International Journal of Early Childhood Special Education (INT-JECSE)
DOI:10.9756/INTJECSE/V14I5.424 ISSN: 1308-5581 Vol 14, Issue 05 2022

An Analysis of Machine Learning Depend on Q-MIND for Defencing The Distributed Denial of Service Attack on Software Defined Network

M. Sakthivel
Professor, Dept of CSE,
Sree Vidyanikethan Engineering College,
Tirupati, Andhra Pradesh, India
sakthisalem@gmail.com

S. Sivanantham
Assistant Professor, Dept of CSSE,
Sree Vidyanikethan Engineering College,
Tirupati, Andhra Pradesh, India
sivanantham.s@vidyanikethan.edu

R. Kamalraj
Associate Professor,
School of CS and IT,
Jain deemed to be University,Bangalore
r.kamalraj1981@gmail.com

V. Krishnamoorthy
Assistant Professor, Dept of CSE, Bannari Amman Institute of Technology, Sathyamangalam, Tamilnadu, India
krishnamoorthy@bitsathy.ac.in

Abstract: A flexible and scalable network control was enabled in the Software Define Networking. It introduces unprotected network which can easily exploits by attackers. Especially, Distributed Denial of service attacks or low rate products are attracting researchers recently due to their detecting challenges. This research proposes novel machine learning depend on defence structure named Q-MIND which is used to mitigate and detect stealthy Distributed Denial of Service attacks in Software Denied Network. At first, everyone should examine the adversary design of stealthy Distributed Denial of Service attacks and negligence in Software Defined Network. And next narrates and examines the detection process which uses a RL approach depend on Q-MIND to maximize the performance of detection. And at last, outlines the whole Q-MIND defence structure should incorporate the policy of optimal derived from the agent of Q-Learning to defeat the Distributed Denial of Service attacks in Software Denied Network.

Keywords: QMIND, SDN, distributed denial of service attack, RL, Q-Learning, networks**.**

1. Introduction

SDN became famous in current days because of its various benefits. It provides advantages like creativity, monitoring, flexibility and scalability (Imran et al., 2019). SDN appears as network paradigm to isolates the important network logic from the switching and routing components which processed as packet forwarders(Dharma et al., 2015). An information plane has network components like switches (ON or OFF) which are regulated by the controller in the control plane (Lin et al., 2018). An isolation of the data and control plane plays an important role in giving topmost performance in high speed and large scale computing process. It also gives a network management in a simplified form. And here management and configuration are managed by the controller (Dharma et al., 2015; Imran et al., 2019). The administrators do not require reconfiguring and accessing thousands of systems in the network because of its performance in adjustments and network upgrades. It can simply impose network and policy configuration mainly in real time via the controller. (Tselios et al., 2017).

Here the controller required various services to handle the data plane. It allows exchange information with the services in the process segment to give network function like intrusion detection, routing and load balancing [1]. All the applications and services are executed in the process segment and also mapped to the entire network through network operating system. It was installed on the administrator who provides a high level of optimization, network control and automation. [1]. The application uses various APIs like REST or java Application programming interfaces for remote areas.[3]

.

The security threats have the most devastating impact on the Software Denied Networking which was the Distributed Denial of Service attack. Distributed Denial of Service can overwhelm the overflow switch or administrator when the network was not protected properly. There are a variety of documentations about the protection of Software Denied Network from Distributed Denial of Service attacks. IDSs are used in the network to aware the controller and detect the packets when a Distributed Denial of Service attack was detected. Machine learning was attracted many researchers to detect Distributed Denial of Service attacks. Therefore, defending the software denied network from problems was active in research area. This research aims to consider the suitable machine learning algorithm to find a Distributed Denial of Service attack in a network area.

2. Related work

2.1 Distributed Denial of Service attacks against the administrator:

In this experiment, the control processes are taken from the switch and provided to the controller. This is considered as the network brain in Software Denied Network architecture. Levels of parent rules are applied easily to the network along with the support of controller. The administrator can include the new principles to the transmission system and alter the rules also. This process carry out these alternation in rules by communicating using transmission gadgets and protect a channel by the OF protocol. A unity and continuity of information traffic are obtained by this channel. If the channel damages, the link between the transmission gadgets and controller also breaks.

