www.revistageintec.net
ISSN: 2237-0722
revista GEINTEC Gestão, Inovação e Tecnologias

Towards Detecting Flooding DDOS Attacks Over Software Defined Networks Using Machine Learning Techniques

Ancy Sherin Jose[1]; Latha R Nair[2]; Varghese Paul[3]

[1]Division of Computer Science and Engineering, Cochin University of Science and Technology, India.

[1]ancysherin@gmail.com

[2]Division of Computer Science and Engineering, Cochin University of Science and Technology, India.

[2]latharnair@cusat.ac.in

[3]Department of Computer Science and Engineering, Rajagiri School of Engineering and Technology, India.

[3]vp.itcusat@gmail.com

Abstract

Distributed Denial of Service Attack (DDoS) has emerged as a major threat to cyber space. A DDoS attack aims at exhausting the resources of the victim causing financial and reputational damages to it. The availability of free software make launching of DDoS attacks easy. The difficulty in differentiating a DDoS traffic from a legitimate traffic burst such as a flash crowd makes DDoS difficult to be identified. A wide range of techniques have been used in conventional networks to detect and mitigate DDoS attacks. Though the advent of Software Defined Networking (SDN) makes a network easy to be managed even SDN is vulnerable to DDoS attacks. In this case, the controller of the SDN gets overloaded with the incoming packets from the switches. In fact, a solution based on security analytics can be put in place to ward off this threat as a proactive security measure using the flow level statistics available from the SDN. Compared to the packet analysis used in traditional networks which is resource expensive the flow level statistics is relatively inexpensive. This paper focuses on the design and implementation of an attack detection system for detecting the flooding DDoS attacks TCP SYN flooding attacks, HTTP request flooding attacks, UDP flooding attacks and ICMP flooding attacks over SDN network traffic. The system uses various classification algorithms to classify a traffic into normal or attack. The feature sets for classification were arrived at using a feature selection module with ANOVA (Analysis of Variance) F-Test statistical method. Performance evaluation of each of the classifiers was carried out for the three feature sets obtained from the feature selection module using various performance measures and the results have been tabulated. The feature set which gives the best performance in detecting malicious traffic has been identified.

Key-words: Software Defined Networking, Machine Learning (ML), Feature Selection, Binary Classification, DDoS Attacks, Attack Detection.

1. Introduction

Software Defined Networking (SDN) is an emerging networking technology, which eliminates the limitations of conventional networks. Complex nature of traditional networks, configuration of individual devices using the vendor specific languages, lack of global view of the network and centralized controlling point were some of the bottlenecks of traditional networks [1]. With the introduction of SDN, global view of network was made possible and this helped for easier configuration and management of networks [2]. The separation of control plane from the data plane is the major highlight of the SDN. The SDN architecture is centered with a logically centralized controller which acts as the network brain. The controller serves as a network operating system. In the SDN architecture, the network becomes programmable through high level programming languages, easily configurable and manageable [1].

Distributed Denial of Service Attack (DDoS) has emerged as a major threat to cyber space. DDoS aims at exhausting the resources of the victim preventing legitimate users from accessing resources thereby causing financial and reputation damages to it. Though SDN is a promising solution and the future of networks, the same can be plagued by DDoS attacks. As the name indicates, DDoS attacks are distributed in its nature and can be launched across the globe by distributed botnets. The distributed nature of attack, variable duration pattern of the attack, variety in the volume of attack, the usage of spoofed IP address and the difficulty in identifying the traffic features are some of the chief reasons which make DDoS hard to be detected and addressed [3].

A wide range of techniques have been used in conventional networks to reduce the effect of DDoS attacks [4]. The packet analysis in traditional networks, was resource expensive and thus sampling techniques were used to verify the packets. The Cisco flow monitoring technology Netflow and packet sampling technology S-flow were used for traffic collection and analytics [8]. Due to the programmable nature of SDN, flow rules can be dynamically inserted into the flow table when a DDoS attack is detected. Many defense mechanisms to detect and mitigate DDoS attacks in SDN use statistical, machine learning and deep learning techniques [6]. OpenFlow which is the commonly used southbound API for communication between switches and controller has the ability to provide the flow statistics. From the flow statistics provided by SDN switches, the necessary features can be extracted and can be used with machine learning techniques for security analytics [9]. Many works use the flow features provided by OpenFlow to detect the DDoS attacks in SDN [11].

This work attempts to detect the presence of DDoS flooding attacks from the flow level features collected from the switches. As SDN follows a flow-based architecture, flow level features can be

easily extracted. Compared to packet analysis, flow analysis is resource inexpensive. The system detects four DDoS attacks: TCP SYN flooding attacks, HTTP request flooding attacks, UDP flooding attacks and ICMP flooding attacks over a SDN simulated network traffic. The system uses various classification algorithms to classify a traffic into normal or attack. The feature sets or feature groups for classification were arrived at using a feature selection module. Performance evaluation of each of the classifiers was carried out for the three feature sets obtained from feature selection module using various performance measures and the results have been tabulated. The feature set which gives the best performance in detecting malicious traffic had been identified.

The arrangement of paper is as follows. Section 2 describes the research questions and the contributions of this work. Section 3 describes the background concepts of DDoS attacks, SDN architecture and Machine Learning classifiers used in the study. Section 4 discusses the important related works on detection of DDoS attacks. Section 5 describes the design of the attack detection system. Implementation of the work is described in Section 6. Section 7 discusses the performance evaluation and important observations. Section 8 provides the conclusion.

2. Research Questions and Contributions of the Work

2.1. Research Questions

Following are the research questions we attempt to address in this work.

1. Determine the effectiveness of SDN flow level features in detecting DDoS attacks. Determine the feature importance of the flow statistic features for detecting flooding DDoS attacks in SDN environment with the flow statistics information available from the SDN switches. This will help for developing machine learning models which are computationally light weight and suitable for the first stage classification when using multiple stage classification pipeline.
2. How effectively DDoS attacks can be detected by using the features collected from the network layer. Many works use features like ‘growth of ports’ and ‘ratio of pairwise flows’ and application specific features for detecting DDoS attacks. In this work, the SDN controller application was using Layer 3 match constraints for building flow rules in switch and features like ‘growth of ports’ and ‘ratio of pair wise flows’ was not collected. Only 7 flow statistics features related to network layer are used and we experimentally evaluated the performance of machine learning classifiers for detecting the DDoS attacks with these features.

3. When the feature groups are identified, experimentally analyse the performance of basic machine learning classifiers in detecting DDoS attacks. The model built shall be lightweight and shall be used for detecting the flooding attacks in real time.

2.2. Contributions of the Work

Following are the contributions of the current study.

1. Creation of SDN dataset with flow statistics information from switch - In this work instead of using the traditional packet capture datasets, SDN dataset is created. For this, a SDN network is simulated with the Mininet emulator. SDN application over the RYU controller is developed to collect the flow statistics and port statistics information from the switches. The various DDoS attacks were launched individually. The dataset with seven flow statistic features were collected.
2. Determining the feature importance in the context of detecting DDoS attacks in SDN environments using univariate feature selection technique ANOVA FTest - We used the 7 features found in literature by Neelam et al [30]. Further we grouped the features into feature groups based on feature scores using ANOVA F-Test feature selection method. We experimentally evaluated the effectiveness of each feature group for detecting DDoS attacks in our dataset. The most important two features for detecting flooding DDoS attacks in SDN was found to be 'Entropy of protocol and Entropy of source IP address'.
3. Effectiveness of features collected from network layer in determining DDoS attacks in SDN context - From our experimental evaluation, we found that the 7 features collected and used in the study are capable of detecting the DDoS attacks effectively. The port information and application specific features were not used for detecting DDoS attacks in this work. We attempted to detect the HTTP request flooding attack which is an application layer attack and found that the traffic was also detected as malicious with the selected flow level features.

3. Background Concepts

3.1. Distributed Denial of Service Attacks

DDoS attacks are distributed in nature and can be launched across the globe by distributed botnets. They aim to disrupt the services hosted by the target which can bring economic, financial and reputational damages and thereby preventing legitimate users from accessing resources. Various DDoS

attacks have been identified in the past years. In February 2018, a memcached server reflection attack with traffic rate of approximately 1.3 Tbps was launched against the well-known source code repository GitHub [13]. DDoS attacks against Dyn (2016), BBC (2015), Spamhaus (2013) were the other major attacks occurred in the decade [8]. The DDoS attack against DNS provider Dyn was an IoT based botnet attack [14]. According to Kaspersky Lab's DDoS Q4 2019 report, DDoS attacks were doubled when compared to the same period of 2018. Average duration of attack as well as number of smart attacks also increased compared to the previous year. According to DDoS Q4 2020 report, there was only 10% rise in DDoS attacks compared to same period of the previous year. The drop in DDoS attacks for 2020 can be related to the increasing interests in the domain of cryptocurrency mining [15].

