SGuard: machine learning-based Distrbuted Denial-of-Service Detection Scheme for Software Defined Network

Shimaa Ezzat Kotb
Shoubra faculty of Engineering
Benha University
Cairo,Egypt
shaimaa.ezzat@feng.bu.edu.eg

Heba .A Tag El-Dien
Shoubra faculty of Engineering
Benha University
Cairo,Egypt
Hebaallah.shahat@feng.bu.edu.eg

Adly S.Tag Eldien
Shoubra faculty of Engineering
Benha University
Cairo,Egypt
Adlytag@feng.bu.edu.eg

2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC) | 978-1-6654-1243-8/20/$31.00 ©2021 IEEE | DOI: 10.1109/MIUCC52538.2021.9447680

Abstract— **A Software Defined Networking (SDN) is an advanced network design that presents central control for a complete network. It is a dynamic, easy-to-manage, cost-efficient, and adaptive advanced architecture, making it utilitarian for dynamic nature and high-bandwidth of the present applications. Distributed Denial-of-Service (DDoS) attacks specific to SDN networks to deplete the control plane bandwidth and overload the buffer memory of OpenFlow switch.**

In this research, a design and implementation of secure guard to assist in solving the issue of DDoS attacks on pox controller is presented, this guard is named SGuard. A novel Five-tuple as feature vector is utilized for classifying traffic flow using Support Vector Machine (SVM). A Mininet is utilized to evaluate SGuard in a software environment. The introduced system is evaluated by measuring the system's performance in terms of delay, bandwidth, traffic flow and accuracy.

Keywords— ***Software Defined Networking (SDN), SGuard, Distributed Denial of Service attack (DDoS attack), Support Vector Machine (SVM).***

I. INTRODUCTION

Software Defined Networking (SDN) is a modern paradigm of centralized network architecture that dramatically changes the traditional network architecture and overcomes its limitations, which isolates forward and processing planes. The control plane makes traffic decisions, while the data plane performs these decisions. The SDN capabilities assist in solving many security issues in a traditional network and provide the ability to control network traffic to an accurate level [1]. The centralized logic control provided by controllers facilitates network management and diminishes the process overhead as it dumped the complex network control functions into the logic central controllers while the data plane tends to become a collection of forwarding devices dumb [2], [3].

In the SDN, switches don't process incoming packets, they only search their forwarding tables for a match, and if a mismatch exists, it transmitted the packets to the controller for processing [4]. The controller is the essential component of it, and it is responsible for the wide state views of the network in addition to implementing the forwarding decisions. Therefore, Controller protection and safety are necessary and the basis for SDN [3]. The connection between the switches and the controller is via OpenFlow, which is characterized by being a standard and open source protocol [5].

Nowadays, SDN research is developing rapidly, and numerous organizations plan to utilize it in future networks. Basic functions of SDN architecture that not exist in the current network help to improve network security such as central network monitoring, policy control, and security centralization. These distinctive features make it one of the best effective systems in the development of security of network [6].

One of the enormous challenges facing "SDN" is security issue because different attacks can influence performance. The DDoS attacks are danger threat affect the safety of networks facing the Internet. Revealing these attacks precisely and rapidly is a master research subject in security. Network bandwidth, application and system resources are the most targeted things for network attackers. DDoS attacks show the increased attack scale; the attack mode is smarter [7]. The successful attack of DDoS may lead to disrupt the entire network service as a result of consumption of the central control unit or memory of SDN controller then halt the network's operation [5].

This research is about discovering DDoS attack in the SDN using SVM of traffic sorting to normal and abnormal. The roadmap for this research is formatted as follows: In Section II, a number of related works are presented. In section III, a proposed detection and mitigation method is applied to the suggested system. The simulation and the analysis of the result are discussed in section IV. Finally, conclusion and future work in section V.

II. RELATED WORK

At present, several SDN security modules, devices, and middle boxes are being deployed to improve network security; however there are only a handful of studies on protecting SDNs from malicious applications such as SDN rootkits. Wang et al. [2] suggested NSV-GUARD to create secure SDN routing paths dependent on Network Security Virtualization (NSV) and the trust of network nodes. Tao Wang et al. [5] implemented SGuard to ease the DoS attacks in the OpenFlow networks. It can control arrival to the networks to block from spoofing attacks, classify and protect from malicious traffic, which employ 6-tuple as feature vector to categorize traffic flows. Tatang et al. [6] introduced SDN-guard for protecting controllers against rootkits. It's thought depends on a dual-view comparison that can reveal malicious networks and alleviating rootkits, regardless of the tool used to ruin controllers. Jin Ye et al. [7] the introduced strategy to classify the traffic of SDN, which extracted the 6-tuple characteristic values and then

