Real-time DDoS Attack Defense System in SDN Using LSSOM

Shijin Liu*, Hiroaki Fukuda†, Paul Leger‡
*Department of Computer Science and Engineering, Shibaura Institute of Technology, Beijing, China
†Department of Computer Science and Engineering, Shibaura Institute of Technology, Tokyo, Japan
‡Escuela de Ingeniería, Universidad Católica del Norte, Coquimbo, Chile
ma22160@shibaura-it.ac.jp, hiroaki@shibaura-it.ac.jp, pleger@ucn.cl

***Abstract*—Software-Defined Networking (SDN) is a new paradigm in network architecture that improves scalability, flexibility, control, and network management by separating the control plane from the data plane. SDN controllers have a global view of the entire network and provide the ability to dynamically change traffic forwarding rules. However, Introducing SDN brings some new DDoS attack vulnerabilities, such as limited flow table capacity and single point failure of a controller.**

This paper proposes an approach that combines linear discriminant analysis (LDA) and a supervised self-organizing map (SOM) called LSSOM that enables to detecting suspicious packets to defend against DDoS attacks in real-time. Our experimental results show that using LSSOM achieves 98.2% accuracy and reduces the classification time by 73.5% compared to using supervised SOM only.

***Index Terms*—Software-Defined Networking, Distributed Denial of Service, Machine Learning, Linear Discriminant Analysis, Real-time System**

I. Introduction

Software-defined Networking (SDN) originated from Stanford University's Clean State project in 2006. It is an innovative architecture to solve the problem that the traditional network cannot meet the increasing requirements by separating the control plane and data plane to define and control the network with the characteristics of openness, flexibility, and programmability [1]. In SDN architecture, the controller located on the control plane allocates and schedules network resources depending on the global topology information of the network, and the data plane is only responsible for data forwarding and state collection. With the above approach, SDN realized flexible control of network traffic, effectively simplifying network management and also offering users better network programmability.

The architecture of SDN is shown in Figure 1. The control plane mainly consists of controllers, which are equivalent to the brain of the entire network. The data plane consists of a large number of SDN switches such as Open vSwitch [2] that comply with SDN standards and other network devices. The SDN switch is mainly responsible for data forwarding, and each switch maintains flow tables that record information about traffic forwarding. Each flow table entry includes several information such as a source address, a destination address and an action (*e.g.,* forward and drop). The controller realizes real-time packet forwarding control by issuing flow table entries to the switches via the southbound interface. However, controller-centric SDN architectures are more vulnerable to the risk of a single point of failure caused by traffic overload than traditional networks [3].

Distributed Denial of Service (DDoS) attacks are considered to be one of the most destructive attacks, and DDoS attacks are capable of causing significant damage to Information and Communication Technology(ICT) infrastructures. DDoS attacks aim to consume the network bandwidth by making a large number of illegal packets. When a large number of packets come into a switch that does not have flow entries for the packets, it will also send a large number of requests to its controller to obtain certain flow entries for deciding actions. As a consequence, the network bandwidth and flow table capacity will be illegally consumed.

This paper proposes a system that uses a machine learning approach named LSSOM to mitigate DDoS attacks on SDN networks in real-time. In the subsequent section, we call this LSSOM-based real-time defense system as real-time system. LSSOM is a combination of linear discriminant analysis (LDA) and supervised Self-organizing map(SSOM). SSOM is selected because it provides high accuracy while keeping the features of visualization and interpretability [4]. At the same time, LDA is selected to reduce the data dimensionality in order to solve the deficiency of SSOM which runs slower on high-dimensional data. We assume that LSSOM is used to classify normal traffic and DDoS attack traffic after detecting DDoS attack. Therefore we do not mention how to detect DDoS attack is happening, which is out of the scope of this paper. Our real-time system consists of three parts. The first part is the feature extraction model, which collects network traffic information in real-time and selects appropriate features for classifying DDoS or normal traffic. The second part is a packet detection model based on LSSOM, which classifies the

979-8-3503-9804-5/23/$31.00 ©2023 IEEE

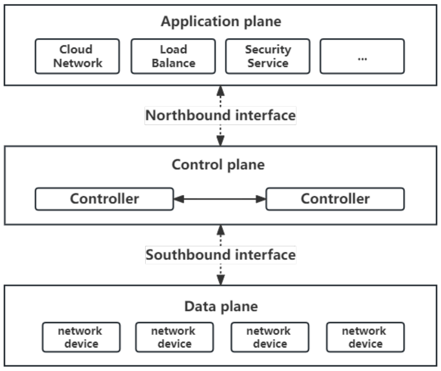

Fig. 1. SDN architectural view.

network traffic into a normal trafic group and DDoS trafic group based on extracted features in the first step. Finally, the mitigation module issues flow rules to drop packets based on the classification results for attack traffic.

