A novel algorithm for software defined networks model to enhance the quality of services and scalability in wireless network

Ahmad Sharadqeh
Department of Computer Engineering, Al-Balqa Applied University, As-Salt, Yordania

Article Info

Article history:

Received Nov 16, 2020
Revised Sep 24, 2021
Accepted Oct 14, 2021

Keywords:

Controller
Quality of service
Restricted Boltzmann machine
Software-defined network

ABSTRACT

Software defined networks (SDN) have replaced the traditional network architecture by separating the control from forwarding planes. SDN technology utilizes computer resources to provide worldwide effective service than the aggregation of single internet resources usage. Breakdown while resource allocation is a major concern in cloud computing due to the diverse and highly complex architecture of resources. These resources breakdowns cause delays in job completion and have a negative influence on attaining quality of service (QoS). In order to promote error-free task scheduling, this study represents a promising fault-tolerance scheduling technique. For optimum QoS, the suggested restricted Boltzmann machine (RBM) approach takes into account the most important characteristics like current consumption of the resources and rate of failure. The proposed approach's efficiency is verified using the MATLAB toolbox by employing widely used measures such as resource consumption, average processing time, throughput and rate of success.

Corresponding Author:

Ahmad Sharadqeh
Department of Computer Engineering, Al-Balqa Applied University
As-Salt, Yordania
Email: dr.ahmed.sharadqah@bau.edu.jo

1. INTRODUCTION

A distributed denial-of-service (DDoS) attack compromises set of hosts and utilizes internet protocol (IP) spoofing techniques to flood large volume of packets in order to make the target down. Flooding based DDoS attacks are highly vulnerable to the network infrastructure. The attack intensity is changing day by day and attackers deploy new techniques to increase the volume of attacks. It results in volumetric based DDoS attacks and bandwidth consumption attacks [1]–[3]. These kinds of attacks are also called resource consumption attacks in which an attacker tries to exhaust resources of the victim such as central processing unit (CPU), memory and network bandwidth [4]. In software-defined networking (SDN), DDoS attack occurs both in data and control plane layers [5]. The control plane layer is flooded with Packet-In request messages for every incoming unknown traffic flows. To handle the above said challenges, this chapter proposes a detection system for DDoS attacks in SDN. Hence, the concept of DDoS attack detection in all network scenarios becomes an active area of research. Most of the existing attack detection methods discussed in section 2 use supervised learning algorithm to detect attack traffic flows and they are unsuitable for stochastic network environment. This reason motivates to choose restricted Boltzmann machine (RBM) for attack detection. Additionally, in systems where access controls are no compulsory, the commitment 0€ guaranteeing data rests on the completion customer. This consistently obliges that customers appreciate the protection instruments offered by the frameworks and how to achieve the fancied security using these parts [6], [7].

SDN may be produced through operational data acquisition or the use of remote animal monitoring technology in livestock production systems [8]. The predictive analytical framework can be systematically applied to generate knowledge from these data to improve decision-making for livestock production, safety, and welfare. Defining a goal variable is analogous to creating the hypothesis in the experimental design of live animals and this variable affects the data management and form of models selected for study [9]. Data partitioning allows for the creation, refinement and evaluation of model output by a single dataset. The predictive analytical system ends with final model evaluation and provides an assessment of prediction accuracy including expected probability of finding events and non-events as described by the target variable [10], [11]. The predictive analytical approach offers a systematic technique for analyzing big data to improve the decision-making of livestock and encourage animal precision.

Determining the traffic density at the time of the network decreasing speed helps to avoid network congestion and can keep the network open longer. The output result shows that our solution is superior in all respects compared with other clustering approaches [12], [13]. Weighted inertia-based dynamic virtual bat algorithm (WIDVBA) is presented to minimize the latency incurred in the dissemination of data among the network 's SDN nodes by trying to integrate the strengths of the traditional bat algorithm with swarm intelligence and inertial weight based on particle swarm optimization with simulated annealing (PSO-SA) [14]. WIDBVA is found to be particularly powerful as a needless search space exploration and exploitation is significantly reduced in line with the literature's existing work [15]. A trust model, namely F-TRUST, is proposed by performing fuzzy logic in a QoS to determine the accuracy and credibility of both event messages and senders of the event messages. Plausibility, experience, and type of vehicle are used as the key criteria for calculating the confidence score to this end [16].

