International Journal of Applied Science and Engineering

A novel sFlow based DDoS detection model in software defined networking

Neelam Gupta [1], Sarvesh Tanwar [1*], Sumit Badotra [2]

[1] *Amity Institute of Information Technology, Amity University, Noida, 201301, India*
[2] *School of Computer Science Engineering and Technology, Bennett University, Greater Noida, 201310, India*

ABSTRACT

Software-defined networking (SDN) is a networking model that makes networks programmable, convenient, and agile. Its centralized control plane is a key component of DDoS, which causes system resources and prevents services from responding to legitimate requests. The SDN controller's centralized structure makes it extremely susceptible to DDoS attacks. DDoS attacks are quickly identified in SDN controllers, which is essential for preventing them. There are several suggested techniques for finding DDoS attacks, but not much research has been done. The first step in preventing DDoS attacks is to identify them. In this paper, sFlow is used to build an early DDoS detection tool with SDN controller integration for widely used SDN controllers (OpenDaylight and Ryu). Several network scenarios are taken into consideration for the experimental configuration, with Mininet and penetration tools used to create hosts and switches. Each situation involves a different quantity of hosts, switches, and packet forwarding. The number of hosts and switches used in each scenario varies, and the created packets of data range from 1,00,000 to 5,00,000 per second. The controllers are inundated with data traffic, and Wireshark is used to analyse the data traffic, and our DDoS detection system is evaluated based on a variety of criteria, including how long it takes to detect a DDoS assault, the round-trip time (RTT), the percentage of packet loss, and the type of DDoS attack. It has been discovered that ODL takes longer than Ryu to shut down after detecting a successful DDoS attack. Our technology makes sure quick DDoS attacks are promptly detected, improving the SDN controller's performance without compromising the network's overall operation.

Keywords: SDN, DDoS attacks, sFlow, Mininet, Wireshark, SDN controllers, Opendaylight and Ryu.

OPEN ACCESS

Received: December 19, 2023
Revised: January 4, 2024
Accepted: January 12, 2024

Corresponding Author:
Sarvesh Tanwar
stanwar@amity.edu

Publisher:
Chaoyang University of Technology
ISSN: 1727-2394 (Print)
ISSN: 1727-7841 (Online)

1. INTRODUCTION

For the past few decades, the traditional network framework has not altered considerably and has become unmanageable. Separating the control plane from the data plane in a network architecture known as SDN makes the system customizable and simpler to administer, as shown in Fig. 1. It is ideal for apps with a distributed nature and a lot of data flow (Pattanaik et al., 2019). The separation of the network's control plane from the underlying switches, modems, and routers reduces the vertical combination and reorganizes the network configuration and policy definition processes. The controller is used to programme the precise path that each packet is forwarded, and the network layer devices are transformed into basic flow tables (Haider et al., 2020), reorganizing the network configuration and policy definition processes.

The SDN controller is the heart and brain of the system, and if someone gains access to it, they can take control of the entire SDN. DDoS assaults (Singh and Behal, 2020) can potentially compromise the controller, as they occur when a specific target, the

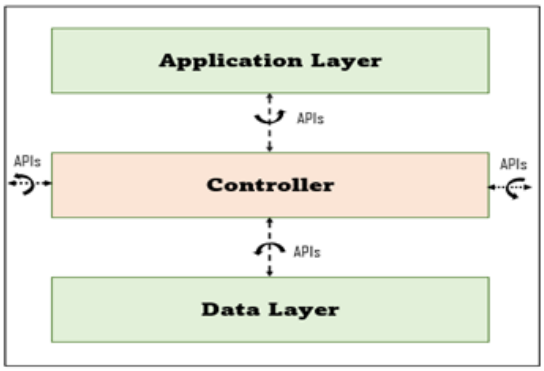

Fig. 1. SDN Architecture

system ceases to provide services to actual users or clients because of a portal, administrative servers, connection supplier, or other components of the system. Three essential elements make up a typical DDoS attack: a master, a victim, and numerous bots. The intermediate layer, known as the SDN controller, uses data from infrastructure layer devices and communicates with SDN (Santos et al., 2020) applications using an abstract network view. Data forwarding and processing are handled by the infrastructure layer, which receives instructions from the controller that are path-based (Shalini et al., 2021). The control plane creates paths for the data plane, which is additionally located on the same device, to employ in transporting packets between the source and the destination. The control plane is used to build network paths and provide instructions to the data plane. The control plane and data plane are embedded into conventional network devices, providing advantages such as lower-cost network infrastructure (Meti et al., 2017), quicker network adoption and troubleshooting, more network visibility, and increased programmability and customization in networking. The systems of conventional network devices like switches and routers in the network industry, centralization has many advantages, especially in the area of data centres where several workstations are interconnected to switch devices.

While SDN is centrally managed and manages the internet architecture by giving guidelines for the data layer that relate to paths, the SDN controller (Joëlle and Park, 2018) is the primary attackers' intended victim. Attackers attempt to access the controller or spoof it in some other way. A hacker can take control of the network if the controller becomes compromised due to the decoupling of the control and data planes, which can lead to a security flaw (Jumani and Laghari, 2021; Yin et al., 2021). SDN has several advantages but also faces difficulties, such as security. Network assaults on SDN (Kumar et al., 2018) can take many forms, such as IP spoofing, control plane attacks, man-in-the-middle attacks, and data plane security, but DDoS attacks are the most prevalent and disruptive. The controller may be the target of DDoS attacks that disrupt its network functions. Due to the fact that ODL and Ryu are two widely used SDN controllers used by large enterprises (Gupta et al., 2019), we chose these two widely used SDN controllers for our experiments in this paper. DDoS attacks against the SDN network have been found by a number of researchers, and a program has been developed to detect them quickly and accurately (Carvalho et al., 2021). We have taken into account many network conditions and factors to analyse its performance. These attacks take advantage of the SDN infrastructure's bandwidth and scale restrictions and finding DDoS assaults is the first and most important step in stopping them.

This paper discusses the major contributions of penetration tools like Hping3 and sFlow to analyse ODL and Ryu's susceptibility to DDoS attacks (HTTP, UDP, and TCP) and create a detection tool. Three separate virtual machines are constructed and connected to one another over a virtual switch using the mininet emulator tool, which is integrated with the sFlow tool to receive alerts and create a log file. We have discovered via experimentation that our detection tool accurately and promptly identifies DDoS assaults in light of their characteristics and various network configurations. We have taken into account a variety of network traffic scenarios, and evaluation and comparison are done based on factors such as the number of data packets flooded, the round-trip time (RTT), the type of DDoS attack, the time it took to detect the attack, the number of hosts, and the percentage of packet loss.