This paper is organized as follows. Section II describes the design of real-time system and implementation is described in Section III. Section IV shows the experiments, including performance metrics, results, and discussion. In Section V, we compare related works and explain our advantages. Finally, we conclude this paper with future issues in Section VI

II. Toward real-time defense against DDoS

This section mainly introduces the real-time system against DDoS attacks. In Section II-B, we describe how the flow table capacity affects the detection period in DDoS attacks. In Section II-C, we briefly explain our real-time system. Finally, we introduce the dataset and features used in our experiments in Section II-D.

A. SDN Packet Handling

An SDN controller communicates with SDN switches using the southbound interface, which is a common protocol such as OpenFlow [5] as shown in Figure 1. OpenFlow defines some messages to maintain flow entries in a flow table. An SDN switch deals with incoming packets based on the matched flow entries. When an incoming packet does not match any flow entries in the SDN switch, the SDN switch sends OpenFlow's *packet_in* message to the corresponding SDN controller. The SDN controller decides how to handle the incoming packet and send flow entries to the SDN switch using OpenFlow's *flow_mod* message as shown in Figure 2.

On the other hand, we might need to control the network behavior (*i.e.,* forwarding/dropping packets) based on applications we develop such as load balancers and security services. These applications are basically located on the application plane, which is a different place (host) from the control plane, then control the network behavior using SDN controllers. Therefore these applications usually use the northbound interface which is commonly provided as Web interfaces such as RESTful APIs by SDN controllers.

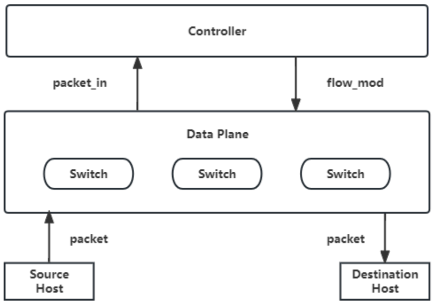

Fig. 2. Packet forwarding process

B. Time urgency for real-time defense

SDN switches usually use Ternary Content Addressable Memory (TCAM) to store their flow table entries given by an SDN controller [6]. It is also reported that a commodity SDN switch can store about only 1500 entries because of the high cost and energy consumption of TCAM [7].

Typically, a DDoS attack aims to consume network resources by generating a large number of illegal packets in a short period of time. Different from traditional networks, flow entries maintained by SDN switches are also target resources to be consumed. Thereby if the number of flow entries in the flow table in SDN switches exceed the capacity limit (*i.e.,* 1500 entries), an SDN controller needs to delete and/or update flow entries in SDN switches, consuming network bandwidth and resources of the SDN controllers.

The time to send a *flow_mod* message is about 0.5 milliseconds [8]. This means that in the worst situation, DDoS attackers can overflow the flow table capacity in 0.75s. Therefore, the period of detecting and analyzing network traffic should be limited to 0.75s in real-time defense, otherwise, it will be exposed to the risk of flow table overload.

C. An architecture of the real-time system

The defense against DDoS attacks can be considered a binary classification problem. After the attack starts, there are large numbers of normal packets and attack packets flooding the network. Identifying and classifying packets more quickly and efficiently means the ability to recover from the attack earlier. To this end, as we have mentioned in Section I, the proposed real-time system is based on LSSOM, which consists of LDA and (S)SOM. This section briefly introduces SOM and LDA first, then show the entire architecture of the real-time system.