Architecture of Software Denied Network was the target of distributed denial of service attacks. If the attacker was attacking the software denied network, it has 3 main targets which was shown in the fig. 1 to enhance the controller sources, to fill the table glow in the switch with unimportant flows and finally to acquire the frequency of the channel between the switch and controller.

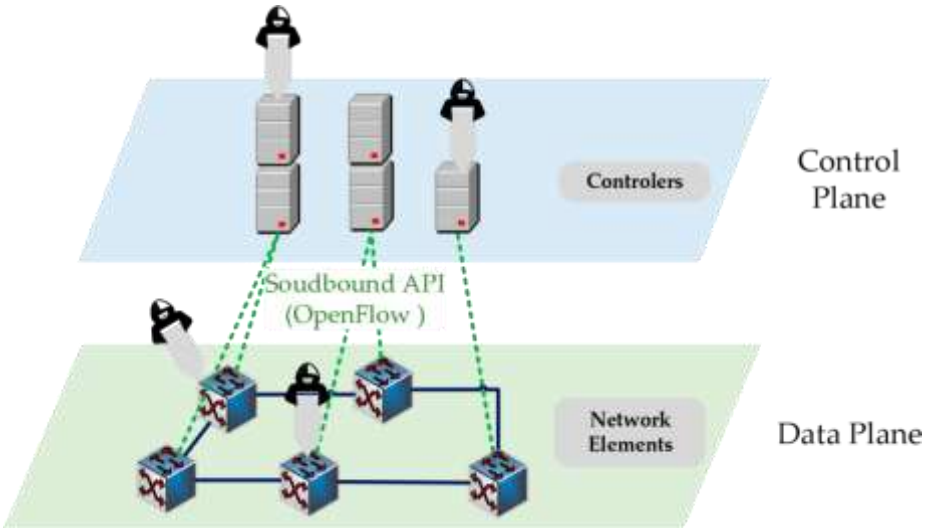

Fig. 1. Important target of DDoS attacks ofn SDN

Alshamrani et al. (Kuerban et al., 2016) accept the process to prevent distributed denial of service attacks which are not very effective. They examined the impact of attacking and latest attacks on software denied network. They collected information traffic using the transmission gadgets on a information plane and later proceed machine learning algorithm to sudden response in traffic altering which acquired in the architecture of Software Denied Network at the time of attack. Vector Machine (SVM), J48, and Naive Bayes (NB) algorithms were employed for classification.

(Alshamrani et al., 2017) analysed distributed denial of service attacks through the measuring rate of packet at the time of attack. If any one of the packet was coming to the administrator which passes the pre-determined issues to everyone. An unit inspection uses the SVM to activate and detect the flooding attack of distributed denial of service

(Latah & Toker, 2018) examined a RNN design to cover the each and every layer of the software denied network framework inorder to block and detect the attacks of distributed denial of service. The examined model was created for the original detection and the restriction of distribution denial of service attacks which has a high level of rate in perfection. This model can disrupt the contemporise work of administrator and discredit the network performance. This process makes the defence less switch against distributed denial of service attacks. Many packets forwards to the overflow switch by severe nodes that are taken into memory of caching and the response from the administrator which adds the data on the overflow orders. This was expected by (W. Li et al., 2016). Packets coming from malicious nodes fill the switch buffer and then packets coming from legal users start to drop (Padmaja & Vetriselvi, 2016)[15].

Ye et al. collected information on traffic network from the gadgets of transmission on the information plane by the response of controller. Features of 6 tuple values are related to the distributed denial of service attacks form the overflow table which was isolated to detect distributed denial of service attacks by SVM. Therefore, the accuracy rate of ICMP test attacks the overflow which was recorded as very low.

Xue et al analysed various requirements regarding security which are required as the software denial network. Here the controller controls various switches in the information plane. This stated that protection should not be succeeding with software and equipment which was adapted to the software denied network architecture..

3. Characteristic selection in ML or AL depend classification algorithm:

To achieve the combination of optimum , the certain characteristic set and particular ML or Al algorithm was used to detect the system. The researcher accepts the finite Markov Decision Process (MDP) approach [13] with *episodic tasks.* The Markov Decision Process structure allows the software AOS to take an immediate action which depend upon the observation to increases it sudden reward in each episode. This kind of reward was analysed in multiple investigation criteria.