978-1-6654-1243-8/21/$31.00 ©2021 IEEE

employed the SVM algorithm to reveal the DDoS attack traffic. Nicholas Gray et al. [8] offer a new way to authenticate with device fingerprints to improve security. This methodology determined a group of features dependent on static and dynamic attributes of OpenFlow-enabled switches to distinguish various products. Lobna Dridi et al. [9] proposed SDN-Guard to relieve DDoS attack by dynamically managing flow routes, rule entry timeouts and the overall flow rule entries dependent on the potential of a threat flow given by Intrusion Detection System (IDS). YANG WANG et al. [10] propositioned Safe-Guard Scheme (SGS) to defend control plane from DDoS attacks. SGS introduces a flow monitoring that extracts a 4-tuple vector from the flows of switches dependent on rate feature and asymmetry feature and distinguishes normal flows, then activates the controller remapping to execute dynamic defense. Shin et al. [11] propositioned scheme to solve the SYN Flood attack challenge which comprises two modules: Connection Migration and Actuating Triggers. It effectively protects against TCP based saturation attacks, yet it doesn't work with different DoS attacks in SDN. Myo Myint et al. [12] presented a framework for DDoS attacks revelation through the utilization of Advanced Support Vector Machine (ASVM) technique. It can reveal two kinds of flood-based DDoS attacks and minimize the training time as well as the testing time. Christian Rôpke et al. [13] focused on rootkit technologies that enable attackers to ruin the operation of the network system and using open-source controller OpenDaylight to introduce SDN rootkit for the industry's.

III. Detection of DDoS Attack on SDN Based on Support Vector Machine (SVM)

The SDN controller is responsible for sending and managing the forward decision, as well as collecting traffic information from switches. Each switch has a flow table, which is the fundamental data structure for controlling the management of the forwarding policy. Every flow table comprises of numerous flow entries, which compose the rules for data forwarding [14]. An SDN manages network traffic by looking at flow table entries. The detection attack schema for our presented system mainly made of flow generation, flow data collection, features extraction, classifier, and mitigation, as demonstrated in Fig. 1. Initially, both attack flow and normal flow are generated. The suggested security system can deal with two main kinds of DDoS attacks: TCP SYN Flooding and IP Spoofing.

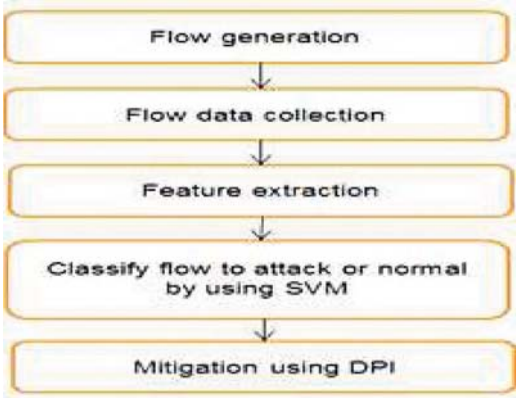

Fig. 1. Five modules of the proposed system.

The request of flow table is sent to switch by the flow data collection and the replay sent from the switch to the flow state collection. Features extraction is responsible for calculating features value from the switch flow table and creating a matrix of five-tuple containing these values.

This research nominates a classification learning method based on the supporting vector machine algorithm (SVM); a learning strategy subject to the statistical learning theory. It enables us to get a better classification between attack traffic and normal traffic.

The SVM draws the data as points in an n-dimensional space where n represents the count of features, then finds a hyperplane to distinguish between the two categories. Choosing the correct hyperplane is the way to design an effective system. The ideal SVM should produce a hyperplane that clearly recognizes cases in two non-overlapping classes. Practically, there are some errors in the classification mechanism and hence, SVM attempts to maximize the margins by finding a suitable hyperplane [1], [7].

We can recognize DDoS attack with appropriate classification results and accuracy by applying SVM. It can explain its fundamental idea by the two-dimensional. There are n points of data $D = \{(X_1, y_1), (X_2, y_2) \ldots (X_n, y_n)\}$, where X_i denotes the attribute vector of the data set and y_i is the associated class label. y_i takes value +1 or −1 ($y_i \in \{+1, -1\}$. In linear SVM, it can draw a straight line to isolate the vector of class +1 from the vector of class −1. The straight line can express by the equation $\omega \cdot x + b = 0$; ω is the weight vector and b is the scalar, called the bias. The points up the separation hyperplane are contented as in (1).

$$\omega \cdot x + b > 0 \quad (1)$$

Also, the points down the separation hyperplane are contented as in (2).

$$\omega \cdot x + b < 0 \quad (2)$$

The hyperplane of the class $y_i = 1$ above the straight line is indicated as: $\omega \cdot x + b \geq 1$, and another hyperplane of the class $y_i = -1$ below the straight line is indicated as: $\cdot x + b \leq 1$. When the data set is separated linearly, these two hyperplanes can be viewed as parallel and the distance between them must be as wide as possible and it determined, as in (3).

$$\text{Distance between two hyperplanes} = \frac{2}{\|\omega\|} \quad (3)$$

For n-dimensional space, the SVM find the optimal hyperplane to solving the optimization problem the formula is expressed, as in (4).

$$\text{Min } \frac{\|\omega\|^2}{2} + C \sum_i^n \xi_i \quad (4)$$

subject to $yi\,(\omega \cdot x + b) + \xi_i \geq 1, \xi_i \geq 0$

where, ξ_i is the slack variable which helps to gauge the distance of the point to its marginal hyperplane with less classification error (C) is the penalty parameter. The optimal value of C can alter the margin since at a major value it gives smaller-margin [11, 8].