SDN node density corresponding to the number of network sharing the route and the channel for wireless transmission of safety messages. The suggestion was made using a discrete simulator of physical layer-based events in parallel with the models [17]. We obtained two equations by simulating and observing the correlation among the amount of nodes present in the field and the amount of nodes transmitted simultaneously [18]. Update voiceprint to enable detection on SCH. This enhancement greatly reduces the computation time and decreases the false positives. In addition, the voiceprint is extended to find sudden modifications to the received signal strength indicator (RSSI) time series using multiple change-point detection method. Voiceprint therefore has the ability to recognize certain illegitimate nodes that exercise power control during Sybil attacks [19]. The power-controlled Sybil attack, when we implement RSSI-based detection scheme it is still a complicated problem [20]. This paper is intended to track network nodes and a network simulator. However, in practice control of network nodes that are linked to the network simulator is also needed. In monitoring large-scale emulated wireless network such as SDN, the design of a lightweight on-memory logging system is implemented using a wireless network tap tool and daemon wtap80211 [21].

Cloud platforms can be divided into three categories: network as a service (PaaS), technology as a service (IaaS) and software as a service (SaaS) to clients through the internet [22]. On a remote basis, SaaS delivers software solutions to clients as online services. It provides cloud-based application that is costly accessible and hosted online by an certain institution [23], [24]. Because it does not need to be installed on specific devices, this sort of service is simple to use and administer. If the apps are not built using standard protocols, SaaS might well have difficulties with privacy, compatibility, and connectivity [25]. PaaS allows clients to create and deploy customized apps in a hosted environment through the website. It enables programmers to create customized apps. Data stored in a vendor-controlled cloud server raises security issues and problems under the PaaS paradigm. IaaS enables clients to remotely access the equipment on a "pay-as-you-go" basis. These technologies can outperform most local alternatives [26], [27]. It allows people or companies to develop and maintain their data as they expand, buy the resources that they require to develop hardware or technology with no need to manage and maintain on-site servers. The business or corporation has no authority on cloud security in a laaS environment and is only liable for updated software and repair [12], [28], [29],

The rest of the paper will be organized as follows. The second section examines the important works in SDN resource allocation. Section 3 discusses the proposed adaptive SDN resource allocation strategy. Section 4 describes the experiments and the findings that were obtained. Section 5 summarizes the suggested technique and suggests additional research.

2. RESEARCH METHOD

Figure 1 depicts the proposed cloud platform. The user's tasks will be collected by the cloud environment broker via the portal. The tasks provided by users are then saved in the request pool, which is a backlog of work. The centralized resource element gathers resource information from the virtual server, which contains information on the virtual machines (VMs) maintained and hosted by the cloud framework.

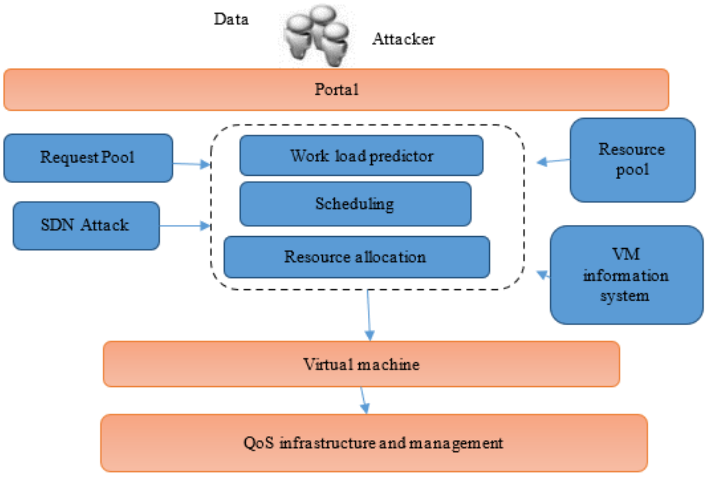

Figure 1. SDN attack level system model

A VM is a simulated version of a computer system where users' tasks are carried out. The failure detector element detects errors in cloud resources and tracks the fault rate of the resources that may have been utilized for assigning resources to work. The workload prediction element recognizes and manages the task of each VM to minimize resource overuse and inactivity. Finally, the scheduling and resource allocation elements assign more appropriate resources to the tasks given by users.