1.2 Authors' contributions

- In this paper, author assess the controllers' performance. To the finest of our knowledge, there aren't many studies in the literature that concentrate on actively measuring the performance of SDN controllers ODL and Ryu.
- The experiment design deviates from the methods used in this publication, and the research only takes a few topologies into account. The ODL and Ryu SDN controllers work in different DDoS attack situations (tree and linear topology in various attacks using sFlow tools).
- The result of packet loss, round trip time (RTT), time to identify a DDoS attack throughput, jitter and latency is depicted inside comparative results. It is crucial to assess new releases of these controllers to better comprehend the performance enhancements. We are expecting that this study will shed lighter on how these controller’s work.

1.3 Motivation

The paper compares two SDN controllers using sFlow tools and three attacks (ICMP, TCP, UDP) using parameters like packet loss, RTT, time to identify DDoS attack, latency, jitter, and throughput. It analyses results to determine the best SDN controller and choose between Ryu and ODL controllers to minimise network complexity, costs, and maintenance in large organisations. Methods/statistical analysis: SDN's advantages in segregating the control plane from the data plane and centralising control from an SDN controller make it an explosive subject. This study examines the following topics: packet loss, RTT, time to identify a

DDoS attack, latency, jitter, ping delays, throughput, and the current network implementation needs of large businesses using traditional networks.

Real-time DDoS attack detection utilising the sFlow tool in conjunction with SDN controllers; after analysing the data, one may conclude that the ODL controller is the best SDN option, which will lower a network's complexity, maintenance costs, and requirements in any large organisation. While many academics have tried to identify DoS/DDoS attacks, relatively few have focused on the detection and mitigation of DDoS attacks. While many researchers employ sFlow to investigate SDN security, only a small number of these studies have additionally taken ODL into account. The purpose of this work is to suggest a method for identifying DDoS-based attacks against the SDN controller. In this research, we use techniques, namely ODL and Ryu, to detect DDoS attacks on SDN controllers. The foundation of these techniques is network traffic analysis. The results demonstrate that our approach can detect DDoS attacks with low error rates in an effective and efficient manner, providing a possible means of enhancing the security of SDN networks.

The paper's remaining sections could be categorised into many groups. Part 2 explains the relevant work. The experimental methodology is explained in part 3, and the experiment's findings are presented together with a discussion in part 4. Part 5 provides our effort's conclusion and the project's future scope.

2. RELATED WORKS

SDN is a viable solution to DDoS attacks, which are becoming more common, which demonstrates that existing defense techniques are only partially successful (Varghese and Muniyal, 2021). It has come to light as a viable solution to the growing issue of DDoS attacks. This section aims to clarify the material from the literature review

In 2017, the author used machine learning techniques to categorize connections into valid and illegitimate ones, using the support vector machine (Meti et al., 2017) classifier and the neural network (Dehkordi et al., 2021) classifier. A 2018 paper reviews prior research on DDoS assaults detection and mitigation based SDN environment approaches. In a different work, Kumar et al. (2018) introduces SAFEETY, a cutting-edge method for preventing and early detection of TCP SYN flooding. To evaluate the unpredictability of flow data, it combines the coding and broad accessibility of SDN methodologies with the entropy approach. The destination IP address (Makuvaza et al., 2021) and a few TCP flag attributes are included in the entropy data (Yin et al., 2023). Implement safety as an extension module in the Floodlight Controller and test it out in various possibilities with constraints. Other factors such as CPU usage and attack detection period are also looked at, and improvements are seen in a number of instances.

The author's goal in 2019 is to show how a Switches that have been compromised can begin a DDoS attack on an SDN controller by changing idle and hard timeout parameters (Patidar and Singh, 2021). Instead of choosing a threshold based on the number of flow entry requests, a mechanism is provided to detect such an attack and counter it. In 2020, it is advised to use a deep convolutional neural network aggregation methodology for SDNs' DDoS attack (Valdovinos et al., 2021) detection. A flow-based dataset is utilised to assess the suggested system against predetermined benchmarks. Increased accuracy is shown in comparison to current related detection methods. Some authors' works use the SDN's centralized management and programming capabilities to achieve the network flow data's randomization. This statistical method makes use of the source IP in the network and different TCP flag attributes to compute entropy from them. The suggested method can identify DDoS assaults such as TCP SYN flood, Ping flood, and slow HTTP attacks (Batool et al., 2022) that are volume-based and application-based. Mininet is used to simulate the approach, and the POX controller is used to execute detection and mitigation measures. The experimental findings demonstrate that the solution has improved performance assessment metrics such as attack detection time, delay to fulfil a valid request while an attacker is present, and CPU utilization.

In 2021, DDoS attack detection will be done by watching TCP handshake packets. To determine the variation in the number of interconnections that are only partially visible, the cumulative sum (CUSUM) algorithm is utilised. To identify DDoS attacks, Meti et al. (2017) suggests data plane-ML, a machine learning (ML) system that operates on the data layer. This approach produces better results than other DDoS solutions using CUSUM. Data plane-ML uses white box switches and P4 components to monitor controlled (Jiang et al., 2022) delivery and ML packages when running ML models at the data plane, allowing for more sophisticated solutions that mimic input flow. The suggested data plane-ability MLs were tested on real network traces for DDoS attack detection using KNN, SVM, and RF algorithms. The experiment results revealed that the proposed MLs used 23% less CPU and were 23% faster than statistically based techniques. The majority of organizations are the targets of these attacks, even the most prestigious financial organizations and legislative agencies. Investigating novel paradigms that can effectively counter DDoS attacks is essential.

In 2022–2023, author used datasets from CTU-13 and ISOT to assess the efficacy of a proposed algorithm for intrusion detection. The algorithm employs a three-phased detection scheme: weighted moving averages, standard deviations, and entropy. Several research' findings reveal that the suggested approach is more reliable, portable, and has a lower detection rate. The efficacy of the proposed strategy (Mishra and Gupta, 2022) in comparison to other related methods is demonstrated by the detection's precision in this work. To reduce the false-positive rate, Jiang et al. (2022) proposed algorithm employs a three-phased

detection scheme. BSD-Guard is an SDN-targeted DDoS defense system built on blockchain technology that was proposed by one author. With a secure intermediary layer built on blockchain that calculates the suspect rate of new flows and sends suspect lists to blockchain, it provides SDN controllers (Valizadeh and Taghinezhad-Niar, 2022) with a cooperative detection and mitigation mechanism. A smart contract created on blockchain consists of joint defensive methods based on suspicion lists reported from various SDN domains (Anyanwu et al., 2022). The secure middle plane installs the appropriate actions into the appropriate switches after converting the received defense strategies into specific flow table actions. The summary of literature review (Sheibani et al., 2022) is shown in Table 1. The experimental findings show that BSD-Guard can effectively identify the attack vector (Swami et al., 2022) in a scenario with several controllers, detect DoS and DDoS attacks, and provide accurate defensive tactics close to the attack source. The grid search cross-validation (GSCV) performed best when the radial basis function kernel of the support vector machine (RBF-SVM) kernel parameters "C" and "gamma" were set to their ideal values of 100 and 0.1, respectively. The recommended method outperformed existing benchmarks with an overall accuracy of 99.33%, a detection rate of 99.22%, and an average squared error of 0.07%. The GSCV exhaustive parameter search technique and the RBF-SVM algorithm are used in several authors' proposed solutions. Key performance metrics were used to compare the effectiveness of different machine learning algorithms, and experimental simulations revealed that the suggested strategy has a mean absolute error of 0.06 and an overall error of 99.4 percent (Sai et al., 2022).