A. Flow Status Collection:

The information inside flow tables is gathered in a network of SDN, principally through the OpenFlow protocol. The switch reacts to the message request OpenFlow, which is sent periodically by the pox controller and run "sudo ovs-ofct1 dump-flow s1" Utilize this

command to gather traffic flow data information for the flow table.

B. Extract the Characteristic Values:

When a network is under a DDoS attack, an enormous count of fake source IP addresses with a fixed size of packets are transmitted to attack the target. The attack flow can be detected by observing and analyzing the characteristic values information of the flow table. These characterizations are gathered as training and testing features for our SVM. The following five-tuple characteristic values for DDoS attacks were obtained to detect this attack.

(1) The speed of source IP (SSIP): the numeral of source IPs per unit of time, as in (5).

$$\mathrm{SSIP} = \frac{\sum \mathrm{IP_{src}}}{\mathrm{T}} \tag{5}$$

where $\sum IP_{src}$ the count of source IPs and T represent the sampling interval. With an attack, the random spoofing of transmitting data packet will generate a big count of attacks, the count of the source IP address will increase rapidly.

(2) The Standard Deviation of Flow Packets (SDFP): the standard deviation of the count of packets in the T period, as seen in (6).

$$SDFP = \sqrt{\frac{1}{N_f}\sum_{i=1}^{N_f}(P_i - MP)^2} \tag{6}$$

where P_i is the count of packets in the i^{th} flow and MP is the average count of packets in time T. N_f is the total count of flow entries per period. In event of an attack, this feature has a high correlation whereas the attacker transmits an enormous count of attack data packets of small size and this will be a smaller standard deviation than normal data packets.

(3) The Standard Deviation of Flow Bytes (SDFB): the standard deviation of the quantity of bytes in the T period, as in (7).

$$\mathrm{SDFB} = \sqrt{\frac{1}{\mathrm{N_B}}\sum_{\mathrm{i=1}}^{\mathrm{N_B}}(\mathrm{B_i} - \mathrm{MB})^2} \tag{7}$$

where B_i is the count of bytes in i^{th} flow and MB is the mean count of bytes in time T. Like SDFP, SDFB also has a high attachment with the event of a DDoS attack, and the predictable value of this feature is lower in attack than in normal traffic.

(4) The speed of flow entries (SFE): the numeral of flow entries to the switch during specific time, as seen in (8).

$$\mathrm{SFE} = \frac{\mathrm{N}}{\mathrm{T}} \tag{8}$$

This parameter is closely related to attack detection because the quantity of flows per unit time entries dramatically increased in a fixed period of time in event of an attack compared to the value of SFE in times of normal traffic flows.

(5) The Ratio of Pair-Flow (RPF): is the outcome of dividing the amount of interactive flow entries in the switch by the total count of flows in the period T, as in (9).

$$\mathrm{RFIP} = \frac{\mathrm{IntIP}}{\mathrm{N_{ip}}} \tag{9}$$

where IntIP the total count of interactive IPs in the flow and N_{ip} is the total count of IPs. Under normal circumstances, the source host transmits a request to the target host to create interactive flows, which are the following conditions. The i^{th} packet flow will have the same source IP as the destination IP of the j^{th} flow and the j^{th} packet flow will have a similar source IP as the destination IP of the $\mathrm{i^{th}}$ packet flow. This constitutes an interactive flow, which won't be the case under DDoS attack. Under attack, the input flow to the destination host in time T increases quickly and the destination host can't respond to them. Thus, there will be a sudden reduces in the count of reactive flows once the attack begins [1], [8].

C. Mitigation using Deep packet inspection:

Deep packet inspection (DPI) is a kind of network packet filtering. It evaluates the data portion and the header of the packet that is being sent through an inspection point. Also employ to decide if packets are detected, categorized, or forwarded that contain specific code or data payloads that were not detected, located, categorized, blocked, or forwarded by traditional packet filtering. Dissimilar to plain packet filtering, deep packet examination goes beyond inspecting packet headers. When an enormous count of phishing IPs send SYN packets, the victim's host's resources become highly occupied with this traffic.

If the Deep Packet Inspection Box (DPI) is aware that the network is under attack, it can check incoming traffic for SYN packets and not allow untrusted IP addresses to establish a TCP connection to the host while the network is under attack. This will allow trusted IP addresses to configure new TCP connections to the host, even in the event of an attack and normal traffic to flow unimpeded. This comes at the cost of restricting unknown, non-malicious IP addresses from establishing a new TCP connection during the attack on the victim host [1], [15].