Algorithm 1 depicts the pseudocode for the proposed adaptive fault tolerant resource allocation (AFTRA) system. The breakdown of resources is prevalent in the cloud environment owing to unavoidable hardware or software faults. The error rate of cloud resources must be considered while establishing an effective resource distribution policy. The failure has a direct impact on the cloud system's dependability. Tasks assigned to these resources will be postponed or denied as a result of resource failure. In most cases, errors in the cloud environment occur randomly, which may be expressed by a Poisson probability distribution. The suggested technique consists of two stages, namely the data collection process and the attack detection process, which are seen in Figure 2.

Algorithm 1: restricted Boltzmann machine (RBM)

```
Begin
While there are tasks in Task Queue
Select a task i from the Queue for all Virtual Machines
Calculate Expected Failure Rate (EFR) and probability of the attack of the VM
Calculate the workload of the VM and denote it as WL
Classify the virtual machine such as overloaded, under loaded and normally loaded as per the
Pseudocode given in Figure 1.
End for
Assign the selected task to the VM which has lowest failure probability in the under
loaded list End while
Remove the task i from the queue End
```

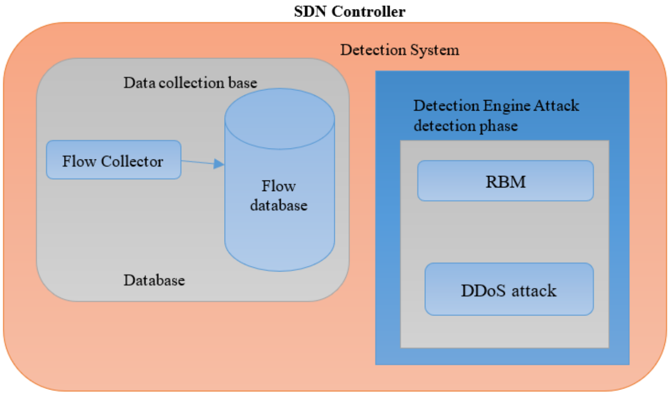

Figure 2. RBM based detection system

Data collection phase can be bonded with flow collector and flow database collector. In this phase, will provide the data to the detection engine system.

$$E(V,H) = \sum_{i=1}^{m} \quad \sum_{j=1}^{n} \quad W_{m,n} h_n v_m - \sum_{i=1}^{m} \quad a_m v_m - \sum_{j-1}^{n} \quad b_n h_n \tag{1}$$

Where, $E(V,H)$: Energy configuration of the RBM deployed network,
$W_{m,n}$: Weight system matrix,
$a_m v_m$: Bias unit,
v_m : Visible layer,
h_n : Hidden layer.

The JPD can be given as

$$P(V,H) = \frac{1}{Z} e^{-(V,H)} \tag{2}$$

where, $Z =$ system partition of the network,

$$P\left(\frac{h}{v}\right) = \prod_{j=1}^{n} \quad sigmoid(b_n + v^T W_{:n}) \tag{3}$$

where,
$W_{:n}$: network feature vector matrix
b_n : bias unit of hidden layer.

Experiments to track both legal and malicious traffic flows shall be performed for an hour. The calculated values for the network traffic flow to the SDN controller are detected every 10 minutes and the values found are on average about 90 to 100 as lower threshold points and on average about 100 to 120 pps as higher threshold points. The process of DDoS attack detection is iterated for 20 times and the values are analyzed. Based on the ranges of lesser and higher threshold points, the average threshold value is fixed as 100 req/sec from a single MAC address for generating Packet-In request messages. Controller hit count can be given as in (4).

$$H_c = \frac{No.of.packets\ with\ same\ IP}{Avg\ threshold\ value} \tag{4}$$

Algorithm 2: RBM system learning procedure

```
Input:  Initially assign the Feature vector V={v1,……,vm}
Output: Machine learning result
Step 1: Assign the hidden and visible layers vm and hn
Step 2: Calculate the JPD function for the initial parameters and finally determine the energy
        value E(V,H).
Step 3: To calculate the PDF by using Gibb's sampling algorithm for hidden network.
```

In the last step, the remaining 41 data points are used to perform the RBM. From 85 training data points, only 50% of the data are utilized to obtain the accurate by RBM technique. Here, the data points can be reduced so it requires only less number of mathematical formulation step to calculate data points. In addition to that the computation time is reduced and obtain higher accuracy.