ODL and Ryu are the two most popular open-source SDN controllers in terms of performance and acceptance. Due to the significance of the ODL and Ryu controllers in SDN, the performance of each controller is assessed in this paper in terms of detection and mitigation of DDoS attacks, RTT, and packets loss. Given that a controller needs some time to run tests for different traffic types and packet sizes, we employed IP traffic with ICMP, TCP, and UDP messages of various sizes to measure performance. sFlow is used to measure both controllers' while taking into account a tree topology and a linear topology for the network. The results of the trial showed that ODL outperformed Ryu based on specific characteristics. This research can assist many academics and businesspeople in deciding which of the two controllers to use in various application settings. The related background of achieving this goal has been explored in software-defined networking, SDN protocol (Open Flow), Ryu, ODL, DDoS attacks, and controller comparison. SDN is a popular topic due to its advantages in centralising control from an SDN controller and isolating the control plane and data plane. This research focuses on detection and mitigation of DDoS attacks, RTT, and packets loss, and current network implementation requirements (Laghari et al., 2023) in large organisations using traditional networks.

Although these controllers are the best, they have some drawbacks, including the fact that the ODL SDN controller, which was developed by Cisco, implies an uncertain connection between the two, which is an enormous setback. Since it appears, they are obligated to a certain provider, most clients are looking for SDN alternatives to their present setups. Open source has the benefit of being free from ownership, but it also has limitations. According to research, Ryu controller performance remains constant as network hosts and switches increase, and the best controller utilisation depends on application needs

3. PROBLEM STATEMENT

This research compares the performance of two popular open-source SDN controllers, Ryu and ODL, while accounting for two distinct network topologies. Every controller exhibits a unique personality. Many academics are now assessing and comparing these controls. Based on certain attributes, ODL performed better than Ryu, according to the trial's findings. This study can help various academics and industrialists choose between the two controllers in a variety of application contexts, such as servers and the Web of things.

3.1 Methodology

Using a network emulator tool called mininet emulation tool, multiple DDoS attacks on the central controller are being identified. This tool proves to be really beneficial for creating a virtual network. The host and switch counts can be made possible with the use of this technology. In order to provide very flexible, customized routing in SDN (Sritharan et al., 2022), it constructs OpenFlow switches for a number of versions, including 1.0, 1.2, and 1.3, among others. It provides a secure communication channel for the hosts, switches, and controllers of a virtual network. In our testing, we make use of the OpenFlow protocol version 1.3. VM-1, VM-2, and VM-3 are three distinct virtual machines that have been created. Fig. 2 shows that VM-1 is made up of mininet.

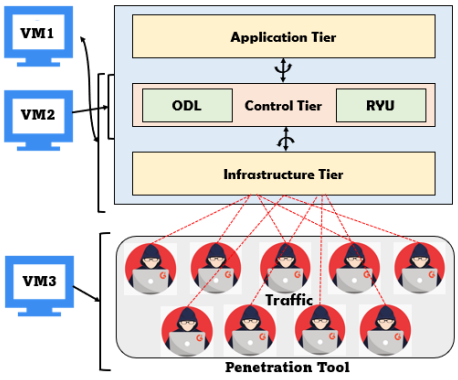

Fig. 2. The experimental setup

Table 1. Summary of literature review

Authors	Challenges	Proposed solution
Pattanaik et al., 2019	Switches that have been hacked and had their idle and hard timeout settings altered continuously ask the controller for entries into flow tables	To rapidly recognise and neutralise such an attack on the second iteration of the request
Haider et al., 2020	An effective method for detecting large-scale, complex DDoS attacks at an early stage	A cutting-edge flow-based dataset and accepted benchmarks are used to analyse the best architecture, a deep convolutional neural network (CNN) ensemble framework
Singh et al., 2020	SDN itself faces significant implementation difficulties and is prone to various forms of network intrusions.	Four types of mechanisms make up four kinds of mechanisms: Information theory-based, machine learning-based, AI-based, ANN-based, and ad hoc approaches
Santos et al., 2020	Attacks on the controller, the flow-table, and the bandwidth	The decision tree algorithm and the random forest method have the fastest processing times.
Shalini et al., 2021	Analysing TCP handshake packets on a regular basis	The quantity of semi connections using the cumulative sum (CUSUM) technique to identify change points
Meti et al., 2017	DDoS attacks are likely to target the controller, draining resources and rendering services unavailable	Two machine learning methods are the support vector machine (SVM) classifier and the neural network
Joëlle et al., 2018	The attackers find the control plane to be an appealing target for security attacks	An analysis of previous studies on DDoS attack detection and mitigation methods utilised in the SDN environment
Kumar et al., 2018	TCP SYN packets are sent in a flood to the control plane using switches in the data plane	The early identification and control of TCP SYN flooding are made possible by a special strategy known as SAFETY
Gupta et al., 2019	DNS Amplification	Using a middle portion solution and a bloom filter as a defence mechanism, an attack is detected and neutralised
Carvalho et al., 2021	On actual network traces, find DDoS attacks	A machine learning (ML) solution called data plane-ML was evaluated using KNN, SVM, and RF algorithms
Varghese et al., 2021	Advantages and disadvantages of each SDN architectural style for detecting DdoS	The evolution in the SDN architecture solution for DDoS assaults by analysing several approaches
Dehkordi et al., 2021	Attack detection using ports and source IP	BestFirst search and the WrapperSubsetEval feature selection technique
Makuvaza et al., 2021	Single-vector attacks converted into multi-vector attacks.	Real-time DDoS attack detection using deep neural network (DNN) solution
Patidar et al., 2021	Threats to security reviewed	Detection methods based on information theory
Valdovinos et al., 2021	Prior to determining potential future research areas, it is important to understand the current security concerns related to SDN and the application of security solutions	Network virtualization, cryptocurrencies, monitoring activities, higher throughput, and dynamic target defence are new methodologies and strategies for DDoS detection