The DDoS attacks are categorized into three groups – application layer attacks, protocol-based attacks and volume-based attacks. An application-level DDoS attack is launched across application layer services like HTTP server, NTP server etc. which utilizes the application vulnerabilities. In HTTP request flooding attack, HTTP GET/POST requests from random source IP Address is initiated, and this leads to incomplete half connections as these connections are requested by spoofed IP Address. As a result, connection to the legitimate clients will be blocked. A protocol-based DDoS attack makes use of protocol vulnerabilities. TCP SYN flood attack is a protocol-based DDoS attack, which utilizes the three-way handshaking process. In this, the attacker sends huge number of TCP connection establishment SYN messages, and the server tries to open many connections and reply with SYN/ACK messages [16]. The server continues waiting for ACK from the source host. As the attacker spoofs the source IP address, the server fails to receive an ACK message, and the server maintains many half open connections and finally crashes [17]. In both TCP SYN flood and HTTP flood attacks, huge number of unnecessary connections are made, which opens simultaneously many ports at the victim [17].

A volumetric DDoS attack sends large volume traffic to victims, an example is flooding attack like ICMP flooding attack and UDP flooding attacks. In the UDP flood attacks victims are overwhelmed by datagrams that comes from spoofed source IP address while in ICMP flooding attack, the victims are overwhelmed by ICMP echo requests. A legitimate traffic contains at least 5 packets [18] [19] and any traffic which contains less than 5 packets, can be considered as abnormal. In order to easily launch the attack and to save the resources at attacker end, attacker prefers to initiate DDoS with very less packet size [20].

In the SDN scenario, both the switches and the controllers can be affected by DDoS. The switches in SDN are simple forwarding devices, which forward packets based on the rules present in the flow tables which are inserted by the controller. Whenever a switch receives a packet, it will check with the matching rule in its flow table and decide to act according to the action defined for that rule.

If the rule is not found, it requests controller for guidance. This request is initiated from the switch as a PACKET_IN message, in OpenFlow based systems. Upon receiving the PACKET_IN message, the controller checks the packet and will insert necessary flow rule in the switch. A DDoS attack sends numerous packets to the network which are often spoofed. The attacker may use botnets to host DDoS attack. The SDN switch will receive many packets which will overwhelm the controller with PACKET_IN messages. The controller will add countless flow rules in the switches which can lead to flow table overloading in the switch [3]. Controller becomes unavailable due to the processing of large number of spoofed requests. This makes the switches and the controller exhausted, leading to the crashing of the network. The attack tree and the attack models help in identifying the impact of DDoS attacks over a network [17].

3.2. SDN Architecture and Controller to Switch Communication

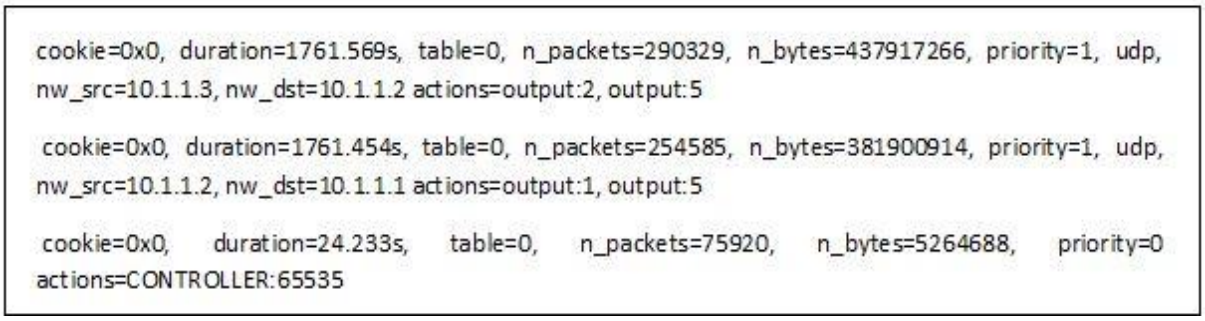
The SDN has a decoupled architecture with a controller which constitutes the control plane and the switches which constitute the data plane [4]. The controller and switches communicate with each other through the secure connection between them. The South bound API, most commonly OpenFlow, is the communication API between the controller and the switch. The SDN enables applications to be written in high level programming languages to communicate with the controller. These applications communicate with the controller using the Northbound API, REST API is a commonly used one.

The OpenFlow enabled SDN switches maintain a pipeline of flow tables which are used for packet forwarding [1]. The flow table contains flow rules which define the actions that should be carried out when a packet is received. The flow rules are defined with match fields, counters and instructions [21]. A match defines a set of conditions for matching an incoming packet and the instructions define the actions to be performed on the matching packets. The flow table contains a default rule (table miss entry) to forward the packet to the controller if a particular flow rule doesn't exist in the flow table. A flow can be defined as a group of packets that has same features like source IP, destination IP, source port, destination port or VLAN. In the absence of a flow rule the switch forwards the packet to the controller as the PACKET_IN message. The byte counters and the packet counters for a flow rule can be used for extracting the features specific to the flow. In OpenFlow based SDN, the switches send statistical messages to the controller with the flow statistics information. The two common statistical messages that can be requested to the switches are the individual flow statistics messages and the aggregate flow statistics messages. The individual flow statistics can be retrieved by sending

OFPMP_FLOW request [22]. This message is a multipart request. A flow entry contains all the details of a flow.

Sudo ovs-ofctl dump-flows s1 command can be used to retrieve flow statistics of datapath S1. A typical flow statistic reply message is given in Figure 1. Three flow entries of the switch S1 with the duration of the flow, number of packets and bytes handled by the flow are retrieved from the switch. The default flow rule is represented by the flow table miss entry whose action is to forward the packet to the controller. Other major actions include forwarding the packet to a particular port, dropping the packet or flooding the packet across all the ports.

Figure 1 - Flow Table Rules



The aggregate flow statistics message provides aggregate information about all the flow entries present in the flow table. In OpenFlow, a OFPMP_AGGREGATE request message is sent from the network application residing over the controller to provide the aggregate statistics of the flow table [22].

3.3. Machine Learning Classifiers

Six machine learning classifiers are used in this work to accomplish the binary classification task.

Logistic Regression

The simplest supervised machine learning classifier logistic regression uses a cost function which is a sigmoid function to map the predictions to probabilities of occurrence of an event. The output of the function which ranges between 0 and 1 is used for labelling the observations to discrete classes based on the value set for the threshold. The equation of logistic regression [51] is

$$P(X) = \frac{e^{(b_0+b_1X)}}{1+e^{(b_0+b_1X)}} \quad (1)$$

where P(X) is the probability of new instance to be of particular class and it is always between 0 and 1, b_0 is the constant or bias and b_1 is the coefficient for the independent variable and e is the base of natural log. When P(X) is greater than the threshold value 0.5, then new instance is classified to class 1 and to class 0 otherwise [52].

Naive Bayes

Naive Bayes is a probabilistic machine learning classifier that is based on Bayes theorem. The algorithm assumes that each attribute is independent [53] and contributes equally for the prediction of the class. The characteristic equation for Naive Bayes [54] is denoted by (2).

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)} \quad (2)$$

The posterior probability of each instance to be of a target class is calculated using the above equation for each class and the instance belongs to the class with the highest probabilistic class value. Naive Bayes algorithm is extensively used for classification tasks in literature. Reasonably good performance could be achieved using this method in this work.

Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a supervised classification technique, which is also used for dimensionality reduction. LDA supports binary as well as multi-class classification problems and is based on Bayes theorem. When LDA is used as classifier, the new instance will be assigned to the class which yields the largest discriminant function value. The derived discriminant function [51] is denoted by the equation (3)

$$\delta_k(x) = \frac{x.\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k) \quad (3)$$

where μ_k and σ^2 are mean and covariance for the kth class and π_k is the prior probability for an instance to belong in kth class. LDA is found very effective when the class frequencies are not same [54]. In this work Linear Discriminant Analysis is used and obtained high accuracy score for all the three feature groups.

SVM

SVM introduced in 1992 is used for classification and regression problems. In SVM, classification is performed by finding the hyper plane which classifies the high dimensional data points into separate predefined classes [52]. The distance between the hyperplane and the support vectors forms the margin of hyperplane. The decision boundary which maximizes the margin between the classes is the optimal hyperplane of SVM. The kernel functions – Linear, Polynomial and Radial Basis Function are selected based on the dataset [56]. In this work, we used Support Vector Machine with polynomial kernel and the classifier yielded the highest accuracy score.

k-Nearest Neighbor (k-NN)

This Machine learning classifier assigns the new instance to the class for which the nearest neighbors of the instance in the training set is assigned. This is done by calculating the distance between the new instance and its neighbors in the training set [56]. The k neighbors with the minimum distance commonly Euclidean distance is selected and the class label of the selected neighbors is assigned to the new instance. Distance parameter is computed for the neighbors to obtain the similarity of the instance with its neighbors. The similarity function used and the selection of parameter k affects the performance of kNN [57]. The performance of kNN for detecting normal and malicious traffic is evaluated in the work. It is noted that KNN takes higher fitting time compared to other algorithms.