The CH with RBM reduces the overall processing time which is lower than original RBM. The accuracy of the classifier also higher than other methods. The main procedure of the CH technique is to perform the cluster on the training data points and also convex hull are structured for each cluster.

Passive monitoring is used to monitor the congestion of the SDN/OpenFlow network when the SDN shows the congestion level of the OpenFlow switches, such as when the utilization of the channel is higher than the 90 percent of the threshold and when the signal is transmitted to the SDN controller. The controller is tested after getting the signals. After this, the SDN evaluates the congested switch flow meter counter to check for filled streams of link use or video streams.

Executions of symmetric key can be especially successful in order to ensure that consumers do not encounter any significant time delay due to unscrambling. Likewise, symmetric-key offers a degree of confirmation as data mixed with one symmetric key cannot be decoded with some other symmetric key. Therefore, as long as can be used by the two gatherings to scramble correspondences and keep the symmetric key private, each gathering will ensure that it communicates to the other as long as the decoded messages become consistent and pleasant.

Designing procedure for RBM:

```
Choose the cluster value K.
Perform the k-means clustering techniques.
For k varies up to K (k≤ $K$) for each cluster do
   Based on the cluster k, checked the data points class label.
      If the cluster data points are the single class.
        Allocate the cluster label as 'Singular'.
      Else, allocate the cluster label as 'Nonsingular'.
      End
End
 For 'Singular' cluster do
      Perform Quickhull techniques.
      Estimate the convex hull ($V_1$), which denotes the class-1 label vertices points.
      Estimate the convex hull ($V_2$), which denotes the class-2 label vertices points.
      Set of vertices points are formed.
      Eliminate each clusters sample not related to the group.
  End
 For 'Nonsingular' cluster do
    Choose each cluster data points and forms into a single set.
 End
 Remaining samples are structured as 'Rem' dataset.
 perform RBM to the 'Rem' values
```

A collection of descriptive attributes attack will define private keys in our construction a party that wants to scramble a message will suggest a method that private keys must follow in order to unscramble through an entrance tree structure. Every inside hub of the tree is an edge door and the leaves are related with qualities. We utilize a similar documentation as to depict the entrance trees, despite the fact that for our situation the credit s is utilized to distinguish the keys (as restricted to the data). Specified in the private key, while the text of the figure is marked with a lot of clarity.

3. RESULTS AND DISCUSSION

Results for different SDN tools are seen in this section, such as controller bandwidth, switch consumption rate of energy, and flow table attacks throughout DDoS attacks. The flow of standard network traffic is detected where the arriving flow demands are higher than the average value of the threshold and where abrupt anomalies are evaluated. In Figure 3. the arriving rate of Packet-In request signals to the SDN controller is greater than 100, the graph indicates a dramatic change in the bandwidth usage rate. In comparison, the proposed detection system is contrasted with the current flood guard process in Figure 4.