The Markov Decision Process was characterized by <S,A,*r*>, where S is the state space, A is the action space, and *r* is the immediate reward of the detection system. For evaluating the anomaly detection performance of an action (feature set and AI/ML algorithm), we consider common metrics [14] including precision (P_r), recall (R_e), F-score (F_s), accuracy (A_c), and false alarm rate (F_a). These metrics are calculated from the following observations: TP (True Positive) - number of attacks precisely detected; TN (True Negative)- number of normal patterns precisely classified; FP (False Positive) - number of normal patterns incorrectly classified; and FN (False Negative) - number of attacks unsuccessfully detected. The details of the MDP model are outlined hereafter.

1) **Space in state*:*** Formally, we can define the state space of the detection system as follows:

$$\mathcal{S} \triangleq \{(P_r, R_e, F_s, A_c, F_a)\}, \qquad (1)$$

where $P_r = \frac{TP}{TP+FP} \in [0,1]$, $R_e = \frac{TP}{TP+FN} \in [0,1]$, $F_s = \frac{2}{1/P_r+1/R_e} \in [0,1]$, $A_c = \frac{TP+TN}{TP+TN+FP+FN} \in [0,1]$ and $F_a = \frac{FP}{FP+TN} \in [0,1]$. Then, the state of the detection system is defined as a vector $s = (P_r, R_e, F_s, A_c, F_a) \in \mathcal{S}$.

Where

Then, the state of the detection system is defined as a vector $s = (P_r, R_e, F_s, A_c, F_a) \in S$.

2) Space in action*:*

$F = \{ f_1, f_2, ..., f_m\}$ defines a set of characteristic which composed for all suited and available features such as a feature set f_m consists of 4 features they are packet change ratio, average packets per flow, flow change ratio and average packet size per flow,). $L = \{l_1, l_2, ..., l_n\}$ represents a set of possible AI/ML algorithms that can be used for traffic flow classification, e.g., Support Vector Machine [12], Random Forest [14], and Self Organizing Map [15]. Then, a tuple, $< f_m, l_n >$, is referred as a combination of a feature set and an AI/ML algorithm. Applying a tuple $< f_m, l_n >$ to the environment, i.e., the detection system - see Fig. 2 (b), means an action is taken by the AOS component. Therefore, the action space is defined as:

$$A_s \{a : a = < f_m, l_n >; f_m \in F ; l_n \in L\}. \qquad (2)$$

3. **Sudden process for rewarding*:*** first examine the detection process of a tuple $< f_m, l_n >$ by five segments. Therefore, they defines the sudden process of reward for detecting the process after the AOS takes an action, state and function f

$r(s,a) = W_{Pr} P_r + W_{Re} R_e + W_{Fs} F_s + W_{Ac} A_c + W_{Fa} - F_a$, (3) where $W_{Pr}, W_{Re}, W_{Fs}, W_{Ac}$ and W_{Fa} are weight factors related to the corresponding evaluation criteria, and $W_{Pr} + W_{Re} + W_{Fs} + W_{Ac} + W_{Fa} = 1$.

Note that after performing an action, the AOS observes the feedback from the environment (detection system), i.e., the state vector and the reward value.

4.Optimization Formulation:

First define an optimization issues to occur the policy of optimal $\pi^*(s)$ which increases the immediate prize in each and every segment. Particularly in a state, it considered as vector such P_r, R_e, F_s, A_c and F_a, each policy accepts an action of optimal $< f_m, l_n >$ to increase the immediate reward for the process of detection which is shown in the equ. 3

$$\{r_i(s_i,\pi(s_i)) : r_i \in R; s_i \in S; \pi(s_i) \in A; 1 \le i \le mn\}, (4)\ \pi$$

where $r_i(s_i,\pi(s_i))$ is the immediate reward value associated with policy π at time step *i* in an episode.

Apply the Q-learning algorithm [13] to solve the optimization problem which is basically known as approach of reinforcement learning. AOS was able to proceed an optimal choosing with no need of knowledge about a characteristics set and linked with the ML or Al algorithm. It focus to identify the policy of optimal by $\pi^*: S \longrightarrow A$, that is a state characteristics or action set and ML or AL algorithm to increase the detection action for stealthy distributed denial of service attacks. The AOS constructs a Q table depend on a Q learning algorithm to protect the pair of state action combinations which is shown in the figure 2 (b). then it analyses the immediate price and the latest state and updates the q table by q function.[13]