Random Forest

High predictive performance is obtained for classification tasks with Random Forest classifier as it uses an ensemble of Decision Trees. The multiple decision trees contribute in classification in such a way that each tree in the forest provides the decision about the class to which the new instance should be assigned. The class label of the new class will be the class which gets the majority vote [53]. Higher accuracy is obtained when number of trees participating in decision making is increased. The number of trees has to be provided before applying the classifier on the datasets.

4. Related Works

Ye Jin et al. had tried SVM based classification technique to detect the presence of DDoS attacks in SDN [10]. Mininet simulated network was created, normal and attack traffic were injected

into the network. Attack traffic was generated by Hping3 tool. This work achieved the accuracy of 95.24%. Phan Trung V et al. had attempted SVM based classification and had designed Idle Timeout Adjustment algorithm to handle DDoS attacks [23]. DDoS attack is classified into two types where Type 1 attacks send few flows to the victims with high volume of packets. Type II attacks send many number of flows, and each flow sends only small number of packets. CAIDA dataset was used for training ML. Both these works were carried out over SDN provided flow level features. OverWatch [24] leverages machine learning based classification algorithm in the control plane, and flow monitoring algorithm in data plane to predict the features of a flow. This work was carried out on a real-world network which extends partial intelligence to the switch. Rahman Obaid et al. compared different ML algorithms to analyse the captured packets over Mininet simulated SDN network [25]. 24 packet level features were used to detect the attack in their work. Hidden Markov Model has been tried to detect LDDoS attack and this work was carried out by Wang et al. [26]. Multiclass SVM classification was done by Kokila et al. [9]. Apart from source IP and destination IP with port, packet length was used with Radial Basis Function (RBF) kernel SVM classification. SDN/NFV in conjunction with machine learning technique was employed by Park, Younghee et al. [11]. Virtual Network Function (VNF) was implemented in data plane to extract features in real time and Random Forest algorithm was used to detect the presence of attacks in the work. Work by Lohit et.al [27] analyse different ML techniques over the real time dataset obtained from Lawrence Berkley Laboratory. Braga et al. attempted Self-Organizing Map with 6 tuple attributes [20]. Average bytes per flow and average packets per flow etc. were used as features for detection of DDoS in this popular research work. Considering the flow entries with high number of packets, they used median of byte count and packet count instead of calculating simple average of packet count. Yang et al. attempted SVM based classification on KDD99 dataset. Packet sniffer was used to extract 8 packet features for this work. Seven node neural network based analysis over Apache Spark cluster was tried out by Hsieh et al. to detect DDoS attack [50]. The training was carried out on 2000 Darpa LLDoS 1.0 dataset. XGBoost algorithm for DDoS detection in SDN based cloud environment was carried out by Chen et al. [29]. Tcpdump was used to collect packet data, and an accuracy rate of 98.53% was achieved. RBF network with Particle Swarm Optimization (PSO) is employed by Neelam et. al [30] for detecting the presence of DDoS attacks. They have also identified the features necessary to detect the various DDoS attack types in [17].

Dehkordi et al. [31] employs entropy-based filtering to sort suspicious flows. Entropy of IP address in the SDN network is calculated and static and dynamic entropy thresholds are applied. If the entropy falls below the threshold, that flows are suspicious and subjected to classification. 15 features

are selected for the classification. This work could successfully detect the presence of high volume and low volume DDoS attacks. Bayes Net, J48, Random Tree, logistic regression and REP Tree classifiers were used in this work. Tuan et al. [32] used k-NN and XGBoost methods to detect the presence of TCP SYN flood and ICMP flooding attacks in SDN based Internet Service Provider Networks. The work uses CAIDA 2007 dataset and Bonesi traffic to create a testbed environment. While detecting an attack, a flow rule with drop action was added to the flow table. This work achieved 98% accuracy in detecting ICMP and TCP SYN flooding attacks.

Sahoo et al. [33] makes use of SVM based classification technique to detect the presence of DDoS attack. In this work kernel principal component analysis (KPCA) technique is used for dimensionality reduction and Genetic Algorithm (GA) is used for SVM parameter optimization. The model achieved accuracy of 98.90% and the work was evaluated against two different datasets. Six machine learning algorithms were used in the work by Diaz et al. [34] in detecting Low-Rate DDoS attacks (LR -DDoS). The work used CIC DOS 2017 datasets as this dataset captures LR -DDoS attacks. Random and grid search hyper parameter optimization techniques were also used. The work achieved accuracy rate of 95%. The work by Sen et al. [35] used Adaboost algorithm with decision stump as weak classifier. The network was simulated and sflow-RT was used to monitor the collected data. 20-fold cross validation technique was used to validate the results of classification. DDoS detection accuracy for this work is 93%. In the work by Polat et al. [36], filter, wrapper and embedded feature selection techniques were used to detect the presence of TCP, UDP and ICMP attacks with SVM, Naive Bayes, Artificial Neural Networks and k Nearest Neighbors classifiers. K-NN algorithm with wrapper feature selection technique yielded accuracy of 98.30% with 10-fold cross validation.

Niyaz Quamar et al. [37] used Stacked Auto Encoder based deep learning model to execute 8 – class classification for DDoS flooding attack. Sparse Auto Encoder was used for finding the optimal features from a set of handpicked features. Deep learning models - RNN, CNN and LSTM were used to detect DDoS attacks in SDN based network by Li et al. [38]. High accuracy could be achieved using ISCX dataset. RBM - Restricted Boltzmann Machine was employed for detecting DDoS by Imamverdiyev et al. [39]. The experiments were done on NSL-KDD dataset. DeepDefense [40] uses Recurrent Neural Network (RNN) model to detect DDoS attack and the work was evaluated using ISCX2012 dataset. This work achieved accuracy of 98.410%. LUCID [41] makes use of Convolutional Neural Networks (CNN) for detection and performs well in resource limited environments. The work uses three standard datasets – ISCX2012, CIC2017 and CSECIC2018 [42]. The important works in this area, the features used and the detected DDoS attacks are summarized in Table 1.

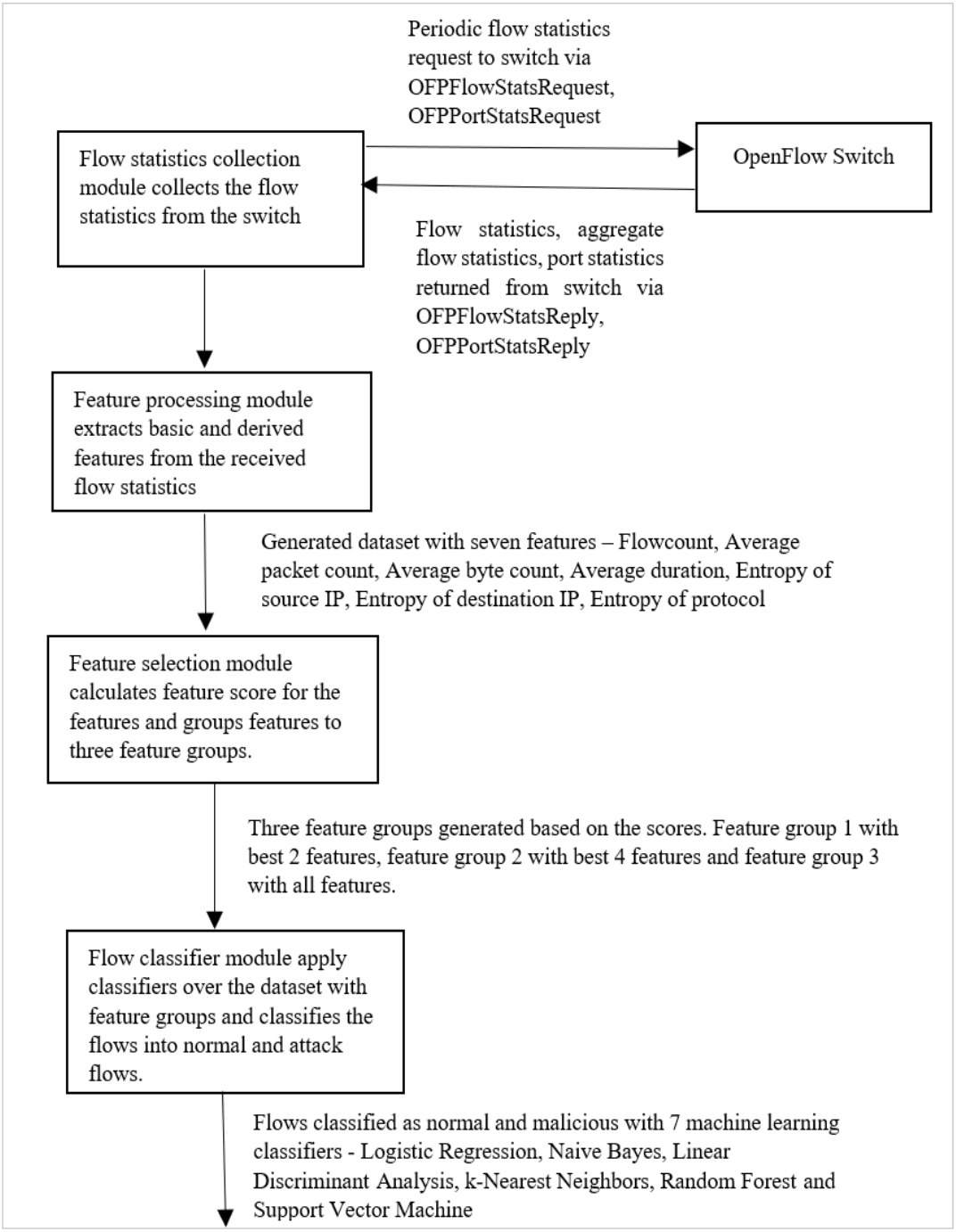
Table 1 - Important Previous Works in Attack Detection