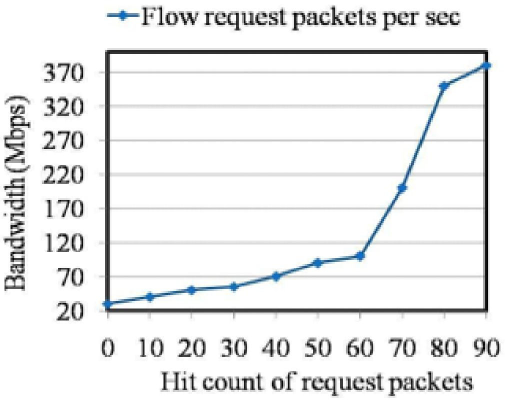

Figure 3. Controller analysis graph

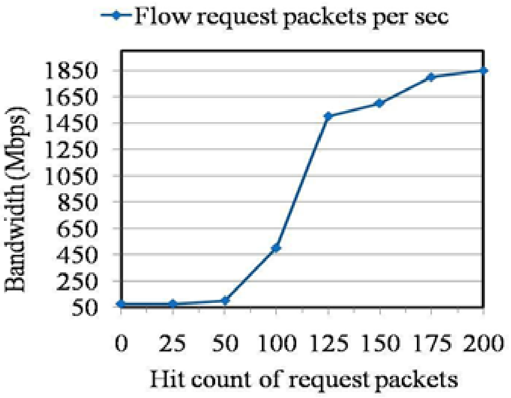

Figure 4. Bandwidth utilization of controller during DDoS attack

Figure 5 shows the performance of the proposed RBM based DDoS attack detection based on the bandwidth use of the SDN controller. The findings reveal that the RBM-based detection system outperforms the flood guard in terms of bandwidth usage and the traces reveal that the bandwidth is more robust even though the packet rate is 130 per second. There is a greater bandwidth consumption in the case of the flood guard system. In this study, Figure 6 and Figure 7 are compared to achieve a contrast among the SDN controller reaction without and with threshold respectively. In Figure 6, the response time of the controller and the number of switches is set to be μ=10,000 and 10 switches respectively. Their flow rate to the controller is 100, 200 and 300. Figure 6 shows that the reaction time of the controller increases when there is no threshold. In Figure 7, the threshold value is set at 150. With the support of Figure 6 and Figure 7, it can be shown that the reaction time of the controller is influenced by the transfer flow requests from the SDN controller. Table 1 shows a comparison of performance metrics between flood guard and the proposed RBM based detection method in that the reaction time of the controller is higher.

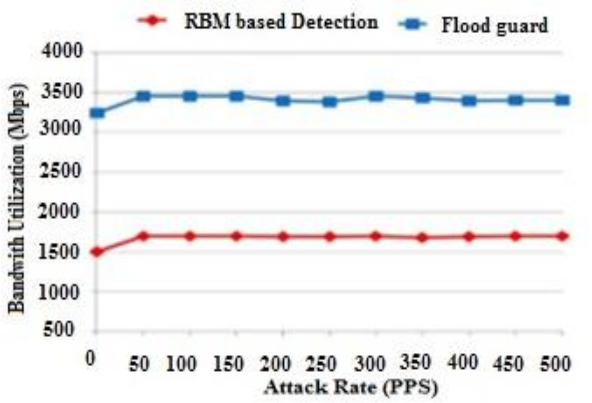

Figure 5. Bandwidth utilization

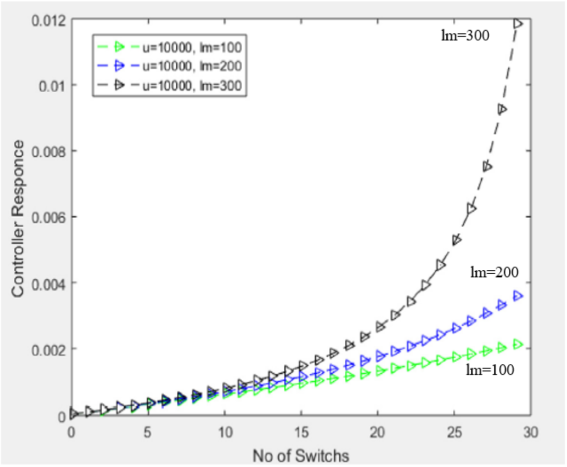

Figure. 6. μ and the number of switches without threshold

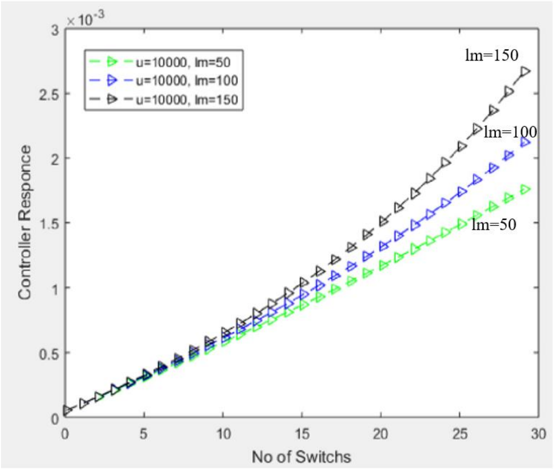

Figure 7. μ and the number of switches with threshold

Table 1. Comparison between flood guard and proposed RBM method

Sl. No	Performance Metrics	Flood Guard	Proposed RBM based Detection Method
1	Attack Detection Rate (R_d)	90%	92%
2	False Positive Rate (R_{FP})	10%	8%

4. CONCLUSION

In this paper, a detection of faults in SDN can be implemented using the RBM. The network attract is performed on the data points and then Controller algorithm is used to find the vertices of the data points belongs to each nodes. The performance of the RBM is analyzed. It effectively detects DDoS attacks with traffic flow variations. Performance research reveals that their proposed RBM-based monitoring method detects DDoS flood guard. For future work, the same dataset can be applied on the different new attack appear in SDN. In addition, identification of IoTs fault accurately in network layer and the sensor nodes.