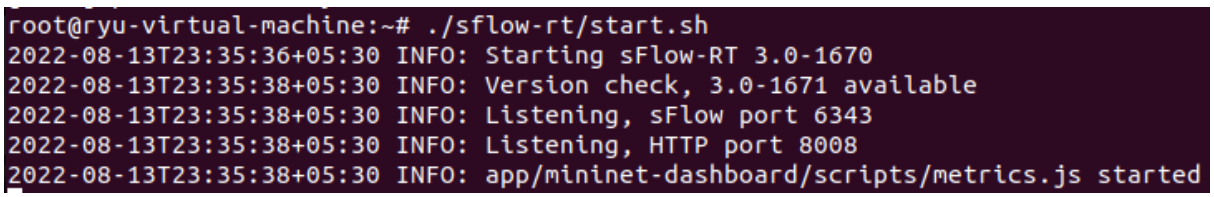

Fig. 3. The sFlow initialization with ODL controller

Table 2. Details of the machine used in the experiment

Virtual Machine	IP addresses	Varied Scenario
Mininet (VM-1)	192.168.208.121	64-bit Ubuntu20.04.1 VM with 5.5 GB RAM Intel Core i3-4700MQ CPU processor
ODL, Ryu, and sFlow (VM-2)	192.168.174.129 and 127.0.0.1	64-bit Ubuntu20.04.1 VM with 5.5 GB RAM Intel Core i3-4700MQ CPU processor
Kali Linux (VM-3)	192.168.253.130	64-bit Ubuntu20.04.1 VM with 5.5 GB RAM Intel Core i3-4700MQ CPU processor

ODL provides an environment with multiple controllers for experimentation. A substantial platform (Shah et al., 2022) with several VM-2 plugins and features is available from ODL and Ryu. The features of both controllers are equivalent in their respective fields; however, when compared, ODL's features are superior to Ryu's in areas like documentation, graphical user interface (GUI), regular updating, modularity, activity, etc. The VM-2 incorporates sFlow, an open-source network intrusion prevention system (Wang et al., 2020; Laghari et al., 2022). The Kali Linux operating system that the VM-3 uses also has a penetration tool for efficient DDoS operations. sFlow can do real-time traffic analysis and packet tracking for IP networks created in VM-2, as shown in Fig. 3.

Table 2 displays a variety of machines with varying hardware specs utilised for experiments. It has the ability to find and compare data, analyse a wide range of protocols, and find a wide range of threats and investigations.

To keep track of switched high-speed networks, the industry uses this technology extensively. Whole network utilisation (Shakil et al., 2022) insight is provided, allowing for performance improvement, usage-based billing, and security threat mitigation. Using sFlow.org, end users, suppliers of network gear, and makers of software can more easily implement sFlow.

The several computers used in experiments, each with a particular hardware configuration. We have employed a data-centric tree topology and a linear topology with varying numbers of hosts and switches for our experimentation. In our experiment, we use an open-source DDoS penetration tool to first determine whether either controller is vulnerable to these attacks, and then we compare the results in various network situations with various parameters. Hping3 is a tool and packet analyser for various TCP/IP packets that is command-line focused. The ping software package command has an impact on the interface, however Hping doesn't solely support ICMP echo requests. It has various options, including a traceroute mode and the capacity to deliver data over a secure communication channel. The UDP, ICMP, and RAW-IP protocols are supported. The penetration tool (Aslam et al., 2022) bombarded the controllers with a large number of data packets. In each particular network state, they are increased by 2,00,000 packets per second. When evaluating this parameter, it is also necessary to look at the type of DDoS attack. Penetration tools generate several attack kinds. During our testing, we have used UDP, ICMP, and TCP SYN assaults. It is crucial to determine when the SDN controllers went offline once the DDoS attack was successfully initiated. Examining the moment, the SDN controllers failed is one of the key criteria for our assessment. A crucial factor is how long it takes the sFlow to receive notifications when the controllers are overloaded with traffic. The RTT (Yaser et al., 2022) between hosts in each situation is evaluated. Furthermore, looked at is the packet loss percentage in various circumstances with varied host counts and traffic types. Initially, using the mininet emulation tool, data-centric tree topology and linear topology are constructed. In our testing, we use a variety of network settings.

4. RESULT AND DISCUSSION

Virtual computers running Linux were used for all simulations and experiments. Kali Linux and the SDN controller have IP addresses of 192.168.253.130 and 127.0.0.1, respectively, and they are both connected to the same network. Three different attack types—UDP, ICMP, and TCP SYN attacks—have been explored during research. This penetration program (Jia et al., 2022) was launched from the Kali Linux virtual machine and effectively breached the DDoS attacks on VM-2. For all Flood assaults on port 8181, Hping3 is used. SDN controllers and Kali Linux are connected to a single network and own IP addresses. The victim host is pounded with random UDP packets when the production of random UDP (Ali et al., 2023; Anyanwu et al., 2023) packets occur. To generate traffic for the victim, the target's IP address must be identified, and source and destination ports initialized. Different IP packets are produced each time. The low orbit ion cannon (LOIC), an open-source C sharp programme for network stress testing and DoS threats, is used to do this. Once the IP packets have been produced, they must be sent to the IP address victim within the allotted time frame. The normal traffic (Bawany et al., 2017) flow rate was 1300 packets per second until DDoS attacks began to penetrate the system, as shown in Fig. 4 and Fig. 5. Evidently, 1300 packets per second was the highest pace of normal traffic sent. Once traffic from the penetration tool begins to flow through the network, massive amounts of TCP, ICMP, and

UDP traffic are blasted in the direction of the controllers (Conti et al., 2017). After configuring, start the flow in test mode. By default, the sFlow start up displays pre-defined ports.

Over 1300 packets per second were hurled at the SDN controllers in a variety of circumstances following the penetration tool's successful execution of DDoS attacks (Meti et al., 2017; Cao and Bian, 2021), finally knocking them offline and preventing them from performing any functions.

As previously noted, we have tried various DDoS attacks (Akbaripour et al., 2015) on these controllers, and successful attacks unmistakably bring the device down. sFlow gathers sample packets from network traffic during an attack, examines any inappropriate activity, creates management policies, and informs the controller of those rule.