Classifier	Features	Attacks detected
Self-Organizing Maps Braga et al. [20]	Average of packets per flow, Average of bytes per flow, Average of duration per flow, percentage of pair flows, growth of single flows, growth of different ports	TCP SYN flood, UDP flood, ICMP flood
Radial Basis Function Network with Particle Swarm Optimization Neelam et al. [30]	Average packets per flow, Average bytes per flow, Number of flows per second, Average duration per flow, Entropy of destination IP address per second, Entropy of source IP address per second, Entropy of IP protocol per second	TCP SYN flood, UDP flood, ICMP flood
Support Vector Machine Jin Ye et al. [10]	Speed of source IP, Standard Deviation of flow packets, Standard Deviation of flow bytes, Speed of flow entries, Ratio of pair flow	TCP SYN flood, UDP flood, ICMP flood
k-Nearest Neighbors Liehuang Zhu et al. [43]	Median of packets per flow, Median of bytes per flow, Percentage of corelative flow, growth of ports, growth of source IP address	Cross domain DDoS attacks
Rule Based Christos Gkountis et al. [18]	Packet average, Byte average	TCP SYN flood, UDP flood, ICMP flood

5. Classification System Design

This section describes the design of the attack classification system. The system uses the flow statistics from the switches and classifies the traffic into normal and malicious. The system mainly has 4 modules namely flow statistics collection module, feature processing module, feature selection module, and flow classifier module. The system process flow is depicted in Figure 2.

Figure 2 - System Process Flow



5.1. Flow Statistics Collection Module

The flow statistics collection module is responsible for extracting the flow features. The application sends request messages to the switches for the flow statistics, the aggregate flow statistics and the port statistics. An interval of 3 secs is considered for the flow statistics request.

5.2. Feature Processing Module

Feature processing module is responsible for extracting the basic and derived features from the received flow statistics. Based on literature survey on previous works, seven flow level features - flow count, average of packet count per flow, average of byte count per flow, average of flow duration, entropy of source IP, entropy of destination IP, entropy of protocol have been used to detect the presence of DDoS attacks. [17] [20] [19].

a. Flow count: It denotes the count of flows present in the data path during the current time period. An increase in flow count is an indicator of DDoS attack. The increase in flow can also be due to flash crowds. Flash crowds are caused when a large number of legitimate users access the resources at the same time.

b. Average of packet count per flow: It is the average of the number of packets for n flows, taken for a time period. In the event of an attack, packet count tends to fall [20] [50]. TCP SYN flooding attack and HTTP flooding attack aim to achieve maximum port consumption by sending minimum number of packets to the victim. Instead of using simple average as the reference, median of packet count is taken for the study. When the number of packets per flow is significantly large, the average computation may smooth the feature [43] [20]. Median is calculated as per equation 4.

$$\text{Median}(F) = \begin{cases} F\left(\frac{n+1}{2}\right) & \text{when n is odd} \\ \frac{F(\frac{n}{2}) + F\left(\frac{n+1}{2}\right)}{2} & \text{otherwise} \end{cases} \quad (4)$$

where F contains all the flows for the interval.

c. Average of byte count per flow: It is the average of number of bytes for the flows during the time interval. In the event of an attack, byte count diminishes, as attacker tries to send tiny packets to save the resources at its end. The average byte count is calculated using equation (4)

d. Average duration of flow: Duration of a flow refers to the total life time of the flow in the data path. Depending on the type of the attack, duration of flow can be either low or high [30]. Average duration of the flow is also calculated using equation (4).

e. Entropy of source IP: High entropy is resulted by a more dispersed probability distribution [16]. In order to achieve many half open connections at the victim, the attacker uses random

source IP address for initiating TCP SYN flooding and HTTP flooding attacks. As a result, entropy of the source IP increases during the attack [17].

f. Entropy of destination IP: Concentration of a distribution is denoted by low entropy [16]. During the DDoS attacks, entropy of destination IP decreases, as the attacker tends to focus on sending traffic to few victim machines. [17].

g. Entropy of protocol: Compared to the normal period, entropy of the protocol tends to decrease during the attack period, as the attack traffic makes use of a single protocol in case of single vector attacks [17].

The features extracted from the feature processing module is stored in a CSV file and is used as the dataset for the study. The classifier is trained with the data set and is used to classify the flows extracted from the switches into normal and attack instances. Though the mirrored traffic is captured as packet capture (pcap file), only flow level analysis is performed, as it is resource inexpensive compared to packet capture analysis.

5.3. Feature Selection Module

Feature selection module is responsible for selecting the best features for the SDN dataset from the 7 flow features. In this work, ANOVA (Analysis of Variance) F-Test is used for ranking the features. ANOVA F-Test is a statistical univariate method which measures the individual variation of the members within the class and variation in the means of classes [58]. Feature selection module feeds on the dataset generated through feature processing module. After performing data pre-processing steps on the data set, the module employs SelectKBest class of scikit-learn package to select the best features ranked using ANOVA F-Test. Feature groups were formed by selecting the best features based on feature score. The overall process is depicted in Algorithm 1.

Algorithm 1: Feature Selection Module

1: **Procedure** FeatureSelection ()
2: **Input: Mi** = Dataset obtained from feature processing module
3: **Output**: Feature groups FG1, FG2, FG3
4: **Compute** feature score for all features using **ANOVA F-Test**
5: **Group** features based on feature score and store in feature groups
FG1: select the best two features based on score using SelectKBest
FG2: select the best four features based on score using SelectKBest
FG3: select all features using SelectKBest

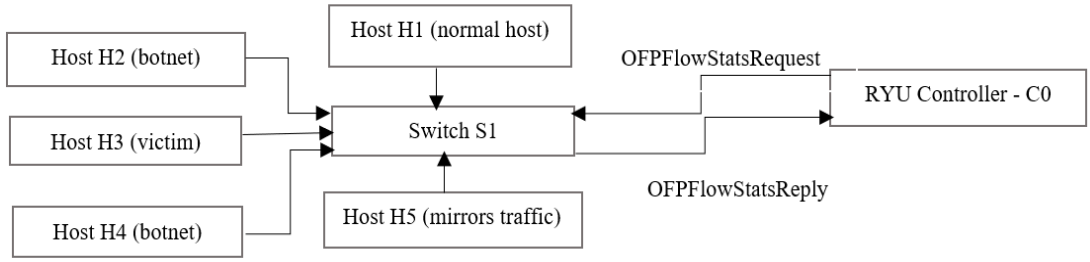
5.4. Flow Classifier Module

The flow classifier module is responsible for classifying the traffic flow into an attack traffic or a normal traffic. In this work, binary classification algorithms namely Logistic Regression (LR), Naive Bayes - Gaussian (NB), Linear Discriminant Analysis (LDA), k Nearest Neighbor (kNN), Random Forest (RF) and Support Vector Machines (SVM with polynomial kernel) have been used over the selected features to detect the presence of an attack.

6. Implementation

The experimental analysis of this work was done in Mininet [44] simulated environment with RYU controller [45]. The network topology and controller-switch communication are depicted in Figure 3. It consists of a single switch network, with S1 as the switch, connected to Host H1 to Host H5. The host H1 is a normal host where as H2 and H4 are botnets which inject attack traffic. The host H3 is the victim. The host H5 records mirrored traffic across all the hosts and saves them in the form of packet capture (pcap) file. The system was implemented using python scripts for generating normal and attack traffic.

Figure 3 - Network Topology and Communication with Controller



Classification system consists of the following steps:

1. **Network simulation** – The network is simulated by executing a python script (network Generator) in the laptop. A simple network is created with 5 hosts connected to a single switch, which is controlled by a RYU controller. Layer 3 switching application is used to control the transmission and matching, and the flow tables are populated with layer 3 information.
2. **Flow statistics collection** – This module periodically requests flow statistics from the switches in every 3sec. Individual flow statistics, aggregate flow statistics and port statistics were collected periodically.