REFERENCES

[1] H. Polat, O. Polat, and A. Cetin, "Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models," *Sustainability*, vol. 12, no. 3, Feb. 2020, Art. no. 1035, doi: 10.3390/su12031035.

[2] S. Wang *et al.*, "SECOD: SDN sEcure control and data plane algorithm for detecting and defending against DoS attacks," in *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, 2018, pp. 1–5, doi: 10.1109/NOMS.2018.8406196.

[3] Z. Shu, J. Wan, D. Li, J. Lin, A. V. Vasilakos, and M. Imran, "Security in Software-Defined Networking: Threats and Countermeasures," *Mobile Networks and Applications*, vol. 21, no. 5, pp. 764–776, Oct. 2016, doi: 10.1007/s11036-016-0676-x.

[4] J. Cui, M. Wang, Y. Luo, and H. Zhong, "DDoS detection and defense mechanism based on cognitive-inspired computing in SDN," *Future Generation Computer Systems*, vol. 97, pp. 275–283, Aug. 2019, doi: 10.1016/j.future.2019.02.037.

[5] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A Survey on Software-Defined Networking," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 27–51, 2015, doi: 10.1109/COMST.2014.2330903.

[6] N. Z. Bawany, J. A. Shamsi, and K. Salah, "DDoS Attack Detection and Mitigation Using SDN: Methods, Practices, and Solutions," *Arabian Journal for Science and Engineering*, vol. 42, no. 2, pp. 425–441, Feb. 2017, doi: 10.1007/s13369-017-2414-5.

[7] M. Kuerban, Y. Tian, Q. Yang, Y. Jia, B. Huebert, and D. Poss, "FlowSec: DOS Attack Mitigation Strategy on SDN Controller," in *2016 IEEE International Conference on Networking, Architecture and Storage (NAS)*, 2016, pp. 1–2, doi: 10.1109/NAS.2016.7549402.

[8] S. Behal, K. Kumar, and M. Sachdeva, "D-FACE: An anomaly based distributed approach for early detection of DDoS attacks and flash events," *Journal of Network and Computer Applications*, vol. 111, pp. 49–63, Jun. 2018, doi: 10.1016/j.jnca.2018.03.024.

[9] L. Barki, A. Shidling, N. Meti, D. G. Narayan, and M. M. Mulla, "Detection of distributed denial of service attacks in software defined networks," in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2016, pp. 2576–2581, doi: 10.1109/ICACCI.2016.7732445.

[10] Y. Chen, J. Pei, and D. Li, "DETPro: A High-Efficiency and Low-Latency System Against DDoS Attacks in SDN Based on Decision Tree," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, vol. 2019-May, pp. 1–6, doi: 10.1109/ICC.2019.8761580.

[11] D. Jankowski and M. Amanowicz, "On Efficiency of Selected Machine Learning Algorithms for Intrusion Detection in Software Defined Networks," *International Journal of Electronics and Telecommunications*, vol. 62, no. 3, pp. 247–252, Sep. 2016, doi: 10.1515/eletel-2016-0033.

[12] A. Alshamrani, A. Chowdhary, S. Pisharody, D. Lu, and D. Huang, "A Defense System for Defeating DDoS Attacks in SDN based Networks," in *Proceedings of the 15th ACM International Symposium on Mobility Management and Wireless Access*, 2017, pp. 83–92, doi: 10.1145/3132062.3132074.

[13] A. Saied, R. E. Overill, and T. Radzik, "Detection of known and unknown DDoS attacks using Artificial Neural Networks," *Neurocomputing*, vol. 172, pp. 385–393, Jan. 2016, doi: 10.1016/j.neucom.2015.04.101.

[14] L. Wei and C. Fung, "FlowRanger: A request prioritizing algorithm for controller DoS attacks in Software Defined Networks," in *2015 IEEE International Conference on Communications (ICC)*, 2015, vol. 2015-Septe, pp. 5254–5259, doi: 10.1109/ICC.2015.7249158.

[15] Kwangtae Jeong, Jinwook Kim, and Young-Tak Kim, "QoS-aware Network Operating System for software defined networking with Generalized OpenFlows," in *2012 IEEE Network Operations and Management Symposium*, 2012, pp. 1167–1174, doi: 10.1109/NOMS.2012.6212044.