1) LDA: Linear discriminant analysis: LDA is a popular dimensionality reduction approach for pre-processing steps in data mining and machine learning applications. The main aim of LDA is to project a dataset with a high number of features onto a less-dimensional space with good class separability. This will reduce computational costs. [9] LDA is a supervised data dimensionality reduction method and can also be used in classification cases. Compared with unsupervised principal component analysis, apart from maximizing the variance of

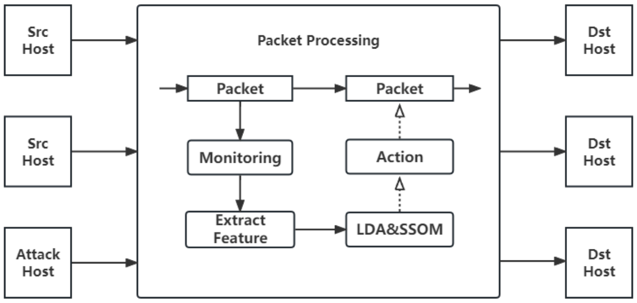

Fig. 3. Architecture of real-time defense system

data, LDA also maximizes the separation of multiple classes. The goal of LDA is to project a dimension space onto a lesser subspace without disturbing the class information.

In the issue of DDoS attack defense, packets are categorized as normal and attack, which is a dichotomous problem. In this case, LDA reduces the input data to 1 dimension for SOM, which will be explained in the next, to classify incoming packets into normal and attack quickly.

2) SOM: Self-organization map: SOM is an unsupervised clustering algorithm, of which the principle is to map sample data into an $n \times n$ two-dimensional grid.

The authors in [4] propose a supervised SOM (SSOM) and perform regression and classification experiments. The result shows that supervised SOM has great potential in classification problems. However, using a single SSOM will cause a long classification time problem, thereby we apply LDA and reduce dimensions before applying SSOM.

The real-time system is designed to provide a capability that can monitor network traffic and mitigate DDoS attacks in real-time. Figure 3 illustrates the entire architecture of real-time system. The solid arrows indicate the packet information flow and the dotted arrows indicate the flow rules affect flow. When a packet passes through the switch, The monitoring module sniffs, and stores the passing packet information. Then, features are extracted from the Formatting stored data and input to the linear discriminant analysis(LDA) model to perform the data dimensionality reduction process. The processed data will be classified as normal or attack packets by SSOM. Our subsequent experiments show that the data after LDA dimensionality reduction can be identified and classified more rapidly. Once the classification result is ”attack”, a ”drop” rule will be generated and issued to the switch to cut the connection of the attack host by using the northbound interface provided by the SDN controller. During the whole process, if no attack traffic packets are found, the flow rules will not be generated so the normal packet forwarding will not be influenced.

D. KDD99 and ignored parameters

KDD99 is a dataset about network intrusion detection. Although KDD99 is not collected from SDN traffic, it is the most extensively used dataset in the network intrusion detection field [10]. The KDD99 has the data on the different type of attacks that includes DOS, probing, R2L, and U2R, in which DOS is used in our study. Features contained in KDD99 is shown in Table I. Although we use the KDD99 as the dataset for training and testing, we do not use all features contained in KDD99 because some of them are useless for the real-time system.

In the description of KDD99 features, from F.NO 10 to 22 in Table I are marked as content features. The content features are mainly applied to mark specific services and are difficult to collect from the data plane. [11] shows that content features are not suitable as a basis for detecting DDoS attacks. Therefore these content features are ignored in our study to reduce the calculation cost. In addition, F.NO 1 to 4 in Table I are highly correlated with the protocol and should be ignored as redundant features. As a consequence, 24 features are retained in this paper.

III. Implementation of the real-time system

We use Mininet [12] as a simulation environment. In Mininet, we also use Ryu [13] as an SDN controller and sFlow technology [14]. Mininet also contains built-in OVS for the OpenFlow protocol. In addition, OVS also supports sFlow monitoring technology, which consists of two components: sFlow agent and sFlow collector. The sFlow agent is deployed in OVS for listening to the ports and sending passing packet information to the sFlow collector. sFlowtool [14], as one of the sFlow collectors, can convert the collected packet information into PCAP format [15]. PCAP is a common format used for network traffic data storage and has good compatibility and open source feature. Using PCAP facilitates the decoupling of various processes. The feature extraction is developed based on the *kdd99_feature_extractor*.