Received: 17.06.2021 – Accepted: 18.07.2021

3. **Traffic generation** – This module generates normal and attack traffic for the user provided time argument. The hardware/software specification of the machine and the tools used for traffic generation are listed in Table 2.

Table 2 - Hardware/Software Specifications and the Tools used for Generating Traffic

S. No	Description	Specification	Version
1	Hardware and software specification	Intel (R) Cor (TM) i7-7500U CPU@ 2.70GHz Multicore (4 core) processor 64-bit, 12 GB RAM	Ubuntu 16.04.7 LTS
2	Network Simulation	Mininet	v 2.2.2
3	SDN Controller	RYU	v 4.30
4	Normal Traffic generation	D-ITG Iperf	v 2.8.1 v 2.0.5
5	TCP SYN flooding, UDP flooding ICMP flooding	Hping3	v 3.0.0-alpha-2
6	HTTP flooding	Bonesi	v 0.3.1

The following steps were executed for traffic generation:

Step 1: The experiment started by executing network generator script. The normal traffic was injected for 4 days. This includes HTTP traffic, UDP traffic, VOIP traffic and ICMP traffic. HTTP traffic was generated by requesting a web page from the webserver. The other tools used for generating normal traffic are listed in Table 2. Traffic features were captured from the flows and were labelled appropriately.

Step 2: TCP SYN flood attack was injected using Hping3 and 10834 rows were captured.

Step 3: HTTP flooding attack was launched next day by using Bonesi tool. HTTP request flood attack was launched by web page request from 50000 random IPs in a closed environment and 12485 rows were captured in the dataset.

Step 4: ICMP flooding attack was launched for another day and 19329 rows were captured.

Step 5: Finally, UDP flooding attack was also injected with Hping 3. The attack flows were labelled appropriately, there were 12334 rows of UDP flood traffic.

The collected dataset had a total of 160115 rows of flow statistics with normal traffic of 105133 rows and attack traffic of 54982 rows. Highest CPU utilization of 99.7% was found while launching attacks.

Data Pre-processing and Feature Selection

The dataset was generated with all the seven features. Data cleaning was applied by removing the NaN values as the first step. Standard scaling was applied. Feature importance was determined by using univariate feature selection technique ANOVA F-Test and features were selected with scikit-learn SelectKBest class of scikit-learn package. This was accomplished by the usage of f_classif function with SelectKBest. The features were grouped into feature groups based on the feature score. The features and the feature scores are listed in Table 3. Feature groups are tabulated in Table 4.

Table 3 - Features and Feature Score

S. No	Feature Name	Abbreviation	Feature score
1	Flow count	Flw_cnt	53489.16
2	Average packet count	Avg_pkt	2120.87
3	Average byte count	Avg_byte	6092.48
4	Average duration	Avg_dur	7301.02
5	Entropy of source IP	Ent_SIP	86627.14
6	Entropy of destination IP	Ent_DIP	74049.53
7	Entropy of protocol	Ent_proto	1329681.91

Table 4 - Feature Groups

S. No	Feature Group	k (number of selected features)	Selected features
1	Feature group 1	Best 2 features	Entropy of protocol, Entropy of source IP
2	Feature group 2	Best 4 features	Entropy of protocol, Entropy of source IP, Entropy of destination IP, Flow count
3	Feature group 3	All 7 features	Entropy of protocol, Entropy of source IP, Entropy of destination IP, Flow count, Average duration, Average byte count, Average packet count

Classification

The pre-processed data was split into 70% training set and 30% testing set. Binary classification to classify the data into normal and attack traffic was performed using the six classification algorithms and 3 feature groups with 10-fold cross validation. The classifiers used in this work are – Logistic Regression (LR), Naive Bayes - Gaussian (NB), Linear Discriminant Analysis (LDA), k-Nearest Neighbors with 5 neighbors (k-NN), Random Forest with 5 estimators (RF) and Support Vector

Machine (SVM with polynomial kernel). Python scikit-learn was used for machine learning based classification.

7. Results and Discussion

Performance of classifiers and feature selection by ANOVA F-Test was evaluated using the four important performance metrics. They are Accuracy, Recall, Precision and F1 score. Accuracy denotes the correctness of algorithm while detecting the attacks over the normal and the attack traffic. Recall indicates the percent of actual attack traffic that are identified correctly. Precision denotes the percentage of positive identification of attack over total predicted positive cases. F1 score which combines recall and precision is also computed and definitions of the metrics are as follows.

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

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

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

$$F1 = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}$$

In this work, 10-fold cross validation is used to evaluate the performance of machine learning classifiers. Fit time which represents the fitting time of classifier in the training set is also tabulated along with the other four performance metrics. Classifier performance with respect to feature groups are listed below in the tables.

The performance of classifiers without using feature selection technique (feature group 3) is listed in Table 5. Here all the 7 features are considered for classification. While analysing the results, it is noted that three classifiers Logistic Regression, Support Vector Machine and LDA classifier scores an accuracy above 99%. Logistic Regression classifier achieve highest accuracy of 99.995 %. SVM classifier is able to detect all the malicious traffic and get a 100 % score for recall. Naive Bayes achieves the lowest, but reasonably good accuracy score of 97.71%.

Table 5 - Performance of Classifiers on Feature Group 3 with 7 Selected Features (k=7)

Algorithm	Accuracy%	Recall%	Precision%	F1 score%	Fit Time
Logistic Regression	99.995	99.995	99.997	99.996	0.735
Naive bayes	97.717	96.523	100	97.895	0.031
LDA	99.695	99.998	99.553	99.772	0.146
k-NN	97.723	96.532	100	97.902	13.179
Random Forest	97.722	96.532	99.999	97.902	0.259
SVM Polynomial	99.872	100	99.809	99.903	5.825

Received: 17.06.2021 – Accepted: 18.07.2021

The classifier performance with feature group 2 is listed in Table 6. Performance of classifiers with feature group 2 with 4 selected features, achieve highest accuracy of 99.73% for SVM classifier. Naive Bayes, k-NN and Random Forest classifiers maintain the same accuracy score with feature group 2 and feature group 3. Reducing the number of features from seven to four does not significantly affect the classifier performance except for Logistic Regression. Also, it is noted that accuracy of Random Forest classifier increases slightly with feature group 2.

Table 6 - Performance of Classifiers on Feature Group 2 with 4 Selected Features (k=4)

Algorithm	**Accuracy%**	**Recall%**	**Precision%**	**F1 score%**	**Fit time**
Logistic Regression	97.617	96.532	99.839	97.822	0.733
Naive bayes	97.717	96.523	100	97.895	0.025
LDA	99.668	99.998	99.514	99.752	0.108
k-NN	97.722	96.532	99.999	97.902	11.473
Random Forest	97.723	96.532	100	97.902	0.104
SVM Polynomial	99.732	99.993	99.614	99.8	1.498

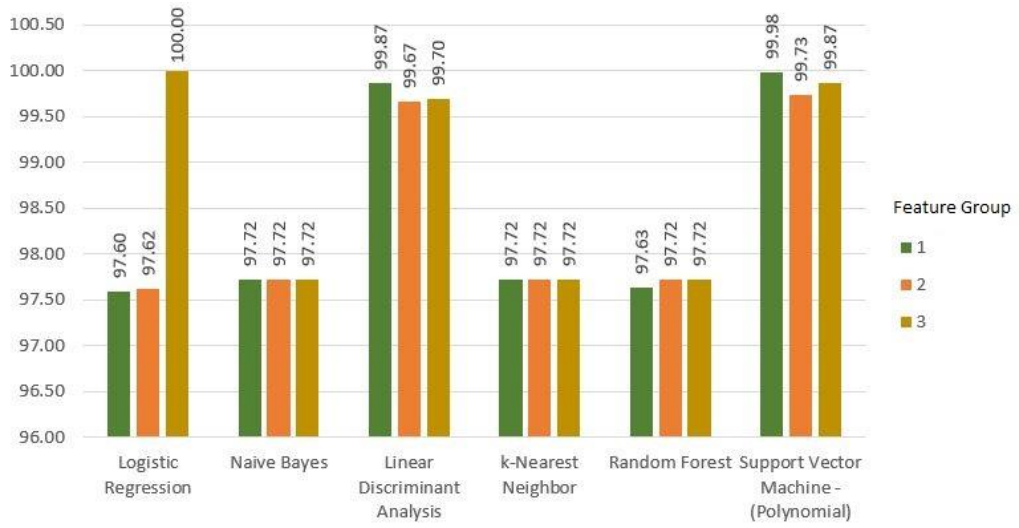
Performance of classifiers with feature group 1 with 2 selected features is listed in Table 7. Here SVM and LDA classifiers achieve highest accuracy of 99.98% and 99.87%. The accuracy of SVM and LDA classifiers get boosted with respect to feature group 2. Naive Bayes and k-NN classifier accuracy remains same for feature group 1 and feature group 2. For all the three feature groups, classifier fitting time is the highest for the k-NN classifier followed by SVM classifier.