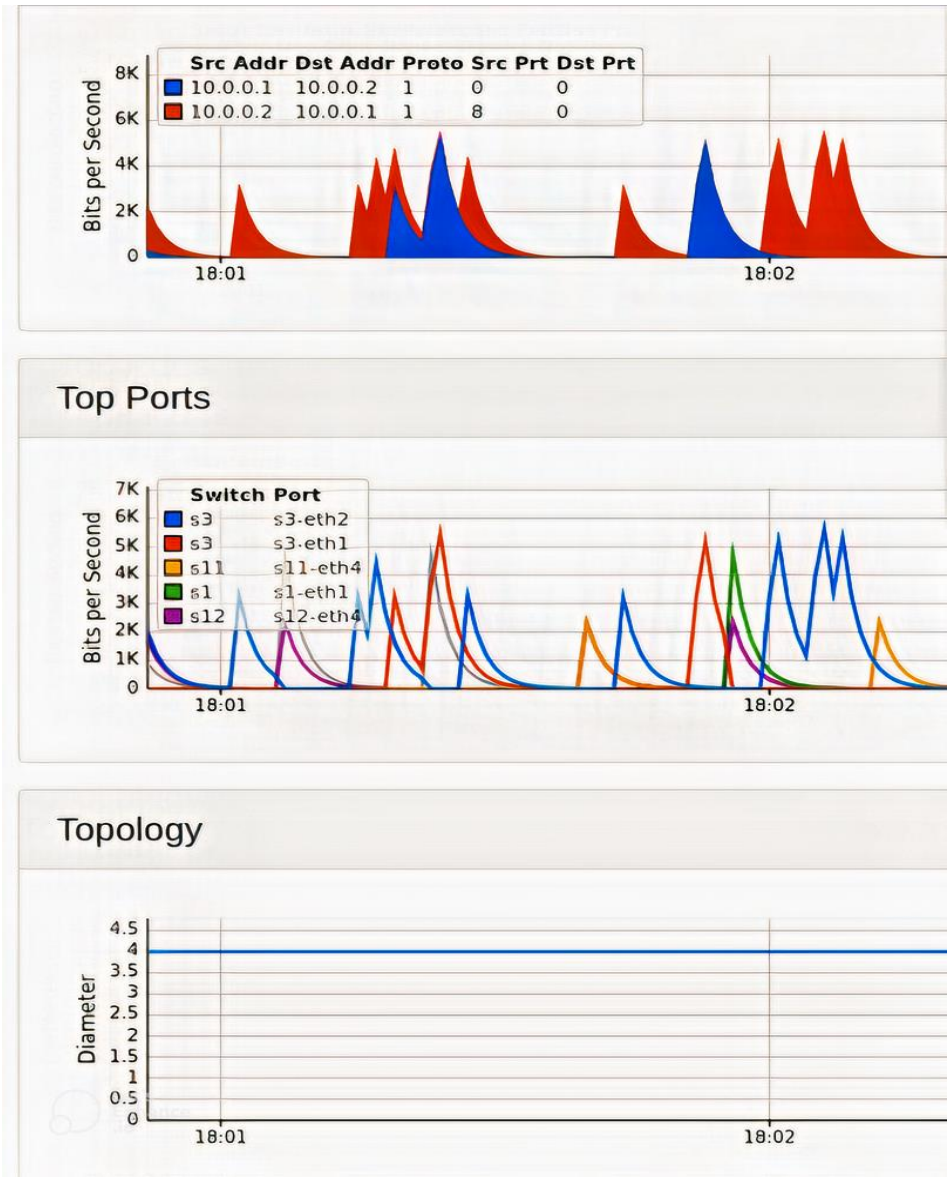

Fig. 4. Normal traffic with Ryu controller on tree topology

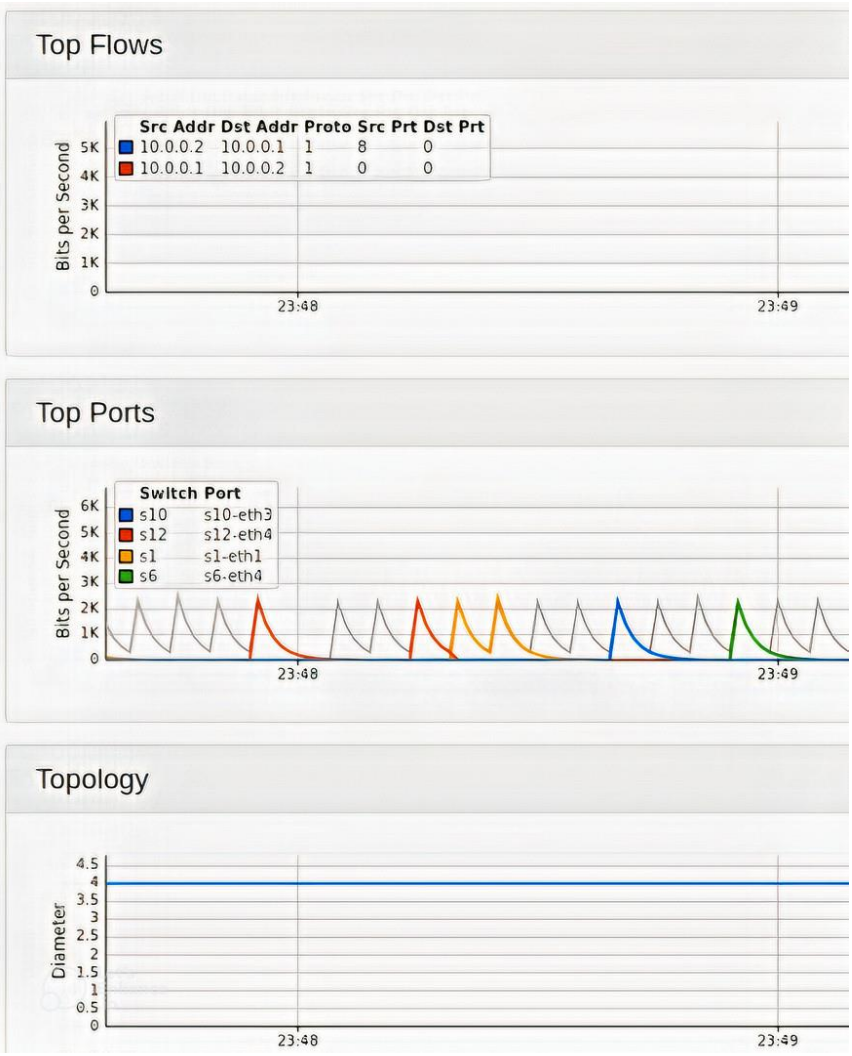

Fig. 5. Normal traffic with ODL controller on tree topology

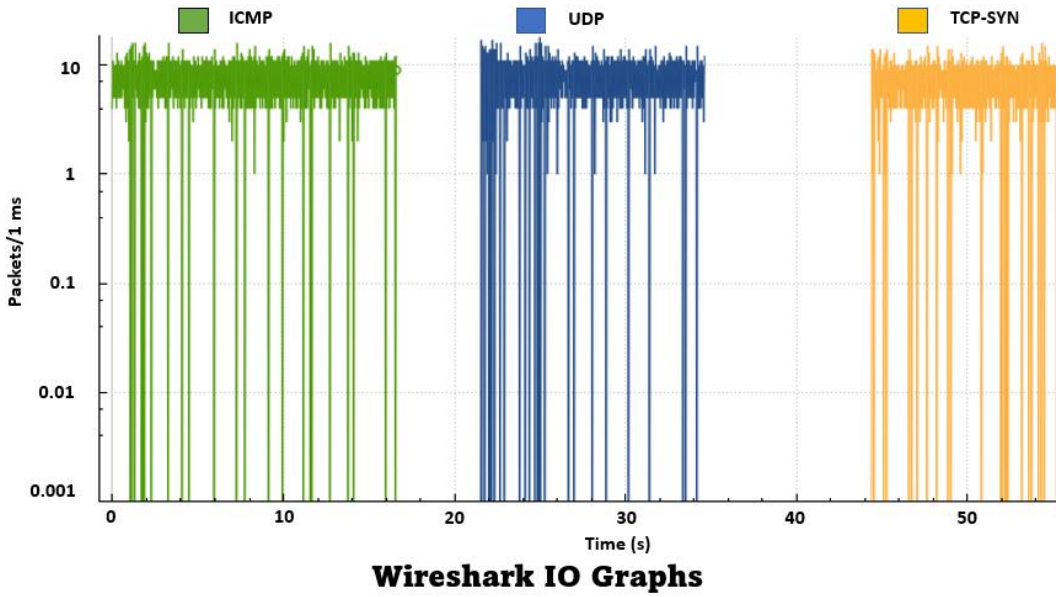

Fig. 6. Wireshark result of tree topology with ODL controller

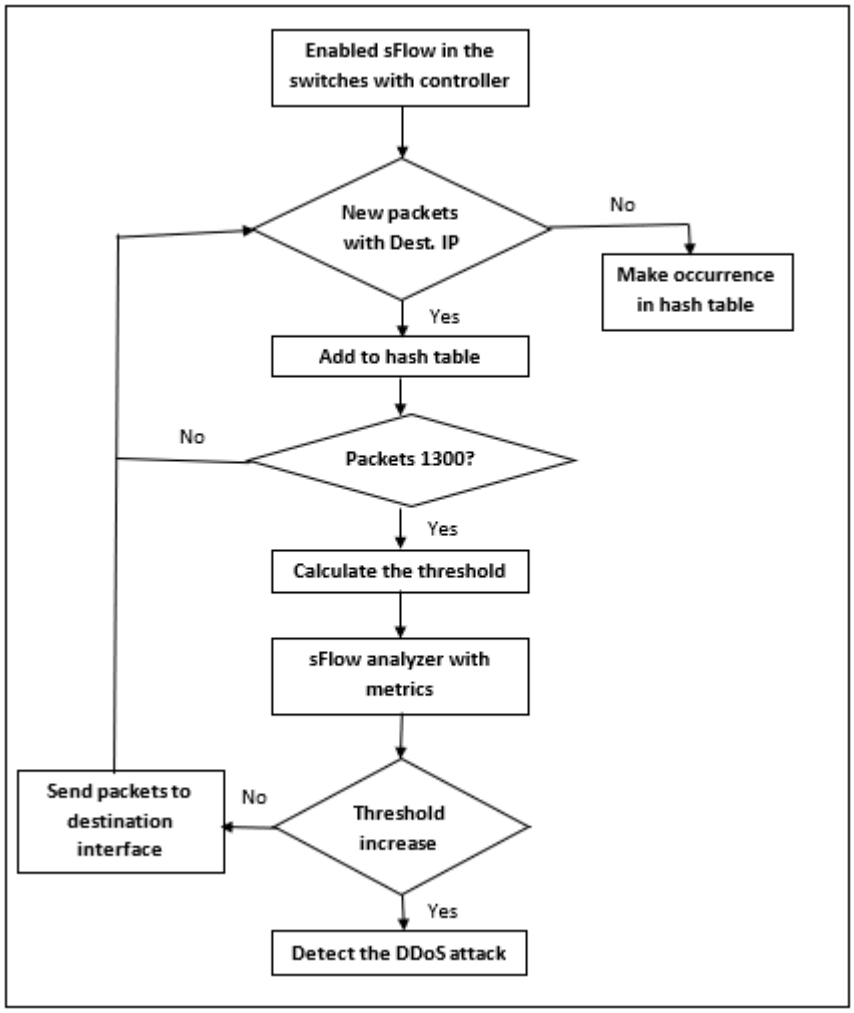

Fig. 7. DDoS detection flowchart

Wireshark was used to capture log files from the network's live traffic. Fig. 6 depicts the volume of typical traffic. The legitimate traffic sent each second is included in this.

The flowchart for our detection method, which uses switches with SDN controllers and the sFlow tool, is shown in Fig. 7. To determine if the destination IP address has an instance in our window, we examine the destination IP address and raise the count if it does. If not, a new IP address will be assigned.

The next step is to determine if there are 1300 packets. The sFlow analyser with metrics compares the window's entropy to the threshold. If it exceeds the threshold, the sFlow analyser returns to step one and waits for more packets. If the entropy is less than the threshold, the sFlow analyser raises the count for subsequent entropies that are also less than the threshold. If the threshold count rises, the presence of an assault is discovered. In an algorithm for computing entropy, lists of statistics are added to the controller and function to get statistics on destination IP addresses, as shown in the Fig. 8.

Table 3 shows that the DDoS detection time grows with the rate of network traffic for the various situations and parameters employed. We have also taken care of other crucial aspects while the DDoS attempts were being detected. The most crucial information is that, in the initial situation, where just 1,00,000 packets were flooded and a variety of hosts, switches, and linear and tree topologies were utilised, each attack resulted in a different degree of packet loss. The linear topology had fewer hosts, resulting in higher packet loss.

Parameters used for various circumstances are displayed in Table 4. ODL and Ryu, two SDN controllers (Bawany et al., 2017), are vulnerable to DDoS assaults. With ODL and Ryu, the maximum rate of transmission flooding the controllers was 1,00,000 packets per second, and the detection time for that increased.