Let us denote $\vartheta_\pi(s) : S \longrightarrow R$ as the expected return of a state *s* under a policy π generally, that is formed as follows:

$$\vartheta_\pi(s) = \mathbb{E}_\pi\left[\sum_{i=1}^{mn} \gamma r_i(s_i,a_i)|s_i = s\right] = \mathbb{E}_\pi[r_i(s_i,a_i) + \gamma\vartheta_\pi(s_{i+1})|s_i = s], \quad (5)$$

$$\vartheta_*(s) = \max\{E_\pi[r_i(s_i,a_i)|s_i = s]\}, \forall s \in S. \quad (6)$$

Hence, for all state-action (s,a) pairs, the optimal *Q*-functions are defined as:

$$Q_*(s,a) , r_i(s_i,a_i), \forall s \in S. \quad (7)$$

$$Q_{i+1}(s_i,a_i) = Q_i(s_i,a_i) + \alpha_i[r_i(s_i,a_i) - Q_i(s_i,a_i)], \quad (8)$$

4. Examined Q-MIND structure:

α_i is the learning rate [13]. The α_i value can be a The value α_i could be dynamic or stable at the learning operation. It also reduce the exploitation and exploration problems which has immediate effect on the convergence rate of algorithm learning and also used the epsilon greedy algorithm.[13]. According to the present state, the Q-learning method will take actions and the possibility of a decision was considered by the value α_i

$$s_i = (P_r, R_a, F_s, A_c, F_a),\ a_i = < f_m, l_n >$$

Algorithm 1

1. Inputs: F ;L; for a state-action pair (s,a) $\forall s \in S$, $a \in A$, then initialize a *Q*-table entry with $Q(s,a)$ value arbitrarily, α and , respectively.
2. Start
3. Repeat the loop for every episode.
4. Loop
5. Present state.
6. Implement action a_i according to the policy(0.
7. Consider the immediate reward r_i and new state s_{i+1}.
8. Update $Q(s_i,a_i)$ by using Equation 8.
9. Replace $s_i \leftarrow s_{i+1}$.
10. End loop
11. Outputs: $\pi^*(s) = \text{argmax}_a Q_*(s,a)$.

5. Structure of the examined model *Q*-MIND

This section deals with the model and process of the entire Q-MIND structure to detect and reduce the Distributed Denied of Service attacks.

5.1. Architecture of Q-MIND:

The structure of Q-MIND comprises the modules like a Knowledge Database for storing information about feasible features, an AI/ML-based anomaly detection system controlled by the *Q*-Learning agent, an Attack Mitigation Policy Creation module, a *Q*-Learning agent residing in the AOS component for optimizing the anomaly detection performance and a Data Pre- processing module which executed reduction rules into the information place to restrict stealthy distributed denial of service attack. The Q-MIND structure was shown in the figure 2.

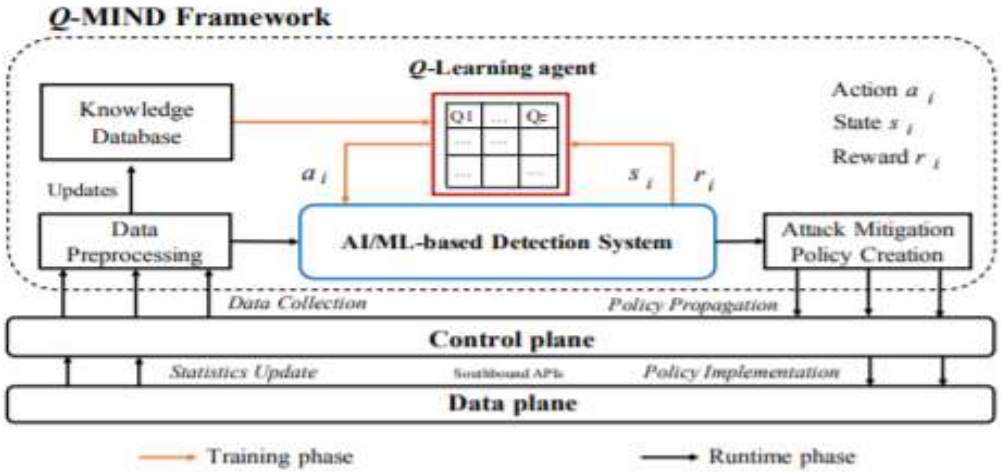

Fig. 2. Q-MIND structure for attacking stealthy distributed denial of service attacks in software denied network

B. Operation of Q-MIND:

The process of Q-MIND was described in the algorithm 2. At first, Q MIND runs the agent of Q learning to construct a Q-table and operates the action or characteristics combination set f_m and ML or AL algorithm was explained. The data which is labelled for the cross-authentication and training phase was obtained from experiments of stealthy distributed denial of service attacks in software denied network environment. It was generated from available information set publicly [11]. The agent of Q-learning organises cross authentication exams after finishing training of engine detection. Q-Mind enters into the runtime area to mitigate and detect stealthy distributed denial of service attacks by implementing the loop part of alg.2.

