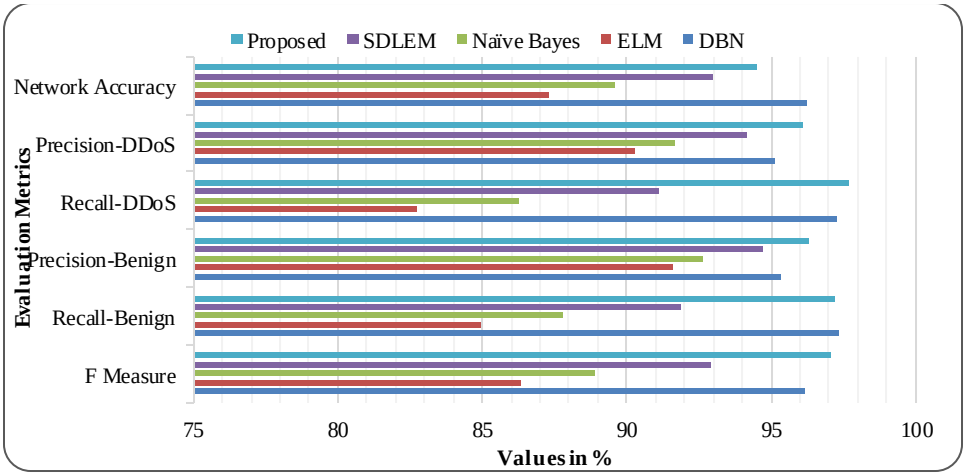


**Figure 5** Performance comparison with UNB-ISCX dataset

When analyzing the attack detection time and throughput, NB performs attacks detection in less time with 2.1ms, SDLEM model has a better throughput of 175, yet the detection results for them are still inaccurate. Thus, the speed of the proposed model and the average throughput are optimum with a good detection rate for the proposed model and even the execution time of the proposed model is half of the execution time of the DBN model. Thus, on the whole, the method offers optimal results in all aspects.

#### 4.3.3Analysis using generated dataset

The general performance metrics such as execution time, accuracy, precision, recall, F1-score are used to evaluate the proposed model using the real time generated datasets. The results of the proposed EA are compared with the existing standard models as mentioned in [35] such as KNN, Decision Tree (DT), NN, SVM, ANN, and RF. The acquired results from different performance metrics are presented in *Table 8*. Initially, the statistical flow analysis is carried out to find whether the traffic flow satisfies the given constraints as described in section 3.1. In the experimental analysis, among 20 attack samples, 12 samples were identified as attacks in the first phase. Then the remaining samples are passed to the second phase that utilizes RST based feature section model and the proposed ensemble model. In the analysis, before performing the classification, the RST based feature selection is applied over the generated dataset according to which 17 features are selected. These selected features with 200 samples are then used to train the models employed in the evaluation study by applying 10-fold cross-validation. The classifiers having better results are represented in bold letters.

The classification time for the proposed model is 3.5ms which is higher than the KNN and DT models having 0.4ms approximately. In comparison with the precision, the KNN classifier has an increased precision of 99.03% and that of recall, the RF classifier offers a maximum of 98.57%. However, with respect to the accuracy and F-score, the proposed model offers better results of 98.89% and 98.45% respectively.

Thus, with the overall analysis, for each metric used in the evaluation, the proposed model acquires at the top three positions and also the average values of the proposed model are higher than many of the existing models used for the comparison. The average rank is also computed for all the models by converting the values into scores and calculating the average for the obtained ranks. Here, the average rank of the proposed model is 2.2, the random forest has 2.6, KNN and DT has 3.8, ANN has 5, NN has 5.2 and SVM acquires 5.4.

This analysis shows that the proposed model outperforms other models with respect to different metrics, yet the overall performance is also better than the other classifiers under comparison. The values in *Table 8* are represented as a graph in *Figure 6*.