IV. Experiment and Analysis

In this experiment, the controller (pox) with stock components and OpenFlow switch are implemented utilizing Ubuntu. The experiments are executed on Miniedit; a simple GUI editor for Mininet; to build topology in Fig. 2 and verify the validity of the DDoS attack detection method in SDN environment. In the designed topology, PC1 is the victim target; it receives attack traffic from different PCs. All the normal and attack traffic to PC1 pass through S1. The flow from this switch are monitored for three seconds to acquire parameters for the SVM using a bash script that reads the previous three seconds, finds the relevant features from the flow by calling a processing python script and stores them as a column separated value file. Through the training phase, 10 PCs created the normal traffic. The data generated in this network utilizes the DDoS attack tool hping3 that supported by the POX controller. Hping3 can determine parts of the packet, so users can deftly attack and recognize the target [7]. After generated and collected traffic data, five traffic features are extracted so that the SVM can detect a DDoS attack. Fig. 3 demonstrates that SSIP increases in the attack more than normal. The SDFP values for attack traffic less than it in the normal traffic, as demonstrated in Fig. 4. Fig. 5 shows that the values of SDFB reduce from the normal traffic in case of attack traffic. With an attack, the count of entries flow per unit time will be increased

dramatically, so the attack SFE values increase than it in normal traffic, as illustrated in Fig. 6. Fig. 7 shows that RFIP for normal traffic is greater than it of attack traffic.

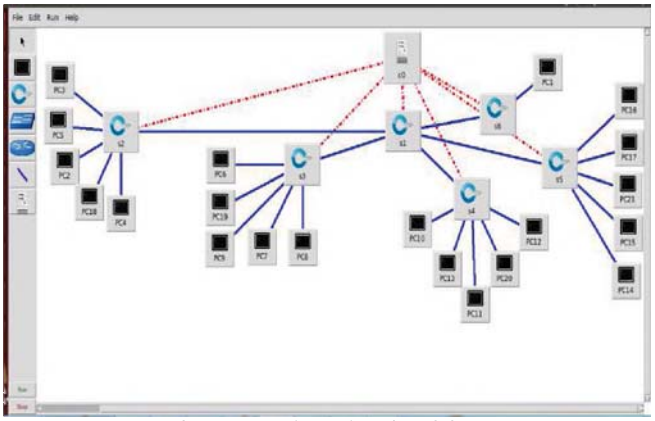

Fig. 2. Network topology in Mininet.

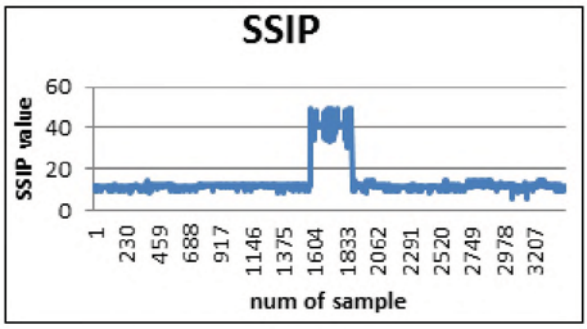

Fig. 3. Speed of Source IP (SSIP) values under normal and attack traffic.

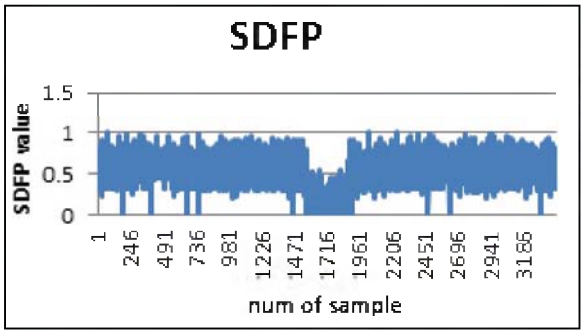

Fig. 4. Standard Deviation of Flow Packets (SDFP) values under normal and attack traffic.

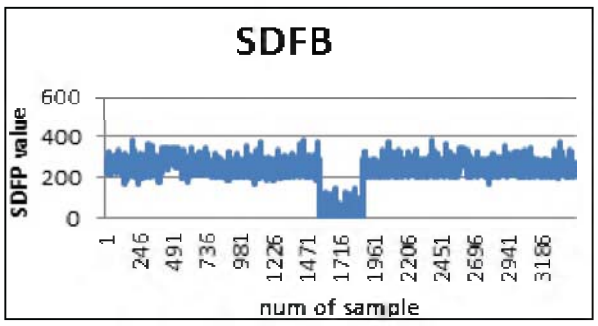

Fig.5. Standard Deviation of Flow Bytes (SDFB) values under normal and attack traffic.

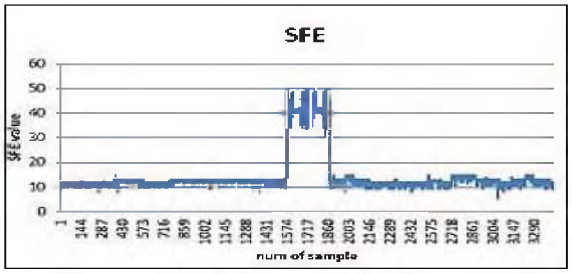

Fig. 6.The speed of Flow Entries (SFE) values under normal and attack traffic.

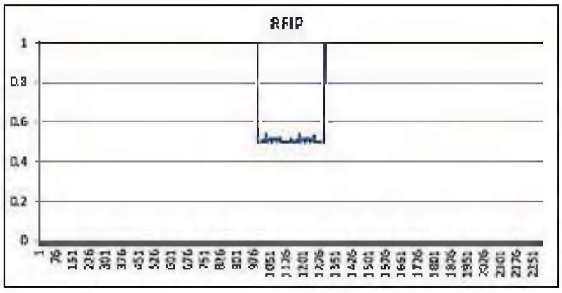

Fig.7. Ratio of Pair-Flow Entries (RFIP) values under normal and attack traffic.