Algorithm 2 QMIND operation:

1. Construct a Q-table using algorithm 1
2. Consider the optimal process $< f_m, l_n >$ from Q-table
3. Execute a characteristic set f_m and ML or AL algorithm into the detection process
4. Loop
5. Gather all traffic statistics information from the data plane.
6. Split characteristics from statistical information
7. Feed characteristics to optimal ML or AL depend detection engine.
8. Get the result of detection for each source Internet Protocol address.
9. Develop attack reduction policies when a source of Internet Protocol address was an attacker.
10. Implement and propagate policies to the data plane.
11. End Loop.

6. Performance Evaluation

This section describes the execution of proof concept of QMIND structure and also an output of performance.

6.1. Investigating the scenario setup

This experiments was performed using Maxi network [16] to imitate a basic software denied network depend upon networking such as 9 hosts that is 6 malicious host and 3 benign and web server. A web connection and hosts are executed in the containers of Linux and link to an OF switch. Software denied network was managed by ONOS software denied network controller. [17]. Next consider 3 known ML or AL depend classifiers such as Random Forest (RF-supervised learning) [14], Self Organizing Maps (SOM-unsupervised learning) [15] and Support Vector Machine (SVM-supervised learning) [12]. A characteristic set was developed by the 10 correct features. They are fraction of TCP flows over total incoming flows, average packets per flow, packet change ratio, percentage of pair flows, entropy of incoming flows, growth of different ports, flow change ratio, average flow inter-arrival time and average packet size per flow. These characteristics are isolated for Internet Protocol an address source which was taken from traffic information adding 4500 ordinary samples and 4500 samples of attack. A traffic information set was considered from simulation of distributed denial of service in the software

denied network which was described above. It must be recorded that ML or AL algorithm needs at least two characteristics and the values of weight in eq.3

At first compare the optimized detection depend upon the Q learning with 3 other ML or Al depend classification and detection methods which apply various characteristics of techniquies like a Generic Algorithm with a SVM classifier (GASVM) [12], a Binary Bat Algorithm with a RF classifier (BBARF) [14] and a main element Analysis with a SVM classifier (PCASVM) [12]. So first examine the performance of stealthy Distributed denial of service attacks both at the run time and cross-authentication. The 2nd step was compare the performance of stealthy distributed denial of service attack of our threat mitigation and must delete all stemming flaws to restrict these threats sources for certain time like 40 seconds or so on.

6.2. Results and Analysis

6.2.1.*Convergence of the selection algorithm:*

First examines the stealthy Distributed Denial of Service attack at the time of cross-authentication and training level of QMIND. The detection performance oscillates at the time of 1st hundred replications of cross authentication and training phase due to the agent of Q-learning. It updates frequently Q-table in the starting stage of learning. This was shown in the figure 4(a). Hence it becomes constant and succeeds a value of 0.876 for the policy of optimal. Another scheme of detection performs well in the 1st replication and do not improve in the remaining time. Finally, it identifies the policy to yield the correct detection performance with the help of Q-Learning.

Attack detection performance applying the optimal policy (runtime phase): This step performs experiments in the maximum network emulation structure to investigate the stealthy Distributed Denial of Service attacks of Q-MIND.