From the above analysis obtained with the three datasets, it is clear that the proposed model outperforms various other models in detecting the DDoS attack specific in the SDN network. Though the execution time and throughput seem to need an improvement, the accuracy of the results produced and attack detection rate is far better than many classifiers used in the study. But the limitation of the proposed model is that the model has been tested with limited samples generated by the real time simulation. Complete list of abbreviations is shown in *Appendix I*.

**Table 8** Performance comparison using generated dataset

| **Metrics** | **KNN** | **DT** | **NN** | **SVM** | **ANN** | **RF** | **EA** |
|---|---|---|---|---|---|---|---|
| Time (ms) | 0.411 | **0.405** | 14.325 | 5.321 | 8.256 | 7.632 | 3.512 |
| Accuracy | 98.21 | 98.15 | 98.84 | 98.25 | 98.48 | 98.72 | **98.89** |
| Precision | **99.03** | 97.67 | 96.69 | 97.25 | 97.91 | 98.21 | 98.69 |
| Recall | 97.14 | 98.44 | 98.32 | 98.12 | 97.85 | **98.57** | 98.23 |
| F1-score | 98.08 | 98.05 | 97.50 | 97.68 | 97.88 | 98.39 | **98.45** |
| *Avg. Rank* | *3.8* | *3.8* | *5.2* | *5.4* | *5* | *2.6* | ***2.2*** |

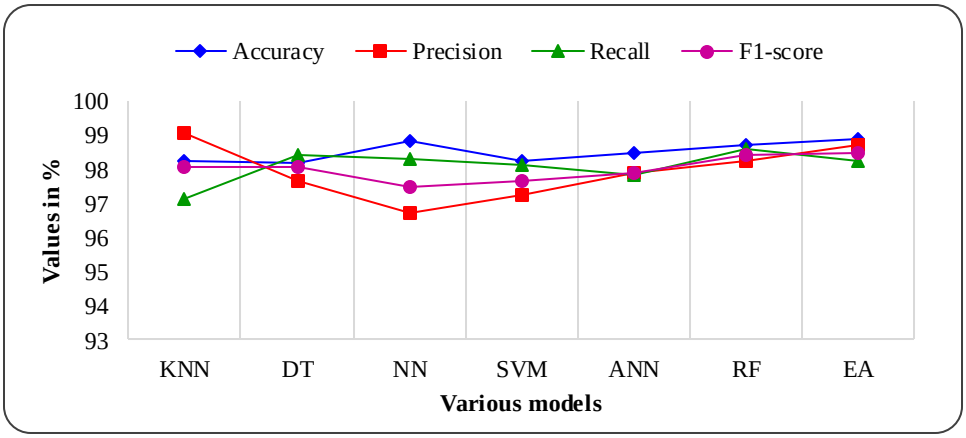


**Figure 6** Performance comparison with generated dataset

## 5.Conclusion and future work

The paper presents the DDoS attack, defence framework specifically for software defined networks. The model is divided into two phases in which one phase is implemented in the infrastructure layer and the second phase is implemented at the control layer. The attack prevention phase in the infrastructure layer performs flow analysis from time to time and identifies the attacks and prevents them from entering the control layer. The attack detection phase applies machine learning to classify the attacks. It extracts the features from the incoming traffic and applies rough set theory-based entropy for selecting important features for the study. Later a trained ensemble classifier classifies the data as normal or attacks which are then managed by dropping the request and updating the flow rules. Various experimental analysis has been performed with two different datasets to analyse the performance of the model. From the results, the proposed model has better average accuracy, the detection rate and false alarm rate at 99.58%, 99.57% and 0.18% respectively. Also, the ensemble classifier takes a minimum time to classify the traffic than many other algorithms. The future work focuses on implementing the model in real-time and analyze the performance using other metrics such as CPU utilization and other overheads. Also, future work tries to enhance the model performance in terms of execution time.

## Acknowledgment

None.