TABLE I
Features of KDD99

F.NO	F.NAME	F.NO	F.NAME
1	duration	22	is_guest_login
2	protocol_type	23	count
3	service	24	srv_count
4	flag	25	serror_rate
5	src_bytes	26	srv_serror_rate
5	dst_bytes	27	rerror_rate
7	land	28	srv_rerror_rate
8	wrong_fragment	29	same_srv_rate
9	urgent	30	diff_srv_rate
10	hot	31	srv_diff_host_rate
11	num_failed_logins	32	dst_host_count
12	logged_in	33	dst_host_srv_count
13	num_compromised	34	dst_host_same_srv_rate
14	root_shell	35	dst_host_diff_srv_rate
15	su_attempted	36	dst_host_same_src_port_rate
16	num_root	37	dst_host_srv_diff_host_rate
17	num_file_creations	38	dst_host_serror_rate
18	num_shells	39	dst_host_srv_serror_rate
19	num_access_files	40	dst_host_rerror_rate
20	num_outbound_cmds	41	dst_host_srv_rerror_rate
21	is_host_login		

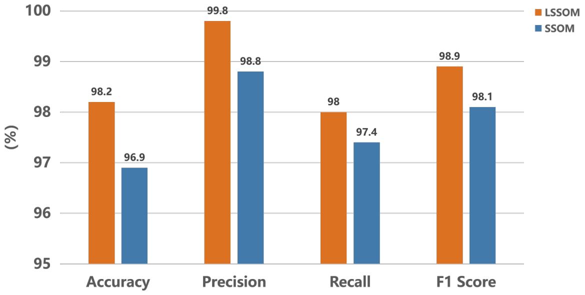

Fig. 4. Performance metrics when grid size is 40

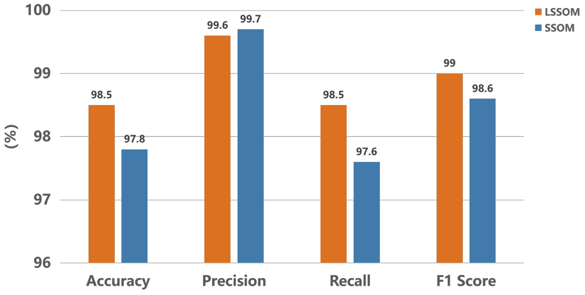

Fig. 5. Performance metrics when grid size is 50

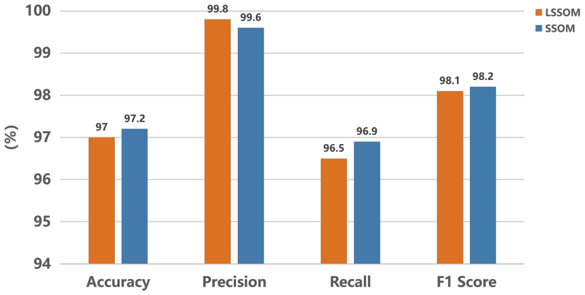

Fig. 6. Performance metrics when grid size is 60

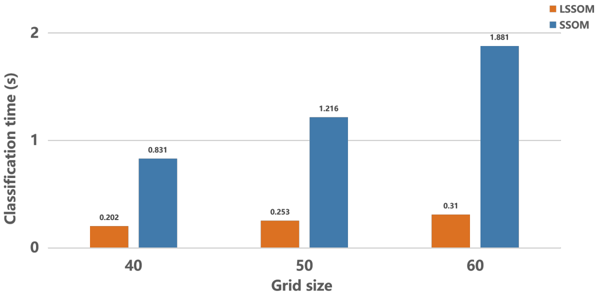

Fig. 7. Classification time in different grid size

IV. Experiment and Discussion

A. Experiment setup

The experimental testbed is a host running Windows 10, CPU is Core i7-12700H 2.70 GHz with 16GB RAM. We compared the supervised SOM and LSSOM with different grid sizes. The recommended grid size is 50 to 80 [4]. Considering the possibility of overtraining for larger grid sizes, We successively set the grid size to 40, 50, and 60. The number of iterations for all models is set to 1,000.

On the other hand, Accuracy, Precision, Recall, and F1 Score are common metrics to measure the performance of the machine learning algorithms [16]. Accuracy can be used to determine the degree of machine learning classification algorithm. Metrics are defined as the following equations (from equation 1 to 4). Among these metrics, TN stands for True Negative, which refers to instances where the model correctly predicted the negative class. True Positive(TP) refer to instances where the model correctly predicted the positive class. False Negative(FN) refer to instances where the model predicted the negative class, but the true class was positive. False Positive(FP) refer to instances where the model predicted the positive class, but the true class was negative.