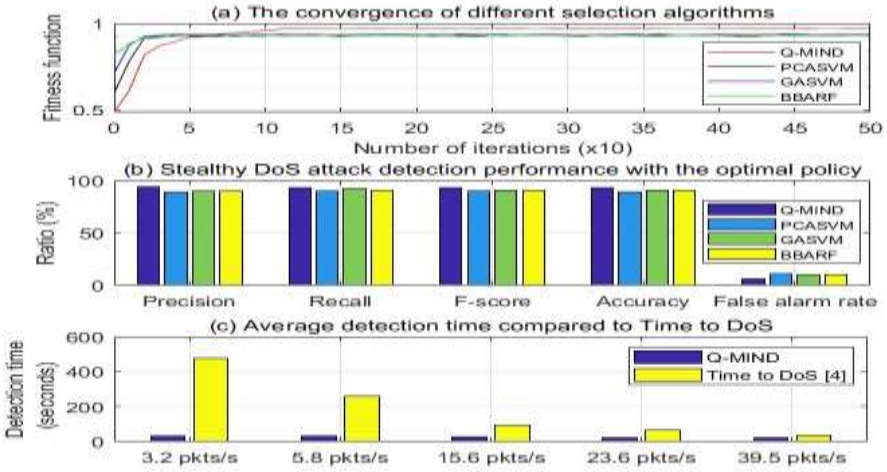

Fig. 3. Stealthy Distributed denial of service attack detection performance comparison

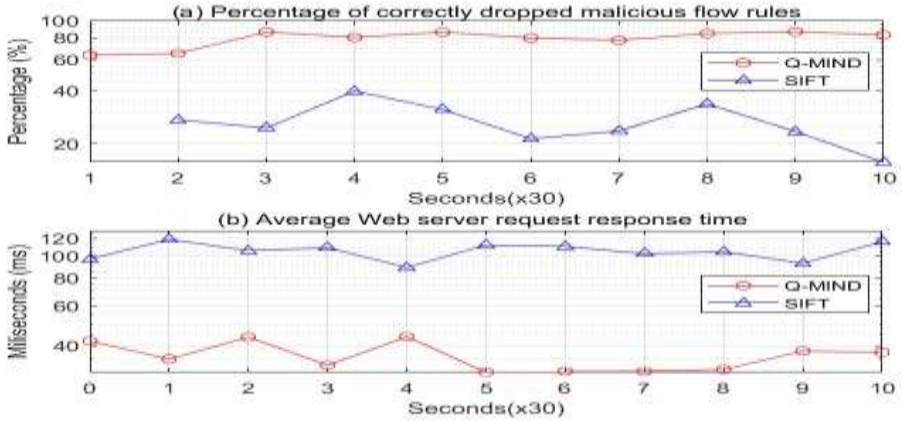

Fig. 4. Stealthy Distributed denial of service attack mitigation performance comparison

An output was shown in the figure 3(b). It could be analysed the policy using in the QMIND structure which performs the obtained results in the cross-authentication phases along with the policy of optimal. It is recorded that a stealthy Distributed Denial of Service attacks where 39.8 different packets were send to the Software

Denied Network switch.[4] It causes the switch to overflow after 40 seconds. So they report the average time of Q MIND structure during detection [3]. An output shows the Q MIND takes less time to detect the Distributed Denial of Service attacks with various rates and it avoids overflow issues in the switch completely.

1) Attack mitigation performance: To evaluate the attack reduction, first measure the malicious flow percentage in the switch and time response of the web connection if the network was under attack. QMIND achieves a perfect percentage of dropped threat flow because it executes policies and Internet protocol to detect the stem of the attacker. The Sum Index Flow Technology method drops attack flow only after the overflowed switched. This was shown in the figure 4

7. Conclusion

This paper proposes a novel machine learning structure named Q-MIND to defence against Distributed Denial of Service attacks in Software Denied Networks. So the researchers conducted a healthy analysis of the detecting process that incorporates a RLS to increases the performance of detection. An evaluation process results to demonstrate the Q-MIND by applying the policy of optimal from the agent of Q learning to achieve a high mitigation and detection in Distributed Denial of Service attack.

Reference

[1] Imran, M., Durad, M. H., Khan, F. A., &Derhab, A. Toward an optimal solution against denial of service attacks in software defined networks. Future Generation Computer Systems, 2019, 92, 444–453.

[2] Dharma, N. I. G., Muthohar, M. F., Prayuda, J. D. A., Priagung, K., & Choi, D. Time-based DDoS detection and mitigation for SDN controller. 2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS), 2015, 550–553.

[3] Lin, C.-H., Li, C.-Y., & Wang, K. Setting malicious flow entries against SDN operations: attacks and countermeasures.2018 IEEE Conference on Dependable and Secure Computing (DSC), 2018, 1–8.

[4] Tselios, C., Politis, I., &Kotsopoulos, S. Enhancing SDN security for IoT-related deployments through blockchain. 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), 2017, 303–308.