## Conflicts of interest

The authors have no conflicts of interest to declare.

## References

[1] Polat H, Polat O, Cetin A. Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models. Sustainability. 2020; 12(3):1-16.

[2] Kreutz D, Ramos FM, Verissimo PE, Rothenberg CE, Azodolmolky S, Uhlig S. Software-defined networking: a comprehensive survey. Proceedings of the IEEE. 2014; 103(1):14-76.

[3] Sahoo KS, Puthal D, Obaidat MS, Sarkar A, Mishra SK, Sahoo B. On the placement of controllers in software-defined-WAN using meta-heuristic approach. Journal of Systems and Software. 2018; 145:180-94.

[4] Yin D, Zhang L, Yang K. A DDoS attack detection and mitigation with software-defined internet of things framework. IEEE Access. 2018; 6:24694-705.

[5] Ujjan RM, Pervez Z, Dahal K, Khan WA, Khattak AM, Hayat B. Entropy based features distribution for anti-DDoS model in SDN. Sustainability. 2021; 13(3):1-27.

[6] https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html. Accessed 17 September 2021.

[7] Saravanan A, Ahmed MI, Bama SS. Automated policy based remote attestation in trusted computing. ARPN Journal of Engineering and Applied Sciences. 2016; 11(7):4485-91.

[8] Akhunzada A, Ahmed E, Gani A, Khan MK, Imran M, Guizani S. Securing software defined networks: taxonomy, requirements, and open issues. IEEE Communications Magazine. 2015; 53(4):36-44.

[9] Kalkan K, Gür G, Alagöz F. SDNScore: a statistical defense mechanism against DDoS attacks in SDN environment. In symposium on computers and communications 2017 (pp. 669-75). IEEE.

[10] Andishmand R, Mohammdi H, Mostafavi S. Detection and analysis of DDoS attacks in software-defined networks. Computer Security and Reliability.2020.

[11] Dehkordi AB, Soltanaghaei M, Boroujeni FZ. The DDoS attacks detection through machine learning and