In addition to these metrics, the classification time for the test data will also be considered. The classification time is defined as the time spent to detect 10,000 packets. For LSSOM, the classification time contains the time consumed by performing the dimensionality reduction process.

$$\text{Accuracy} \;=\; \frac{TP+TN}{TP+FP+FN+TN} \qquad (1)$$

$$\text{Precision} \;=\; \frac{TP}{TP+FP} \qquad (2)$$

$$\text{Recall} \;=\; \frac{TP}{TP+FN} \qquad (3)$$

$$\text{F1 score} = 2 \cdot \frac{\text{precision} \;\cdot\; \text{recall}}{\text{precision} \;+\; \text{recall}} \qquad (4)$$

B. Result and Discussion

The results of these experiments are shown in Figure 4 to 6. Through these results, LSSOM can perform almost the same results in all metrics as SSOM, which is considered useful in machine learning classification [4]. In [4], the accuracy of SSOM in the classification task was 81.6%, which is lower than our results even though the classification tasks are different. Based on these observations, we think LSSOM can work effectively to classify incoming packets into normal traffic and attack traffic.

Apart from the performances in these metrics, Figure 7 shows the classification time for different grid size in which we measured the total time to detect 10,000 packets. Compared to SSOM, LSSOM shows significantly better performance in terms of classification time. Especially, in the case of 40 grid sizes, LSSOM performs better scores than SSOM in all metrics in addition to classification time. Based on these results, theoretically, LSSOM can detect over 35000 packets in less than 0.75s in the case of 40 grid size (*i.e.,* 0.202s for 10,000 packets), which can avoid flow table overload in SDN switches.

V. Related Work

Ibrahim *et al*. [17] use the unsupervised artificial neural networks to construct an intrusion traffic detection mechanism. The proposed system uses a self-organizing map (SOM) artificial neural network for detection and classifies network traffic into the attack and normal. The detection rate can reach 92.37% on the KDD 99 dataset and 75.49% on the NSL-KDD dataset. It has been experimentally proven that SOM is more powerful than static networks because dynamic networks have memory, they can be trained to learn sequential or time-varying patterns. On the basis of this research, LSSOM improves the accuracy and retains the high interpretability of SOM. Simplifying the feature set makes LSSOM more suitable to be applied in real-time systems.

In [18], the authors present a DDoS attack defense system called FL-GUARD. The system provides the ability to detect and mitigate DDoS attacks at the application layer, based on the network traffic monitoring tool sFlow-RT. The SVM algorithm is used in attack detection to classify the traffic, and the total system can detect DDoS attacks with high accuracy. However, the sFlow-RT-based traffic monitoring method does not facilitate getting all packet information and converting to various formats in order to take into account the subsequent in-depth analysis..

Deepa *et al*. [19] design and apply a DDoS detection mechanism based on hybrid machine learning techniques. This hybrid machine learning algorithm combines supervised SVM and unsupervised SOM. as a result, it is shown that the hybrid algorithm achieves better accuracy, detection rate, and low false alarm rate compared to using a single algorithm. However, the packet collection process is not performed in real-time.

SL model proposed in [20] collects the number of Packet_In requests per time slot through the SDN controller and analyzes the flow fluctuations to detect DDoS attacks against the SDN controller. In their tests, the SL model significantly reduces the training time and can predict under several milliseconds with a real-time detection accuracy of over 90%. However, the Packet_In-based detection method has hysteresis and is exposed to the risk of flow table overload when under attack. LSSOM does not rely on the controller's response to the attack traffic. Because the packet detection module is separate from the SDN control plane and the traffic information collection is performed in the data plane. We can detect packets at the same time as the attack happens, therefore there is no hysteresis.

VI. Conclusion

This paper proposes an architecture of the real-time defense system against DDoS attacks in SDN. This system is high decoupling between every module and can continuously detect and mitigate attack packets. In addition, we propose an approach called LSSOM and a suitable feature set that applies to the real-time system. Compared with using single supervised SOM, LSSOM improves accuracy, precision, recall, and F1 score when the grid size is 40, and the classification time is reduced to satisfy the requirements of real-time systems. It means that the real-time system using LSSOM can defend against DDoS attacks efficiently.