SGuard structure consists of three modules collection, classification and mitigation as exhibited in Fig. 8. It used Switch S6 as a DPI box in the network topology for mitigation. Under normal traffic, output of SVM is "0", S6 has bidirectional flows installed, where permitted the traffic to flow in and out of PC1. If the SVM output is "1", the DPI supposes an attack and starts packet inspection. The source IP addresses of all SYN packets are matched with a pool of trusted IP addresses maintained in this DPI switch. This pool contains IP addresses of sources that on a TCP connection with PC1 for more than time trust seconds when the SVM output was “0”. The SYN requests from other IP addresses are dropped by S6. This results in the dropping of SYN requests from DDoS attackers while legit traffic continues to be routed to PC1 through S6. Fig. 9, shows the shape of the normal traffic in which the count of packets to be sent is kept in range between 10 and 20 for each second.

One characteristic of a DDoS attack is sent more packets to the target and making it impossible for legal hosts, so when there is an attack the count of packets become out of normal range as shown in Fig. 10. To solve this problem, we use SGuard to mitigate fake traffic that decreases the rejected packets as demonstrated in Fig. 11.

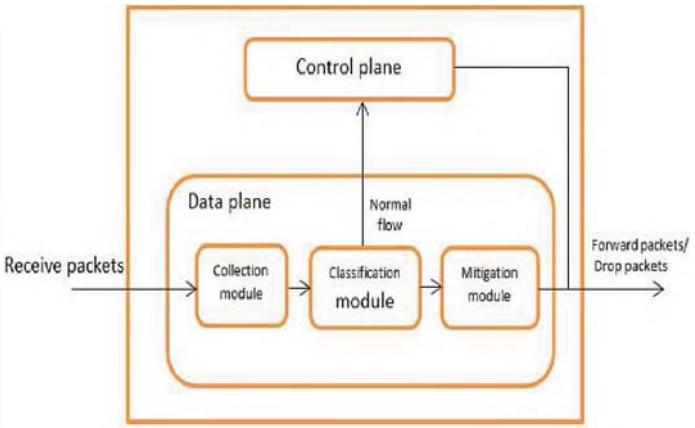

Fig.8. SGuard structure.

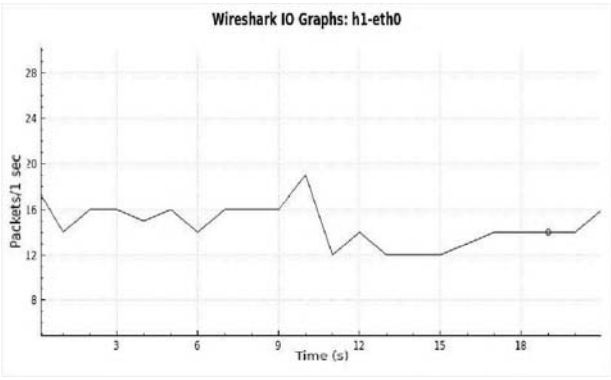

Fig.9. Normal traffic

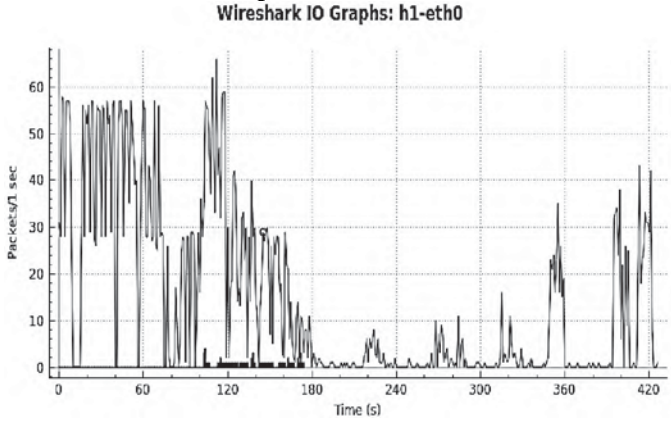

Fig.10. Traffic with attack

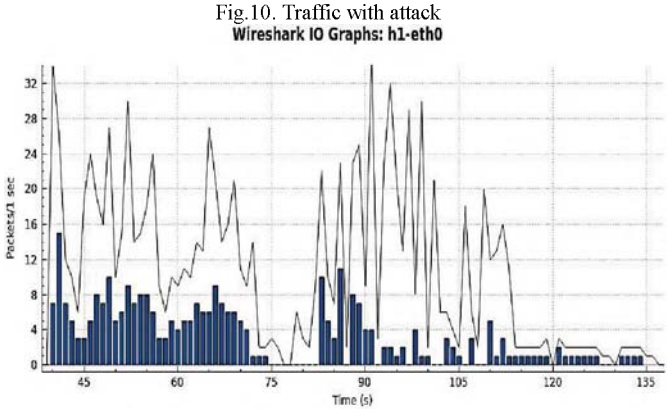

Fig.11. Traffic with guard.