[5] Kuerban, M., Tian, Y., Yang, Q., Jia, Y., Huebert, B., &Poss, D. FlowSec: DOS attack mitigation strategy on SDN controller. 2016 IEEE International Conference on Networking, Architecture and Storage (NAS),2016, 1–2.

[6] Alshamrani, A., Chowdhary, A., Pisharody, S., Lu, D., & Huang, D. A defense system for defeating DDoS attacks in SDN based networks. Proceedings of the 15th ACM International Symposium on Mobility Management and Wireless Access, 2017, 83–92.

[7] Li, C., Wu, Y., Yuan, X., Sun, Z., Wang, W., Li, X., & Gong, L. Detection and defense of DDoS attack--based on deep learning in OpenFlow-based SDN. International Journal of Communication Systems, 2018,31(5), e3497.

[8] Li, W., Meng, W., & Kwok, L. F. A survey on OpenFlow-based Software Defined Networks: Security challenges and countermeasures. Journal of Network and Computer Applications, 2016,68, 126–139.

[9] Padmaja, S., &Vetriselvi, V. Mitigation of switch-Dos in software defined network. 2016 International Conference on Information Communication and Embedded Systems (ICICES), 2016, 1–5.

[10] Polat, H., &Polat, O. The effects of DoS attacks on ODL and POX SDN controllers. 2017 8th International Conference on Information Technology (ICIT), 2017, 554–558.

[11] Bholebawa, I. Z., &Dalal, U. D. Design and performance analysis of OpenFlow-enabled network topologies using Mininet. International Journal of Computer and Communication Engineering, 2016,5(6), 419.

[12] Ye, J., Cheng, X., Zhu, J., Feng, L., & Song, L. A DDoS attack detection method based on SVM in software defined network. Security and Communication Networks, 2018.

[13] Nanda, S., Zafari, F., DeCusatis, C., Wedaa, E., & Yang, B. Predicting network attack patterns in SDN using machine learning approach. 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), 2016, 167–172.

[14] Phan, T. V, Gias, T. M. R., Islam, S. T., Huong, T. T., Thanh, N. H., &Bauschert, T. Q-MIND: Defeating Stealthy DoS Attacks in SDN with a Machine-learning based Defense Framework. 2019 IEEE Global Communications Conference (GLOBECOM), 2019, 1–6.

[15] Li, J., Zhao, Z., Li, R., & Zhang, H. Ai-based two-stage intrusion detection for software defined iot networks. IEEE Internet of Things Journal, 2018, 6(2), 2093–2102.

[16] DeFreitas, H. S., Schwarz, M. F., AguiarBezerra, J., Ibarra, J. E., & others. Deploying SDN experiments in Latin America: the ONOS and SDN-IP application use case at AmLight, 2016.

[17] Naing, M. T., Khaing, T. T., & Maw, A. H. Evaluation of TCP and UDP Traffic over Software-Defined Networking. 2019 International Conference on Advanced Information Technologies (ICAIT), 2019, 7–12.

[18] Sakthivel, M. "An Analysis of Load Balancing Algorithm Using Software-Defined Network." Turkish Journal of Computer and Mathematics Education (TURCOMAT), 2019, 12.9 : 578-586.

[19] Sakthivel, M., J. Udaykumar, and V. Saravana Kumar. "Progressive AODV: A Routing Algorithm Intended for Mobile Ad-Hoc Networks." International Journal of Engineering and Advanced Technology (IJEAT) ISSN, 2019, 2249-8958.

[20] Kamalraj, R., and M. Sakthivel. "A hybrid model on child security and activities monitoring system using iot." 2018 International Conference on Inventive Research in Computing Applications (ICIRCA). IEEE, 2018.

[21] Sakthivel, M., T. Gnanaprakasam, and K. Siva Krishna Rao. "Reliable data delivery in MANETs using PGF and VH scheme." 2017 IEEE International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE). IEEE, 2017.

[22] Mohan, Ellappan, Arunachalam Rajesh, Gurram Sunitha, Reddy Madhavi Konduru, Janagaraj Avanija, and Loganathan Ganesh Babu. "A deep neural network learning-based speckle noise removal technique for enhancing the quality of synthetic caperture radar images." Concurrency and Computation: Practice and Experience 33, no. 13 (2021): e6239.