statistical methods in SDN. The Journal of Supercomputing. 2021; 77(3):2383-415.
[12] Conti M, Lal C, Mohammadi R, Rawat U. Lightweight solutions to counter DDoS attacks in software defined networking. Wireless Networks. 2019; 25(5):2751-68.
[13] Santos R, Souza D, Santo W, Ribeiro A, Moreno E. Machine learning algorithms to detect DDoS attacks in SDN. Concurrency and Computation: Practice and Experience. 2020; 32(16).
[14] Yang L, Zhao H. DDoS attack identification and defense using SDN based on machine learning method. In 15th international symposium on pervasive systems, algorithms and networks (I-SPAN) 2018 (pp. 174-8). IEEE.
[15] Ali M, Benamrane F, Luong DK, Hu YF, Li JP, Abdo K. An AI based approach to secure SDN enabled future avionics communications network against DDoS attacks. In digital avionics systems conference 2019 (pp. 1-7). IEEE.
[16] Yu S, Zhang J, Liu J, Zhang X, Li Y, Xu T. A cooperative DDoS attack detection scheme based on entropy and ensemble learning in SDN. EURASIP Journal on Wireless Communications and Networking. 2021.
[17] Manso P, Moura J, Serrão C. SDN-based intrusion detection system for early detection and mitigation of DDoS attacks. Information. 2019; 10(3):1-17.
[18] Priyadarshini R, Barik RK. A deep learning based intelligent framework to mitigate DDoS attack in fog environment. Journal of King Saud University-Computer and Information Sciences. 2019:1-7.
[19] Joëlle MM, Park YH. Strategies for detecting and mitigating DDoS attacks in SDN: a survey. Journal of Intelligent & Fuzzy Systems. 2018; 35(6):5913-25.
[20] Karan BV, Narayan DG, Hiremath PS. Detection of DDoS attacks in software defined networks. In 3rd international conference on computational systems and information technology for sustainable solutions 2018 (pp. 265-70). IEEE.
[21] Meti N, Narayan DG, Baligar VP. Detection of distributed denial of service attacks using machine learning algorithms in software defined networks. In international conference on advances in computing, communications and informatics 2017 (pp. 1366-71). IEEE.
[22] Fitriani S, Mandala S, Murti MA. Review of semi-supervised method for intrusion detection system. In Asia pacific conference on multimedia and broadcasting 2016 (pp. 36-41). IEEE.
[23] Sharma S, Sahu SK, Jena SK. On selection of attributes for entropy based detection of DDoS. In international conference on advances in computing, communications and informatics 2015 (pp. 1096-100). IEEE.
[24] Mehdi SA, Khalid J, Khayam SA. Revisiting traffic anomaly detection using software defined networking. In international workshop on recent advances in intrusion detection 2011 (pp. 161-80). Springer, Berlin, Heidelberg.
[25] Omar T, Ho A, Urbina B. Detection of DDoS in SDN environment using entropy-based detection. Califonia State Polytechnic University.
[26] Carvalho RN, Bordim JL, Alchieri EA. Entropy-based DoS attack identification in SDN. In international parallel and distributed processing symposium workshops 2019 (pp. 627-34). IEEE.
[27] Ahmed ME, Ullah S, Kim H. Statistical application fingerprinting for DDoS attack mitigation. IEEE Transactions on Information Forensics and Security. 2018; 14(6):1471-84.
[28] Durner R, Lorenz C, Wiedemann M, Kellerer W. Detecting and mitigating denial of service attacks against the data plane in software defined networks. In conference on network softwarization 2017 (pp. 1-6). IEEE.
[29] Gkountis C, Taha M, Lloret J, Kambourakis G. Lightweight algorithm for protecting SDN controller against DDoS attacks. In IFIP wireless and mobile networking conference 2017 (pp. 1-6). IEEE.
[30] Sahay R, Blanc G, Zhang Z, Debar H. ArOMA: an SDN based autonomic DDoS mitigation framework. Computers & Security. 2017; 70:482-99.
[31] Sharma PK, Singh S, Park JH. OpCloudSec: open cloud software defined wireless network security for the internet of things. Computer Communications. 2018; 122:1-8.
[32] Singh J, Behal S. Detection and mitigation of DDoS attacks in SDN: a comprehensive review, research challenges and future directions. Computer Science Review. 2020.
[33] Aamir M, Zaidi SM. Clustering based semi-supervised machine learning for DDoS attack classification. Journal of King Saud University-Computer and Information Sciences. 2019; 33(4):436-46.
[34] Ye J, Cheng X, Zhu J, Feng L, Song L. A DDoS attack detection method based on SVM in software defined network. Security and Communication Networks. 2018:1-8.
[35] Tuan NN, Hung PH, Nghia ND, Tho NV, Phan TV, Thanh NH. A DDoS attack mitigation scheme in ISP networks using machine learning based on SDN. Electronics. 2020; 9(3):1-19.
[36] Phan TV, Bao NK, Park M. A novel hybrid flow-based handler with DDoS attacks in software-defined networking. In conferences on ubiquitous intelligence & computing, advanced and trusted computing, scalable computing and communications, cloud and big data computing, internet of people, and smart world congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld) 2016 J (pp. 350-7). IEEE.
[37] Mehmood A, Mukherjee M, Ahmed SH, Song H, Malik KM. NBC-MAIDS: naïve bayesian classification technique in multi-agent system-enriched IDS for securing IoT against DDoS attacks. The Journal of Supercomputing. 2018; 74(10):5156-70.
[38] Ravi N, Shalinie SM. Learning-driven detection and mitigation of DDoS attack in IoT via SDN-cloud