Table 7 - Performance of Classifiers on Feature Group 1 with 2 Selected Features (k=2)

Algorithm	**Accuracy%**	**Recall%**	**Precision%**	**F1 score%**	**Fit time**
Logistic Regression	97.595	96.532	99.806	97.806	0.519
Naive bayes	97.717	96.523	100	97.896	0.022
LDA	99.871	99.991	99.816	99.903	0.08
k-NN	97.72	96.532	99.995	97.901	10.467
Random Forest	97.631	96.532	99.861	97.833	0.098
SVM Polynomial	99.981	100	99.971	99.985	2.105

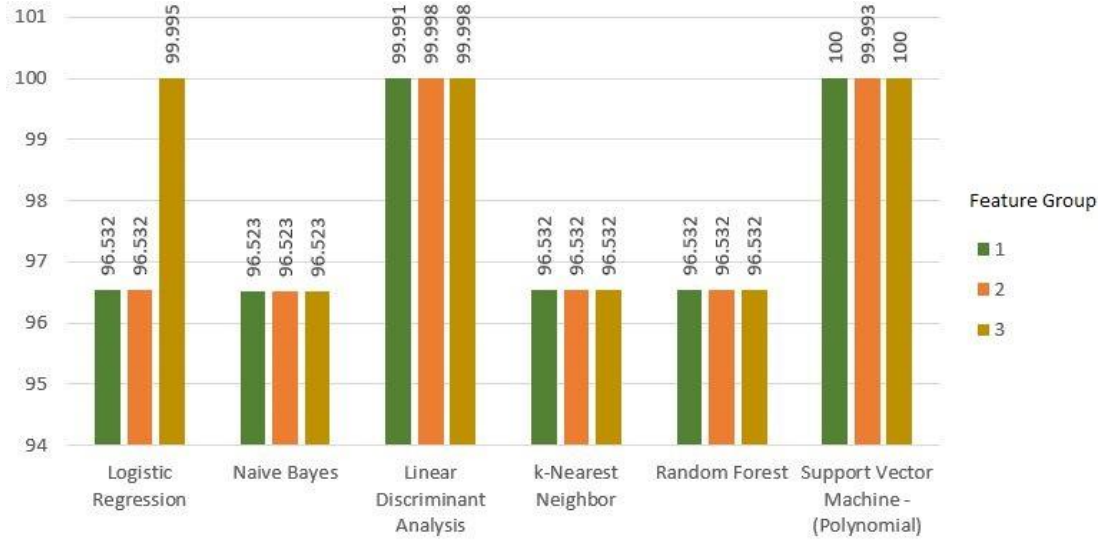
The comparison of classifier accuracy, recall, precision and F1 score for three feature groups are depicted from Figure 4 – Figure 7.

Figure 4 - Accuracy of Classifiers for three Feature Groups



Accuracy of classifiers for the three feature groups is depicted in Figure 4. Accuracy of Naive Bayes classifier and kNN classifier remain constant across three feature groups. Highest accuracy of 99.995% is achieved by Logistic Regression classifier for the feature group 3 with all 7 features. LDA and SVM classifiers scores highest accuracy for feature group 1 with only two features. The average accuracy of algorithms with feature group 1, feature group 2 and feature group 3 are 98.42%, 98.36% and 98.79% respectively. The highest overall classifier accuracy is achieved with feature group 3 with all the seven features.

Figure 5 - Recall of Classifiers for three Feature Groups

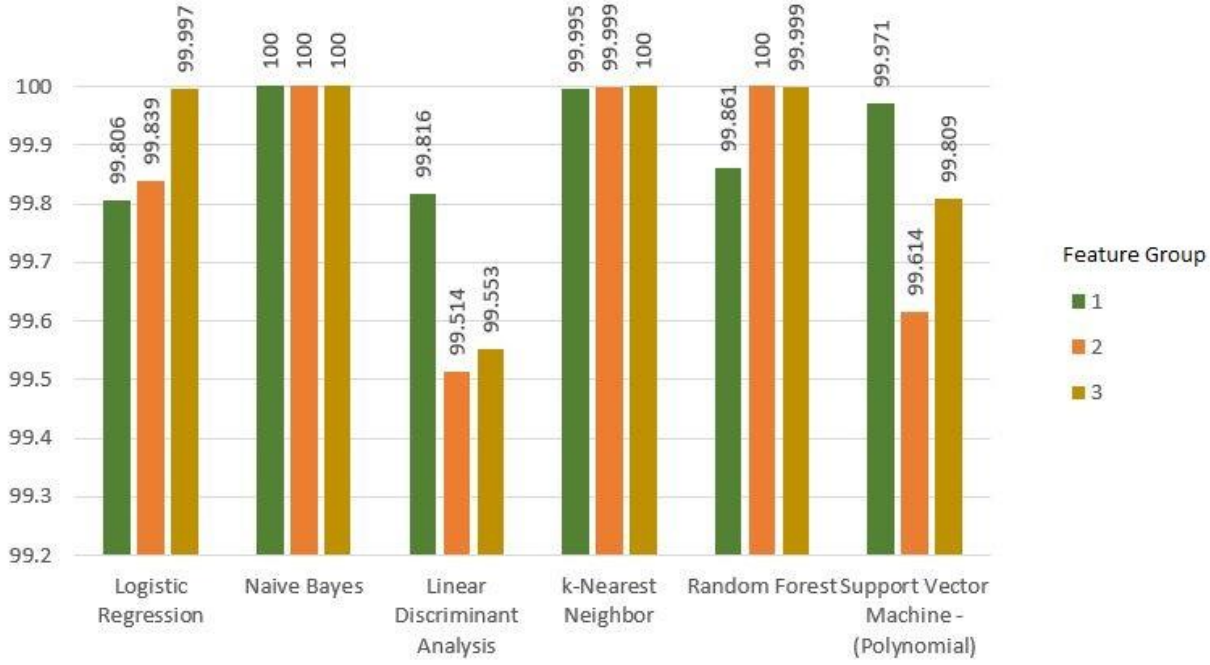


Recall of classifiers for the three feature groups is depicted in Figure 5.

Recall of classifiers Naive Bayes, kNN and Random Forest remains same with three feature groups. Logistic Regression classifier achieves better recall with feature group 3. Recall of LDA and SVM classifiers are almost stable across three feature groups. Highest recall is achieved by SVM classifiers for feature group 1 and feature group 3.

The average recall of algorithms with feature group 1, feature group 2 and feature group 3 are 97.69%, 97.69% and 98.26% respectively. The highest overall recall is achieved with feature group 3.

Figure 6 - Precision of Classifiers for three Feature Groups

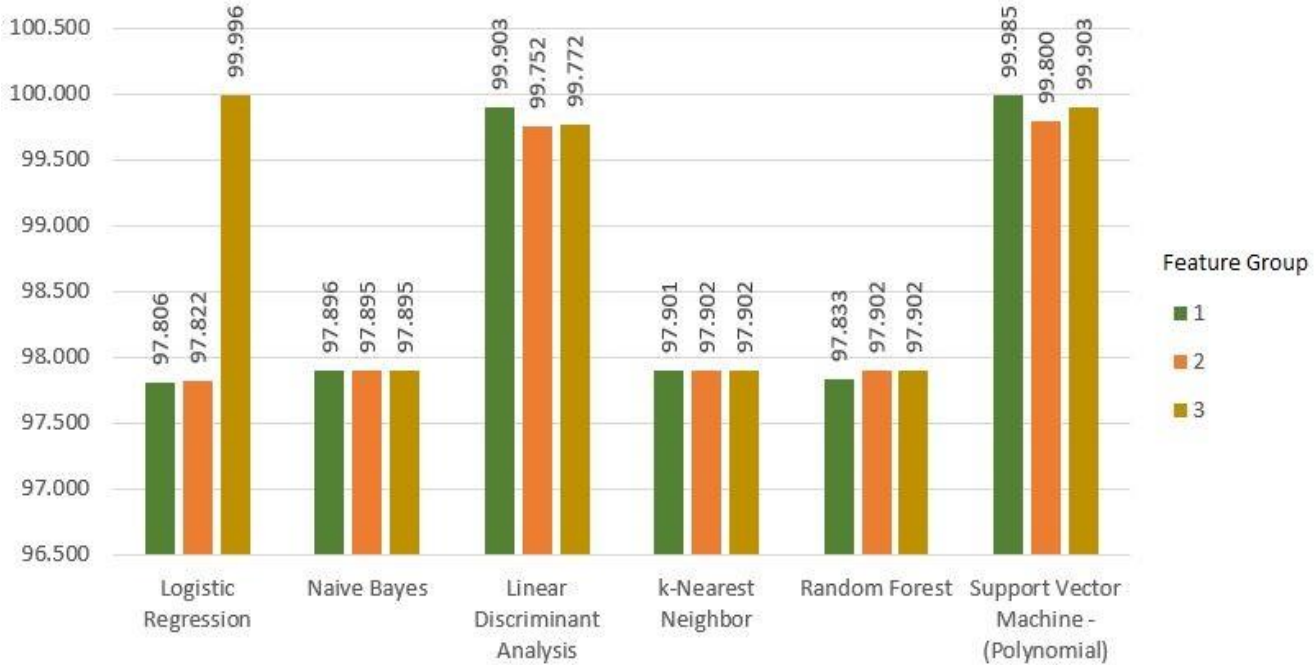


With respect to precision, which is depicted in Figure 6, Naive Bayes classifier scores 100% for all the three feature sets. The recall of k-NN classifier is not affected by feature groups. All the classifiers except LDA and SVM scores better precision with feature group 3. All algorithms are capable of detecting DDoS attacks with high precision. The average precision of algorithms with feature group 1, feature group 2 and feature group 3 are 99.91%, 99.83% and 99.89% respectively. The highest overall recall is achieved with feature group 3.