Collect flow statistics from the SDN switches:

- Let s_i be the flow statistics for switch i
- Let p_i,j be the packet count for flow j on switch i
- Let b_i be the byte count for all flows on switch i

Compute the threshold ($a = 1300$) of each flow on each switch using the formula:

$$a(s\_i,j) = -p\_i,j * log(p\_i,j / b\_i)$$

Compute the average entropy for each switch:

$$Avg\text{Threshold}(s\_i) = (1 / |S\_i|) * \sum (j = 1 \; to \; |S\_i|) \; a(s\_i,j)$$

Compute the standard deviation of the entropy for each switch:

$$StdDevThreshold(s\_i) = sqrt((1 / |S\_i|) * \sum(j = 1 \; to \; |S\_i|) \; (a(s\_i,j) - AvgThreshold(s\_i))\text{^}2)$$

Identify a threshold value for the entropy:

$$Total\ Threshold = AvgEntropy(s) + k * StdDevEntropy(s)$$

- k is a constant value chosen based on the network configuration and the desired level of sensitivity for the DDoS detection system

Monitor the threshold values for each switch over time:

- If the entropy value for a switch exceeds the threshold, the switch is marked as a potential target for a DDoS attack

Aggregate the potential targets across all switches:

Fig. 8. An algorithm for computing entropy with lists of statistics added to the controller

Table 3. The outcomes of various scenarios

Type of attack	Controller	No. of Packets/ second	Topology (no. of hosts & switches)	Time to identify a DDoS attack (in seconds)	Round trip time (RTT) (in seconds)	Packet loss
ICMP	ODL	1,00,000	Tree (27 hosts and 13 switches)	2	63.28	74.2%
ICMP	Ryu	1,00,000		5.3	245.56	83%
UDP	ODL	3,00,000		2.5	0	100%
UDP	Ryu	3,00,000		5.8	0	100%
TCP-SYN	ODL	5,00,000		3.7	0	100%
TCP-SYN	Ryu	5,00,000		6	0	100%
ICMP	ODL	1,00,000	Linear (25 hosts and 20 switches)	3.3	59.11	93.8%
ICMP	Ryu	1,00,000		4	118.489	97%
UDP	ODL	3,00,000		3.9	0	100%
UDP	Ryu	3,00,000		5.2	0	100%
TCP-SYN	ODL	5,00,000		5.1	0	100%
TCP-SYN	Ryu	5,00,000		6	0	100%

Table 4. Comparison of our approach with other existing approaches

Authors	Year	Controllers	Detection and prevention of DDoS attack	Using tools	Platform	Wireshark	RTT
Ruchel et al.	2022	ODL and ONOS	-	×	-	×	×
Singh et al.	2022	ODL and ONOS	×	×	Mininet	✓	✓
Badotra et al.	2022	ODL and ONOS	✓	×	-	×	×
Rodriguez et al.	2022	ODL and ONOS	×	×	Mininet	×	×
Ganesan et al.	2022	ODL and OpenKilda	×	Ping tool, Cbench, and OFNet	-	×	✓
Kumar et al.	2022	ODL	✓	SNORT and Flow	Mininet	×	×
Badotra et al.	2021	ODL and ONOS	Detection of DDoS attack	SNORT IDS	Mininet	✓	✓
Cajas et al.	2021	ODL	Detection of DoS attack	×	Mininet	×	×
Lunagariya et al.	2021	Ryu, ODL, FloodLight, Beacon, IRIS, ONOS, OpenMUL, Mastero, POX, NOX	-	×	Mininet, iperf, Gnuplot	×	×
Smida et al.	2020	POX, Floodlight, ONOS and ODL	-	×	Iperf and Mininet-wifi	×	×
Ali et al.	2020	ODL-CO	×	LB algorithm	-	×	×
Amiri et al.	2020	NOX, POX, Beacon, Floodlight, Ryu, ODL, and ONOS	×	×	×	×	×
Badotra et al.	2020	ODL and ONOS	-	×	Mininet emulation tool	✓	✓
Abdullah et al.	2020	POX, Ryu, and ODL	Detection of DoS attack	×	Hping3, iperf, jperf,and miniedit	✓	×
Uddin et al.	2020	ODL, POX, Floodlight, Ryu	-	×	-	×	✓
Latah et al.	2020	POX, Ryu, Floodlight, ODL and ONOS	DoS/DDoS attacks	×	×	×	×
Chauhan et al.	2019	POX, Open vSwitch (OVS) and ODL	-	×	×	×	✓
Chaipet et al.	2019	ODL and ONOS	×	×	-	×	×
Our Approach	2023	ODL and Ryu	Detection and mitigation of DoS attack	✓	Mininet	✓	✓

Table 4 shows that only the suggested model with an ODL controller (Gadze et al., 2021) has been examined between 2022 and 2019. This suggests that very few authors have used the ODL and Ryu controllers in conjunction to produce similar work or outcomes. This paper will be helpful for beginning researchers and present a new area of study.

The author of Badotra and Panda (2021) conducted a real-world experiment using the mininet emulation tool with the following parameters with value such as year is 2021, type

of attack is transmission control protocol synchronize sequence numbers (TCP SYN) and hypertext transfer protocol (HTTP), controller is ODL, number of packets per second is fifty thousand, topology (number of hosts and switches) is fifty. Round trip time (RTT) (in seconds) value and packet loss (in percentage) are two results that the author has analysed with SNORT. The outcome is that the RTT value is 1097.6 and the packet loss is 97.9%. Using the same experimental set-up and sFlow, we also worked on the same parameter and assessed the outcomes. According to our research, the RTT is 863.07 and the packet loss is 98.6%. Through this experiment, the RTT value has lowered, but the packet loss is still considerable. The difference between the values of our RTT and this author's RTT shows that our experiment produced better results and that the sFlow tool outperforms SNORT

Singh et al. (2022) performed a real-world experiment using the mininet emulator tool and parameters with value including year is 2021, controller is ODL, topology is linear, the number of hosts and switches is 16. Latency (ms), jitter (ms), and throughput (server) value are three results that the author has analysed. The outcome is that the latency value is 0.455, the jitter value is 0.131, and the throughput is 2.55. The results were examined using the same experimental design and configurations. Our analysis shows that the corresponding values for latency, jitter, and throughput are 0.90, 0.48, and 2.87. We found that the latency and jitter values are very low, and the throughput value is high. The usage of distinct servers for each system could be one reason for this improvement in throughput. Our experiment yielded better findings, as seen by the disparity between the values of our research and the author's result.

5. CONCLUSION AND FUTURE SCOPE

SDN is growing in popularity because of all of its advantages, but it also has security concerns. An entry point for attackers is created by the control plane's separation from the data plane, which can lead to DDoS attacks (Dissanayake et al., 1997; Gupta et al., 2022a). The first step is to identify them before the network administrator may take any mitigation measures. In this paper, the ODL and Ryu are used to construct a DDoS detection system utilizing sFlow. To evaluate the effectiveness of the deployed DDoS detection programme, a variety of scenarios are employed with varying numbers of hosts, switches, and generated data traffic. A penetration tool is used to create traffic, and a number of hosts and switches are emulated using the Mininet utility.