To examine the time delay that the packet consumes to reach PC1, samples of the traffic are taken in the different cases and taking the average value for each case as shown in TABLE I. The average time in attack at the beginning is variable between increase and decrease, and over time, reaching the target for legitimate devices becomes difficult to get, it is one of the characteristics of the attack. Case of using the SGuard; the average time is greater than the normal and less than its value with attack.

To appraise the framework performance, the effect of attack and SGuard on the normal host bandwidth is monitored. Fig. 12 shows the bandwidth of the normal traffic. Fig. 13 demonstrates the bandwidth in case of attack, which is reduced and become zero, meaning that there is no route between the two hosts. The proposed system using SGuard results in a reduced bandwidth as compared to the case of normal traffic, but still better than it in case of attack as demonstrated in Fig. 14.

TABLE I. AVERAGE TIME DELAY.

Case of traffic	Normal	Attack	With SGuard
Time delay	0.332 ms	0.580 ms	0.366 ms

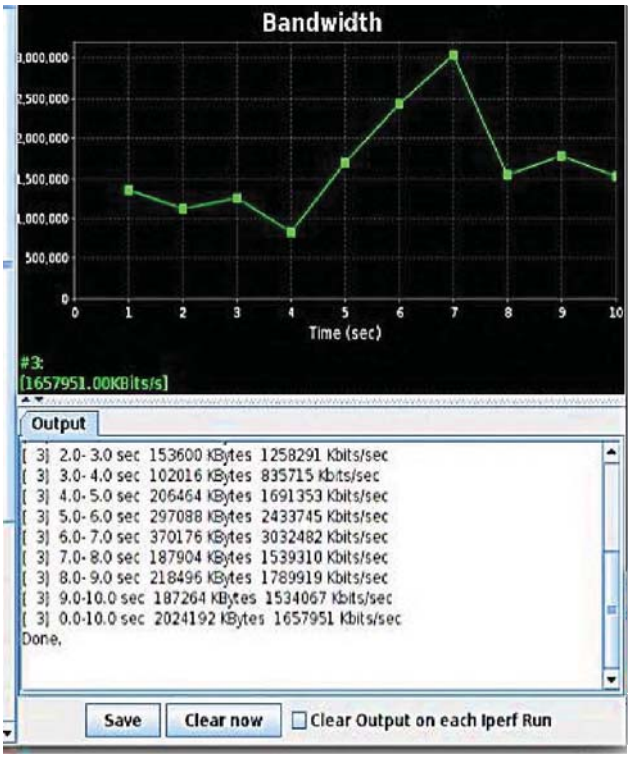

Fig.12. Bandwidth in case of normal traffic on PC9.

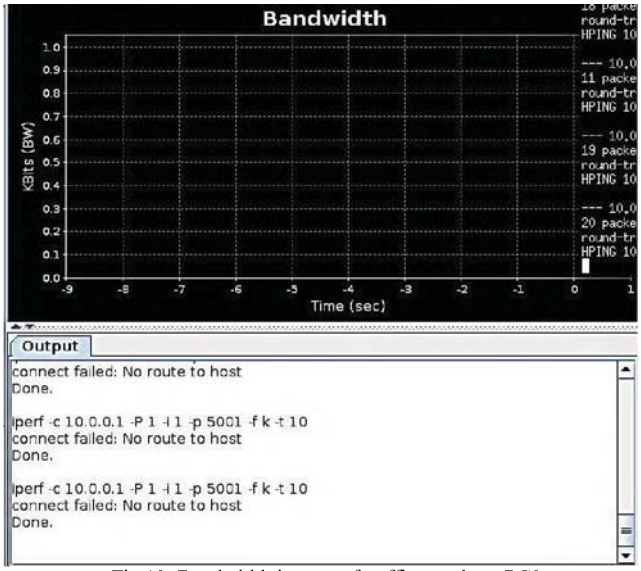

Fig.13. Bandwidth in case of traffic attack on PC9.

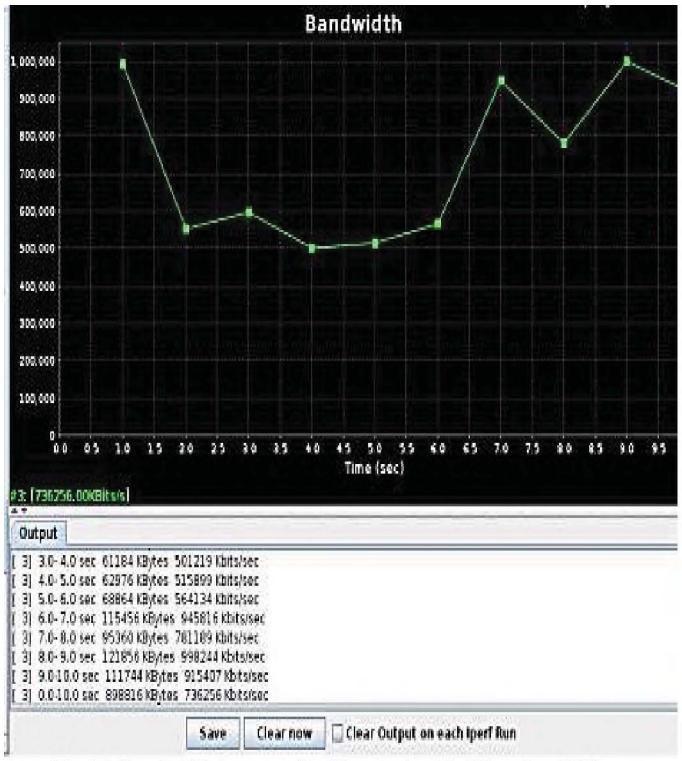

Fig.14. Bandwidth in case of traffic attack with SGuard on PC9.