F1 score of classifiers for the three feature groups is depicted in Figure 7. Naive Bayes classifier, kNN and Random Forest classifier achieves same F1 score with three feature groups. These algorithms are not affected by feature groups. However Logistic Regression classifier achieves highest F1 score with only complete features. Average F1 score for all classifiers for feature group 1, feature group 2,

feature group 3 are 98.55%, 98.51% and 98.90% respectively. The highest overall classifier F1 score is achieved by feature group 3.

Figure 7 - F1 Score for Three Feature Groups



After analyzing the four performance measures attained for the six classifiers for three feature groups, it is noted that highest overall classifier performance is achieved with feature group 3 with all the features. From the experimental evaluation, the SDN flow statistics features collected and used in this study are capable of detecting DDoS attacks with average accuracy of 98.79%, average recall of 98.26%, average precision of 99.89% and average F1 score of 98.90%. The best accuracy score for each feature group is tabulated in Table 8. With all the seven features, Logistic Regression classifier scored the highest accuracy of 99.99%. While performing the classification with 4 features, SVM and LDA classifiers were able to detect attacks with very good accuracy of 99.73% and 99.67% respectively. Further selecting only two important features for classification, SVM and LDA classifiers achieved accuracy score of 99.98% and 99.87% respectively. Among the classifiers SVM and LDA are capable of detecting DDoS attacks - TCP SYN flooding attacks, HTTP request flooding attacks, UDP flooding attacks and ICMP flooding attacks with the two best ranked features – 'Entropy of protocol' and Entropy of source IP address'. These features can be used for building light weight model for first stage classification in multi stage classification systems.

Table 8 - Best Accuracy Score for Feature Groups

S. No	Feature Group	Features in the group	Best classifier accuracy score
1	Feature group 1	Entropy of protocol, Entropy of source IP	99.98% with SVM classifier
2	Feature group 2	Entropy of protocol, Entropy of source IP, Entropy of destination IP, Flow count	99.73 % with SVM classifier
3	Feature group 3	Entropy of protocol, Entropy of source IP, Entropy of destination IP, Flow count, Average duration, Average byte count, Average packet count	99.99 % with Logistic Regression classifier

The classification accuracy obtained for previous works is tabulated in Table 9.

Table 9 - Comparison with Previous Classification Works

S. No	Authors	Dataset used	Classifier accuracy score
1	Kokila et al [9]	2000 DARPA intrusion detection dataset	95.11 %
2	Dayal, Neelam et al [30]	Emulated SDN dataset	99.83 %
3	Ahuja, Nisha, et al [59]	Emulated SDN dataset	98.8 %
4	The current study	Emulated SDN dataset with 2 features	99.98 %

The proposed model with the best 2 features based on feature score calculated by ANOVA -F Test was able to detect the DDoS attacks in SDN with very high accuracy compared to other works. The system used flow-based features from a Mininet simulated network for the detection of DDoS attacks. Features used by Braga [20] namely, growth of ports and percentage of pair wise flows, were not collected in this work, the RYU controller application implemented a Layer 3 match constraint for building flow rules in the switch. The reason for the high accuracy achieved in the experiment can be due to the limitation in the simulation of the normal traffic. The traditional network based datasets like CICIDS2017 [49] can be used for better traffic diversity. Even though packet capture traffic was collected, only flow level features were used in this work. The focus of our future work is to perform packet analysis using the mirrored traffic captured at host H5.

8. Conclusion

Distributed Denial of Service Attack (DDoS) has emerged as a major threat to cyber space. Though the advent of Software Defined Networking (SDN) makes a network easy to be managed even

SDN is vulnerable to DDoS attacks. A wide range of techniques have been used in conventional networks to detect and mitigate DDoS attacks. In this work, flow features obtained from the switches were considered for detecting DDoS attack. The OpenFlow enabled SDN allows for collecting the flow level features which can be used for obtaining derived features. The DDoS attack classification was performed with the dataset collected from a Mininet emulated network. From the experimental evaluation, it is noted that features used for the study are capable of detecting DDoS attacks with high accuracy. With all the seven features of feature group 3, accuracy score of 99.99% was obtained. These seven features are capable of detecting DDoS attacks in SDN environment. The best two features for detecting DDoS attacks in SDN environment based on ANOVA F-Test were found to be 'Entropy of protocol and Entropy of source IP address'. By using these two features ML classifier SVM and LDA were able to detect the attack traffic with an accuracy of 99.98% and 99.87% respectively. These features can be used for building light weight model in multistage classification systems. At the same time, in this work attacks were launched individually. While launching multiple attacks at the same time, entropy of protocol may not decrease. The selection of best features for classification in such cases has to be studied further. The work can further be explored using standard datasets ISCX2012, CIC2017 and CSECIC2018. Detailed analysis can also be performed on the captured packets for detecting multi vector attacks. The work can be extended to a multi class approach to classify the attack into various types like ICMP, UDP, TCP and their combinations. Deep learning-based classification techniques can also be attempted for efficient detection of DDoS attacks.

References

Kreutz, Diego, et al. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE 103.1*(2014): 14-76.

H. Kim and N. Feamster, Improving network management with software defined networking, *Communications Magazine, IEEE*, 51(2), 114–119, 2013.

Bhushan, Kriti, and Brij B. Gupta. Distributed denial of service (DDoS) attack mitigation in software defined network (SDN)-based cloud computing environment. *Journal of Ambient Intelligence and Humanized Computing 10.5* (2019): 1985-1997.

Y. Jarraya, T. Madi, and M. Debbabi, A survey and a layered taxonomy of software-defined networking, *Communications Surveys Tutorials, IEEE*, 99, 1–1, 2014.

Wang, Rui, Zhiping Jia, and Lei Ju., An entropy-based distributed DDoS detection mechanism in software-defined networking. *Trust-com/BigDataSE/ISPA, 2015 IEEE*. 1. IEEE, 2015.

Kalkan, Kbra. JESS: Joint Entropy-Based DDoS Defense Scheme in SDN, *IEEE Journal on Selected Areas in Communications 36.10* (2018): 2358-2372.

"Netflow", http://cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html.

InMon corp. (2020), “sflow-rt”, http://sflow-rt.com/, Last accessed 2021/05/03

Kokila, R.T., S. Thamarai Selvi, and Kannan Govindarajan. DDoS detection and analysis in SDN-based environment using support vector machine classifier. *2014 Sixth International Conference on Advanced Computing (ICoAC). IEEE*, 2014.

Ye, Jin. A DDoS Attack Detection Method Based on SVM in Software Defined Network. *Security and Communication Networks* (2018).

Park, Younghee, Nikhil Vijayakumar Kengalahalli, and Sang-Yoon Chang. Distributed Security Network Functions against Botnet Attacks in Software-defined Networks. *2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN). IEEE*, 2018.

Dimolianis, Marinos. Mitigation of Multi-vector Network Attacks via Orchestration of Distributed Rule Placement. 2019 *IFIP/IEEE Symposium on Integrated Network and Service Management (IM). IEEE,* 2019.

Singh, Kulvinder, and Ajit Singh. Memcached DDoS Exploits: Operations, Vulnerabilities, Preventions and Mitigations. *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS). IEEE*, 2018.

Kolias, Constantinos. DDoS in the IoT: Mirai and other botnets. *Computer 50.7* (2017): 80-84.

Kapersky’s DDoS Attack Reports, https://securelist.com/ddos-report-q4-2019/96154/, https://securelist.com/ddos-attacks-in-q4-2020/100650/

Bawany, Narmeen Zakaria, Jawwad A. Shamsi, and Khaled Salah. DDoS attack detection and mitigation using SDN: methods, practices, and solutions. *Arabian Journal for Science and Engineering 42.2*(2017): 425-441.