[16] M. Zhang, J. Bi, J. Bai, and G. Li, "FloodShield: Securing the SDN Infrastructure Against Denial-of-Service Attacks," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 2018, pp. 687–698, doi: 10.1109/TrustCom/BigDataSE.2018.00101.

[17] V. K. Yadav, M. C. Trivedi, and B. M. Mehtre, "DDA: An Approach to Handle DDoS (Ping Flood) Attack," in *Advances in Intelligent Systems and Computing*, vol. 408, Springer Singapore, 2016, pp. 11–23.

[18] S. Behal and K. Kumar, "Characterization and comparison of DDoS attack tools and traffic generators - a review," *International Journal of Network Security*, vol. 19, no. 3, pp. 383–393, 2017, doi: 10.6633/IJNS.201703.19(3).07.

[19] O. Joldzic, Z. Djuric, and P. Vuletic, "A transparent and scalable anomaly-based DoS detection method," *Computer Networks*, vol. 104, pp. 27–42, Jul. 2016, doi: 10.1016/j.comnet.2016.05.004.

[20] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDOS) flooding attacks," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 4, pp. 2046–2069, 2013, doi: 10.1109/SURV.2013.031413.00127.

[21] S. Wang, K. G. Chavez, and S. Kandeepan, "SECO: SDN sEcure COntroller algorithm for detecting and defending denial of service attacks," in *2017 5th International Conference on Information and Communication Technology (ICoIC7)*, 2017, pp. 1–6, doi: 10.1109/ICoICT.2017.8074692.

[22] R. Durner, C. Lorenz, M. Wiedemann, and W. Kellerer, "Detecting and mitigating denial of service attacks against the data plane in software defined networks," in *2017 IEEE Conference on Network Softwarization (NetSoft)*, 2017, pp. 1–6, doi: 10.1109/NETSOFT.2017.8004229.

[23] H. Wang *et al.*, "DDoS Attack in Software Defined Networks: A Survey," *Neural Regeneration Research*, vol. 7, no. 14, pp. 1095–1100, 2017, doi: 10.3969/j.issn.1673-5188.2017.03.003.

[24] B. Wang, Y. Zheng, W. Lou, and Y. T. Hou, "DDoS Attack Protection in the Era of Cloud Computing and Software-Defined Networking," in *2014 IEEE 22nd International Conference on Network Protocols*, 2014, pp. 624–629, doi: 10.1109/ICNP.2014.99.

[25] D. Hyun, J. Kim, D. Hong, and J. P. Jeong, "SDN-based network security functions for effective DDoS attack mitigation," in *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, 2017, vol. 2017-Decem, pp. 834–839, doi: 10.1109/ICTC.2017.8190794.

[26] Z. Liu, Y. He, W. Wang, and B. Zhang, "DDoS attack detection scheme based on entropy and PSO-BP neural network in SDN," *China Communications*, vol. 16, no. 7, pp. 144–155, Jul. 2019, doi: 10.23919/JCC.2019.07.012.

[27] S. Nanda, F. Zafari, C. DeCusatis, E. Wedaa, and B. Yang, "Predicting network attack patterns in SDN using machine learning approach," in *2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 2016, pp. 167–172, doi: 10.1109/NFV-SDN.2016.7919493.

[28] N. Meti, D. G. Narayan, and V. P. Baligar, "Detection of distributed denial of service attacks using machine learning algorithms in software defined networks," in *2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017*, 2017, vol. 2017-Janua, pp. 1366–1371, doi: 10.1109/ICACCI.2017.8126031.

[29] C. Li *et al.*, "Detection and defense of DDoS attack-based on deep learning in OpenFlow-based SDN," *International Journal of Communication Systems*, vol. 31, no. 5, Mar. 2018, Art. no. e3497, doi: 10.1002/dac.3497.

BIOGRAPHY OF AUTHOR

Ahmad Sharadqeh received his PhD Degree in Computer, computing system and networks from National Technical of Ukraine "Kyiv Polytechnic Institute Ukraine in 2007. Since 2009, Ahmed Sharadqeh has been an Associate professor in the Computer Engineering Department, Faculty of Engineering Technology, at Al-Balqa Applied University. His research interests include Performance of network, Quality services, security network, image processing, digital systems design, operating system, and Microprocessors. He can be contacted at email: dr.ahmed.sharadqah@bau.edu.jo.