To estimate system classification performance, we generate a data set which is split into training and testing. Accuracy is the estimation of the system that successfully sorts both normal traffic and anomalous traffic, so we used accuracy for evaluating our detection result, as in (10). We test the accuracy of the system under the different numeral of attacks; also measure the accuracy with different count of sampling.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} * 100\% \qquad (10)$$

True Positive (TP) is the quantum of network traffic that is detected as an attack. False Positive (FP) represents the quantum of network traffic that is incorrectly detected and recognized the normal traffic as the attack traffic. True Negative (TN) represents the count of normal traffic that identified correctly. False Negative (FN) is the amount of attack traffic that is incorrectly detected.

In this research, we trained and tested with a splitting rate from 10% to 90% of collected data set. We simulate different number of attacks and take 200 samples in each simulation. Fig. 15 shows the accuracy of the system with different testing size. It is observed that the accuracy is independent on the number of attacks or testing size and still recording higher values. According to the experimental results demonstrated in Fig. 15, the mean accuracy of the detection can be seen in TABLE II. It is found that the accuracy is inversely proportional to the testing size. The relation between the accuracy and the count of samples from the collected data in case of only one attack is shown in Fig. 16. TABLE III shows the average accuracy rate of the system detection with different amount of flow. The proposed system gives a good value for the detection of DDoS attack in SDN and it has proven that the SVM is one of the best methods of detecting the DDoS attack with high efficiency and accuracy.

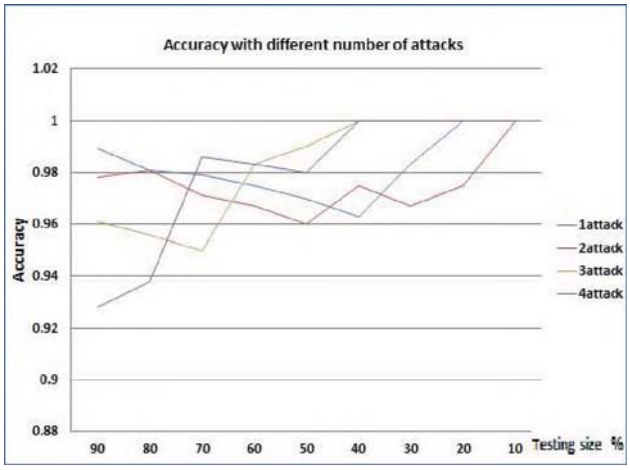

Fig.15. Accuracy with different number of attacks.

TABLE II. AVERAGE ACCURACY WITH DIFFERENT NUMBER OF ATTACKS.

Testing size	Average accuracy %
90%	96.4
80%	96.4
70%	97.1
60%	97.7
50%	97.8
40%	98.4
30%	98.7
20%	99.3
10%	100

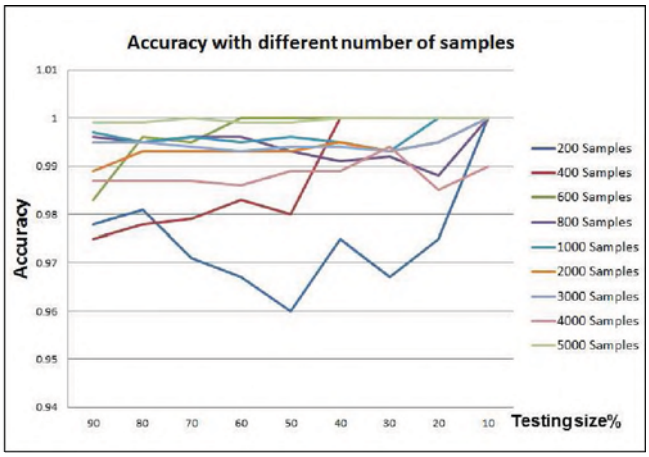

Fig.16. Accuracy with different number of samples.