Dayal, Neelam, and Shashank Srivastava, Analyzing behavior of DDoS attacks to identify DDoS detection features in SDN. *2017 9th International Conference on Communication Systems and Networks (COMSNETS). IEEE*, 2017.

Gkountis, Christos, et al. Lightweight algorithm for protecting SDN controller against DDoS attacks. *2017 10th IFIP Wireless and Mobile Networking Conference (WMNC).* IEEE, 2017.

Peng, Tao, Christopher Leckie, and Kotagiri Ramamohanarao. Protection from distributed denial of service attacks using history-based IP filtering. *IEEE International Conference on Communications, 2003.* ICC' 03. 1, IEEE, 2003.

Braga, Rodrigo, Edjard Mota, and Alexandre Passito. Lightweight DDoS flooding attack detection using NOX/OpenFlow. *IEEE Local Computer Network Conference. IEEE,* 2010.

B. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, A survey of software-defined networking: Past, present, and future of programmable networks, *Communications Surveys Tutorials, IEEE, 16*(3), 1617–1634, Third 2014.

OpenFlow specification. https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-spec-v1.3.2.pdf Last Accessed 2021/05/03

Phan, Trung V. OpenFlowSIA: An optimized protection scheme for software-defined networks from flooding attacks. Communications and Electronics (ICCE), *2016 IEEE Sixth International Conference on. IEEE*, 2016.

Han, Biao. OverWatch: A cross-plane DDoS attack defense framework with collaborative intelligence in SDN. *Security and Communication Networks* (2018).

Rahman, Obaid, Mohammad Ali Gauhar Quraishi, and Chung-Horng Lung. DDoS attacks detection and mitigation in SDN using machine learning. *2019 IEEE World Congress on Services (SERVICES)*, 2642. IEEE, 2019.

Wang, Wentao, Xuan Ke, and Lingxia Wang. A HMM-R Approach to Detect L-DDoS Attack Adaptively on SDN Controller. *Future Internet 10.9* (2018): 83.

Barki, Lohit. Detection of distributed denial of service attacks in software defined networks. 2016 *International Conference on Advances in Computing, Communications, and Informatics (ICACCI). IEEE*, 2016.

Hsieh, Chang-Jung, and Ting-Yuan Chan. Detection DDoS attacks based on neural-network using Apache Spark. *2016 International Conference on Applied System Innovation (ICASI). IEEE*, 2016.

Chen, Zhuo. XGBoost classifier for DDoS attack detection and analysis in SDN-based cloud. *2018 IEEE International Conference on Big Data and Smart Computing (BigComp). IEEE*, 2018.

Dayal, Neelam, and Shashank Srivastava. An RBF-PSO based approach for early detection of DDoS attacks in SDN. *2018 10th International Conference on Communication Systems & Networks (COMSNETS). IEEE*, 2018.

Dehkordi, Afsaneh Banitalebi, Mohammad Reza Soltanaghaei, and Farsad Zamani Boroujeni, The DDoS attacks detection through machine learning and statistical methods in SDN. *The Journal of Supercomputing 77.3*(2021): 2383-2415.

Tuan, Nguyen Ngoc. A DDoS attack mitigation scheme in ISP networks using machine learning based on SDN. *Electronics 9.3* (2020): 413.

Sahoo, Kshira Sagar. An evolutionary SVM model for DDOS attack detection in software defined networks, *IEEE Access 8* (2020): 132502-132513.

Perez-Diaz, Jesus Arturo. A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning, *IEEE Access* 8 (2020): 155859-155872.

Sen, Sajib, Kishor Datta Gupta, and Md Manjurul Ahsan., Leveraging machine learning approach to setup software-defined network (SDN) controller rules during DDoS attack. *Proceedings of International Joint Conference on Computational Intelligence.* Springer, Singapore, 2020.

Polat, Huseyin, Onur Polat, and Aydin Cetin, Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models. *Sustainability 12.3* (2020): 1035.

Niyaz, Quamar, Weiqing Sun, and Ahmad Y. Javaid. A deep learning based DDoS detection system in software-defined networking (SDN). arXiv preprint arXiv:1611.07400 (2016).

Li, Chuanhuang. Detection and defense of DDoS attack–based on deep learning in OpenFlow-based SDN. *International Journal of Communication Systems 31.5*(2018): e3497.

Imamverdiyev, Yadigar, and Fargana Abdullayeva. Deep learning method for denial of service attack detection based on restricted boltzmann machine. *Big Data 6.2*(2018): 159-169.

Yuan, Xiaoyong, Chuanhuang Li, and Xiaolin Li. Deep Defense: identifying DDoS attack via deep learning. *2017 IEEE International Conference on Smart Computing (SMARTCOMP). IEEE*, 2017.

Doriguzzi-Corinα, R. LUCID: A Practical, Lightweight Deep Learning Solution for DDoS Attack Detection. (2019).

Canadian Institute for Cybersecurity datasets, https://www.unb.ca/cic/datasets/, last accessed 2021/04/24.

Zhu, Liehuang. Privacy-preserving DDoS attack detection using cross-domain traffic in software defined networks. *IEEE Journal on Selected Areas in Communications 36.3*(2018): 628-643.

Lantz, Bob, Brandon Heller, and Nick McKeown. A network in a laptop: rapid prototyping for software-defined networks. *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks.* 2010.

https://ryu.readthedocs.io/en/latest/index.html, Last accessed on 2021/05/03

Avallone, Stefano. D-ITG distributed internet traffic generator. *First International Conference on the Quantitative Evaluation of Systems*, 2004. QEST 2004. Proceedings. IEEE, 2004.

Sanfilippo, Salvatore. "hping3 (8)-linux man page." https://linux. die.net/man/8/hping3 (2005).

BoNeSi, *"The DDoS Botnet Simulator,"* https://github.com/markus-go/bonesi. Last accessed on 2021/05/03

Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani, Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization, *4th International Conference on Information Systems Security and Privacy (ICISSP)*, Portugal, January 2018

Yang, Lingfeng, and Hui Zhao. DDoS attack identification and defense using SDN based on machine learning method. *2018 15th International Symposium on Pervasive Systems, Algorithms and Networks (I-SPAN).* IEEE, 2018.

James, Gareth. *An introduction to statistical learning.* 112. New York: springer, 2013.

Saranya, T., "Performance analysis of machine learning algorithms in intrusion detection system: A review." *Procedia Computer Science 171* (2020): 1251-1260.

Belouch, Mustapha, Salah El Hadaj, and Mohamed Idhammad. "Performance evaluation of intrusion detection based on machine learning using Apache Spark." *Procedia Computer Science 127*(2018): 1-6.

Belavagi, Manjula C., and Balachandra Muniyal. "Performance evaluation of supervised machine learning algorithms for intrusion detection." *Procedia Computer Science 89*(2016): 117-123.

Bhardwaj, Aayush. "Classification of human emotions from EEG signals using SVM and LDA Classifiers." *2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN). IEEE*, 2015.

Xie, Junfeng. "A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges." *IEEE Communications Surveys & Tutorials 21.1*(2018): 393-430.

Mishra, Preeti. "A detailed investigation and analysis of using machine learning techniques for intrusion detection." *IEEE Communications Surveys & Tutorials 21.1* (2018): 686-728.

Shakeela, Shaikh. "Optimal Ensemble Learning Based on Distinctive Feature Selection by Univariate ANOVA-F Statistics for IDS." *International Journal of Electronics and Telecommunications 67.2* (2021): 267-275.

Ahuja, Nisha. "Automated DDOS attack detection in software defined networking." *Journal of Network and Computer Applications (2021):* 103108.

Authors

Ancy Sherin Jose is pursuing research under division of Computer Science Engineering, Cochin university of Science and Technology. She is a B.Tech, M.Tech holder in Computer Science. Her research areas are SDN, Network Security, Machine Learning, Deep Learning and Big Data Analytics. She has published papers in the network security domain.

Latha R Nair is working as Associate Professor in the Division of Computer Engineering, Cochin University of Science and Technology. She is a B. Tech, M. Tech and Ph.D. holder in Computer Science. She has published a number of papers in the areas of machine intelligence and natural language processing. She has done extensive research in Malayalam language computing. Her areas of interest are machine intelligence, natural language processing and image processing.

Varghese Paul is working as Post Graduate Professor in Computer Science and Engineering Department, in Rajagiri School of Engineering and Technology. He is a BSc. MTech, Ph.D. holder in Computer Science. His research areas are Data security using Cryptography, Data Compression, Data Mining, Image Processing and E_Governance. He is the developer of TDMRC Coding System for character representation and encryption system using this coding system. He has got many research publications in international as well as national journals. He is a certified Software Test Manager, Ministry of Information Technology, Government of India. Also, member of Information System Audit and Control Association USA and Indian Society for Technical Education, India.