Our future work will focus on the perception of whether the DDoS attack has occurred and more flexible mitigation strategies, aiming to further reduce the real-time cost.

References

[1] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2014.

[2] B. Pfaff, J. Pettit, T. Koponen, E. Jackson, A. Zhou, J. Rajahalme, J. Gross, A. Wang, J. Stringer, P. Shelar *et al.*, "The design and implementation of open {vSwitch}," in *12th USENIX symposium on networked systems design and implementation (NSDI 15)*, 2015, pp. 117–130.

[3] 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, 2017.

[4] F. M. Riese, S. Keller, and S. Hinz, "Supervised and semi-supervised self-organizing maps for regression and classification focusing on hyperspectral data," *Remote Sensing*, vol. 12, no. 1, p. 7, 2019.

[5] S. Ali, M. K. Alvi, S. Faizullah, M. A. Khan, A. Alshanqiti, and I. Khan, "Detecting ddos attack on sdn due to vulnerabilities in openflow," in *2019 International Conference on Advances in the Emerging Computing Technologies (AECT)*, 2020, pp. 1–6.

[6] F. Hu, Q. Hao, and K. Bao, "A survey on software-defined network and openflow: From concept to implementation," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2181–2206, 2014.

[7] T. Li, H. Zhou, H. Luo, I. You, and Q. Xu, "Sat-flow: Multi-strategy flow table management for software defined satellite networks," *IEEE Access*, vol. 5, pp. 14 952–14 965, 2017.

[8] C. Metter, S. Gebert, S. Lange, T. Zinner, P. Tran-Gia, and M. Jarschel, "Investigating the impact of network topology on the processing times of sdn controllers," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 2015, pp. 1214–1219.

[9] G. T. Reddy, M. P. K. Reddy, K. Lakshmanna, R. Kaluri, D. S. Rajput, G. Srivastava, and T. Baker, "Analysis of dimensionality reduction techniques on big data," *IEEE Access*, vol. 8, pp. 54 776–54 788, 2020.

[10] S. Bajpai and K. Sharma, "A framework for intrusion detection models for iot networks using deep learning," 2022.

[11] R. C. Staudemeyer and C. W. Omlin, "Extracting salient features for network intrusion detection using machine learning methods," *South African computer journal*, vol. 52, no. 1, pp. 82–96, 2014.

[12] L. Bob. Mininet. [Online]. Available: http://mininet.org/

[13] F. Tomonori, "Introduction to ryu sdn framework," *Open Networking Summit*, pp. 1–14, 2013.

[14] InMon. (2015) sflowtool. [Online]. Available: https://github.com/sflow/sflowtool

[15] TcpdumpGroup. tcpdump. [Online]. Available: https://www.tcpdump.org/

[16] K. M. Sudar, M. Beulah, P. Deepalakshmi, P. Nagaraj, and P. Chinnasamy, "Detection of distributed denial of service attacks in sdn using machine learning techniques," in *2021 International Conference on Computer Communication and Informatics (ICCCI)*, 2021, pp. 1–5.

[17] L. M. Ibrahim, D. T. Basheer, and M. S. Mahmod, "A comparison study for intrusion database (kdd99, nsl-kdd) based on self organization map (som) artificial neural network," *Journal of Engineering Science and Technology*, vol. 8, no. 1, pp. 107–119, 2013.

[18] J. Liu, Y. Lai, and S. Zhang, "Fl-guard: A detection and defense system for ddos attack in sdn," in *Proceedings of the 2017 international conference on cryptography, security and privacy*, 2017, pp. 107–111.

[19] V. Deepa, K. M. Sudar, and P. Deepalakshmi, "Detection of ddos attack on sdn control plane using hybrid machine learning techniques," in *2018 International Conference on Smart Systems and Inventive Technology*, 2018, pp. 299–303.

[20] S. Wang, J. F. Balarezo, K. G. Chavez, A. Al-Hourani, S. Kandeepan, M. R. Asghar, and G. Russello, "Detecting flooding ddos attacks in software defined networks using supervised learning techniques," *Engineering Science and Technology, an International Journal*, p. 101176, 2022.