TABLE III. AVERAGE ACCURACY WITH DIFFERENT NUMBER OF SAMPLES.

number of samples	Average accuracy %
200	97.5
400	97.5
600	97.5
800	99.4
$10\times \mathbf{10^2}$	99.6
$20\times \mathbf{10^2}$	99.3
$30\times \mathbf{10^2}$	99.4
$40\times \mathbf{10^2}$	98.8
$50\times \mathbf{10^2}$	99.9

V. CONCLUSION AND FUTURE WORK

In this research, a method is suggested to detect DDoS attacks using the SVM method. An emulated topology is implemented using Mininet, in which 21 PCs, six switches, and one controller are used, and both normal and malicious traffic data are generated using Hping3. Five-tuple characteristic values are analyzed to differentiate DDoS attack traffic from normal traffic. The traffic is monitored and evaluated to notice the difference between the normal traffic, the presence of an attack and the use of the SGuard. The bandwidth between PC9 and PC1 is measured, and we observed that in the case of the attack over time the communication between the two hosts was interrupted. It also observed that the bandwidth with the SGuard is less than the normal traffic, but the connection is still present between the hosts. The accuracy of the system is measured with different number of attackers and different number of samples. According to the experimental results, the proposed model produces a very high accuracy.

In the future, we wish to expand this work to include newer machine learning techniques to improve performance, for example against other attacks. To give a more hearty assessment of the system, we plan to improve mitigation strategy. The goal of these techniques is to avoid blocking legitimate users when the false positive rate increases.

Improvement the traffic generation, the classifier can understand the metrics better by using real traffic data instead of using traffic generated from simulation tools. The proposed system intends to apply it to different controllers and compare them to find the best for this system. Another proposed improvement is the feature extraction process. Parameters should be extracted and constructed that have a significant correlation with their classification event.

REFERENCES

[1] William Isaac, Suraj Iyer (2018). "SOFTWARE-DEFINED SECURITY" available : https://www.researchgate.net/publication/324716038

[2] Wang, Jie Chen, Jianwei Liu, Jian Mao, and Mengmeng. "NSV-GUARD: constructing secure routing paths in software defined networking." In 2016 IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom)(BDCloud-SocialCom-SustainCom), pp. 293-300. IEEE, 2016.

[3] Wang, Tao, Hongchang Chen, Guozhen Cheng, and Yulin Lu. "SDNManager: A safeguard architecture for SDN DoS attacks based on bandwidth prediction." Security and Communication Networks 2018 (2018).

[4] Wang, Tao, and Hongchang Chen. "SGuard: A lightweight SDN safe-guard architecture for DoS attacks." China Communications 14, no. 6 (2017): 113-125.

[5] Fajar, Andry Putra, and Tito Waluyo Purboyo. "A Survey Research of Distributed Denial-of-Service Attack in Software Defined Networking (SDN)." International Journal of Applied Engineering Research 13, no. 1 (2018): 476-482.

[6] Tatang, Dennis, Florian Quinkert, Joel Frank, Christian Röpke, and Thorsten Holz. "SDN-Guard: Protecting SDN controllers against SDN rootkits." In 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), pp. 297-302. IEEE, 2017.

[7] Ye, Jin, Xiangyang Cheng, Jian Zhu, Luting Feng, and Ling Song. "A DDoS attack detection method based on SVM in software defined network." Security and Communication Networks 2018 (2018).

[8] Gray, Nicholas, Thomas Zinner, and Phuoc Tran-Gia. "Enhancing SDN security by device fingerprinting." In 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), pp. 879-880. IEEE, 2017.

[9] Dridi, Lobna, and Mohamed Faten Zhani. "SDN-guard: DoS attacks mitigation in SDN networks." In 2016 5th IEEE International Conference on Cloud Networking (Cloudnet), pp. 212-217. IEEE, 2016.

[10] Wang, Yang, Tao Hu, Guangming Tang, Jichao Xie, and Jie Lu. "SGS: Safe-Guard Scheme for Protecting Control Plane Against DDoS Attacks in Software-Defined Networking." IEEE Access 7 (2019): 34699-34710.

[11] Shin, Seungwon, Vinod Yegneswaran, Phillip Porras, and Guofei Gu. "Avant-guard: Scalable and vigilant switch flow management in software-defined networks." In Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, pp. 413-424. ACM, 2013.

[12] Myo Myint Oo, Sinchai Kamolphiwong, Thossaporn Kamolphiwong, and Sangsuree Vasupongayya. "Advanced Support Vector Machine-(ASVM-) Based Detection for Distributed Denial of Service (DDoS) Attack on Software Defined Networking (SDN)." Journal of Computer Networks and Communications, vol. 2019, 2019, pp. 1–12.

[13] Röpke and Thorsten Holz. 2015. "SDN Rootkits: Subverting Network Operating Systems of Software-Defined Networks." In RAID 2015 Proceedings of the 18th International Symposium on Research in Attacks, Intrusions, and Defenses - Volume 9404, 339–56.

[14] Khairi, Mutaz HH, Sharifah HS Ariffin, NM Abdul Latiff, A. S. Abdullah, and M. K. Hassan. "A Review of Anomaly Detection Techniques and Distributed Denial of Service (DDoS) on Software Defined Network (SDN)." ENGINEERING TECHNOLOGY & APPLIED SCIENCE RESEARCH 8, no. 2 (2018): 2724-2730.

[15] Hamilton, Robert, Wayne Gray, Clifford Sibanda, Subbiah Kandasamy, Raimund Kirner, and Athanasios Tsokanos. 2020. "Deep Packet Inspection in Firewall Clusters." In 2020 28th Telecommunications Forum (TELFOR).