architecture. IEEE Internet of Things Journal. 2020; 7(4):3559-70.
[39] Alshamrani A, Chowdhary A, Pisharody S, Lu D, Huang D. A defense system for defeating DDoS attacks in SDN based networks. In proceedings of the ACM international symposium on mobility management and wireless access 2017(pp. 83-92).
[40] Yang X, Han B, Sun Z, Huang J. SDN-based ddos attack detection with cross-plane collaboration and lightweight flow monitoring. In global communications conference 2017 (pp. 1-6). IEEE.
[41] Saravanan A, Bama SS, Kadry S, Ramasamy LK. A new framework to alleviate DDoS vulnerabilities in cloud computing. International Journal of Electrical & Computer Engineering. 2019; 9(5): 4163−75.
[42] Saravana A, Sathya BS. Multi model anti DDoS framework for detection and mitigation of high rate DDoS attacks in the cloud environment. International Journal of Scientific & Technology Research. 2020; 9(3):4503-11.
[43] Hu D, Hong P, Chen Y. FADM: DDoS flooding attack detection and mitigation system in software-defined networking. In global communications conference 2017 (pp. 1-7). IEEE.
[44] Irfan AMS, Riyad AM. Rough set theory based entropy approach for feature selection in adaptive intrusion detection system. International Journal of Scientific & Technology Research, 2020; 9(3):5734-5740.
[45] http://ntwag.sourceforge.net/. Accessed 17 September 2021.
[46] http://nsl.cs.unb.ca/nsl-kdd/. Accessed 17 September 2021.
[47] Bama SS, Ahmed MI, Saravanan A. A survey on performance evaluation measures for information retrieval system. International Research Journal of Engineering and Technology. 2015; 2(2):1015-20.

**Dr Riyad AM** is an Assistant Professor and head of the department of computer science at the EMEA College of arts and science, Malappuram district, Kerala, India. He did his Ph.D from Bharathiar University, Tamilnadu, India. He has done two M.Phil programmes from Bharathiar University in data mining and software engineering respectively. His research areas are Network Security and Data Mining. He has published various research papers in international journals and conferences.
Email: amriyad@gmail.com

## Appendix I

| S.No. | Abbreviation | Description |
|---|---|---|
| 1 | ADT | Average Detection Time |
| 2 | ANN | Artificial Neural Networks |
| 3 | API | Application Programming Interface |
| 4 | APV | Actual Parameter Value |
| 5 | AT | Average Throughput |
| 6 | CFS | Correlation based Feature Selection |
| 7 | DBN | Deep Belief Network |
| 8 | DDoS | Distributed Denial of Service attack |
| 9 | DoS | Denial of Service |
| 10 | DT | Decision Tree |
| 11 | EA | Ensemble Approach |
| 12 | ELM | Extreme Learning Machines |
| 13 | EMA | Exponential Moving Average |
| 14 | F | F-measure |
| | ICMP | Internet Control Message Protocol |
| 15 | IoT | Internet of Things |
| | IP | Internet Protocol |
| 16 | KNN | K-Nearest Neighbours |
| 17 | LB | Lower Bound |
| 18 | LEDEM | Learning Driven Detection Mitigation |
| 19 | NA | Network Accuracy |
| 20 | NB | Naive Bayes |
| | NN | Neural Network |
| 21 | PBD | Precision of Benign Detection |
| | PCA | Principal Component Analysis |
| 22 | PDD | Precision of DDoS Detection |
| 23 | R2L | Remote to Local |
| 24 | RBD | Recall of Benign Detection |
| 25 | RDD | Recall of DDoS Detection |
| 26 | RF | Random Forest |
| 27 | RST | Rough Set Theory |
| 28 | SDLEM | Semi-supervised Deep Extreme Learning Machine |
| 29 | SDN | Software Defined Networking |
| 30 | SVM | Support Vector Machines |
| | TCP | Transmission Control Protocol |
| 31 | U2R | User to Root |
| 32 | UB | Upper Bound |
| | UDP | User Datagram Protocol |