On the basis of the chosen parameters, the DDoS detection tool was evaluated, and it was found that ODL detects DDoS attacks faster and goes offline before Ryu. There are numerous researchers studying SDN security with sFlow (Hu et al., 2017; Lawal and Nuray, 2018; Vishnu, 2019; Abou et al., 2020; Kumar et al., 2020; Tayfour and Marsono, 2020; Mani and Nene, 2021; Vishnu and Singh, 2021; Hyder et al., 2023), but no research has utilized sFlow with ODL and Ryu. This research's new path could lead to a DDoS prevention framework (Gupta et al., 2022b). In order to have an adequate understanding of the performance analysis of these controllers, we intend to continue developing this research with additional parameters and APIs and clustering with multiple controllers. We propose to construct a real-time training dataset for SDN, use extra algorithms or methodologies, and perform outcome analysis by employing real-time techniques for enhancing traffic flow. SDN will develop into a more flexible, autonomous, and secure system in the not-too-distant future.

REFERENCES

Abdullah, A.F., Salem, F.M., Tammam, A., Azeem, M.H.A. 2020. Performance analysis and evaluation of software defined networking controllers against denial-of-service attacks. Journal of Physics: Conference Series, 1447, 012007.

Abou E.H.Z., Khoukhi, L., Hafid, A.S. 2020. Bringing intelligence to software defined networks: Mitigating DDoS attacks. IEEE Transactions on Network and Service Management, 17, 2523–2535.

Akbaripour, H., Houshmand, M., Fatahi Valilai, O. 2015. Cloud-based global supply chain: A conceptual model and multilayer architecture. Journal of Manufacturing Science and Engineering, 137(4), 040913.

Ali, M.N., Imran, M., din, M.S.U., Kim, B.S. 2023. Low-rate DDoS detection using weighted federated learning in SDN control plane in IoT network. Applied Sciences, 13, 1431.

Ali, T.E., Morad, A.H., Abdala, M.A. 2020. Traffic management inside software-defined data centre networking. Bulletin of Electrical Engineering and Informatics, 9, 2045–2054.

Amiri, E., Alizadeh, E., Rezvani, M.H. 2020. Controller selection in software defined networks using best-worst multi-criteria decision-making. Bulletin of Electrical Engineering and Informatics, 9, 1506–1517.

Anyanwu, G.O., Nwakanma, C.I., Lee, J.M., Kim, D.S. 2022. Optimization of RBF-SVM Kernel using grid search algorithm for DDoS attack detection in SDN-based VANET, IEEE Internet of Things Journal, 10(10), 8477–8490.

Anyanwu, G.O., Nwakanma, C.I., Lee, J.M., Kim, D.S. 2023. RBF-SVM kernel-based model for detecting DDoS attacks in SDN integrated vehicular network. Ad Hoc Networks, 140, 103026.

Aslam, M., Ye, D., Tariq, A., Asad, M., Hanif, M., Ndzi, D., Chelloug, S.A., Elaziz, M.A., Al-Qaness, M.A., Jilani, S. F. 2022. Adaptive machine learning based distributed denial-of-services attacks detection and mitigation system for SDN-enabled IoT. Sensors, 22, 2697.

Badotra, S., Tanwar, S., Bharany, S., Rehman, A.U., Eldin, E.T., Ghamry, N.A., Shafiq, M. 2022. A DDoS

vulnerability analysis system against distributed SDN controllers in a cloud computing environment. Electronics, 11, 3120.

Badotra, S., Panda, S.N. 2020. Evaluation and comparison of Opendaylight and open networking operating system in software-defined networking. Cluster Computing, 23, 1281–1291.

Badotra, S., Panda, S.N. 2021. SNORT based early DDoS detection system using Opendaylight and open networking operating system in software defined networking. Cluster Computing, 24, 501–513.

Bawany, N.Z., Shamsi, J.A., Salah, K. 2017. DDoS attack detection and mitigation using SDN: methods, practices, and solutions. Arabian Journal for Science and Engineering, 42, 425–441.

Batool, S., Khan, F.Z., Ali Shah, S.Q., Ahmed, M., Alroobaea, R., Baqasah, A.M., Ali, I., Ahsan Raza, M. 2022. Lightweight statistical approach towards TCP SYN flood DDoS attack detection and mitigation in SDN environment. Security and Communication Networks, 2022, 2593672.

Cajas, C.D., Budanov, D.O. 2021. Mitigation of denial-of-service attacks using Opendaylight application in software-defined networking. 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus), 260–265.

Cao, Y., Bian, Y. 2021. Improving the ecological environmental performance to achieve carbon neutrality: The application of DPSIR-improved matter-element extension cloud model. Journal of Environmental Management, 293, 112887.

Carvalho, R.N., Costa, L.R., Bordim, J.L., Alchieri, E.A. 2021. Detecting DDoS attacks on SDN data plane with machine learning. 2021 Ninth International Symposium on Computing and Networking Workshops (CANDARW), 138–144.

Chaipet, S., Putthividhya, W. 2019. On studying of scalability in single-controller software-defined networks. 2019 11th International Conference on Knowledge and Smart Technology (KST), 158–163.

Chauhan, N., Sood, M. 2019. Performance analysis of POX, open vswitch and open day light SDN controllers on cloud. International Journal of Innovative Technology and Exploring Engineering, 8, 332–339.

Conti, M., Gangwal, A., Gaur, M.S. 2017. A comprehensive and effective mechanism for DDoS detection in SDN. 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), 1–8.

Dehkordi, A.B., Soltanaghaei, M., Boroujeni, F.Z. 2021. A hybrid mechanism to detect DDoS attacks in software defined networks. Majlesi Journal of Electrical Engineering, 15(1), 1–8.

Dissanayake, M.B., Kumari, A.L.V., Udunuwara, U.K.A. 2021. Performance comparison of ONOS and ODL controllers in software defined networks under different network typologies. Journal of Research Technology & Engineering, 2(3), 94–105.

Gadze, J.D., Bamfo-Asante, A.A., Agyemang, J.O., Nunoo-Mensah, H., Opare, K.A.B. 2021. An investigation into the application of deep learning in the detection and mitigation of DDoS attack on SDN controllers. Technologies, 9, 14.

Ganesan, N., Thangaraju, B. 2022. Performance analysis of SDN controllers within an OpenStack infrastructure. 2022 IEEE India Council International Subsections Conference (INDISCON), 1–7.

Gupta, N., Maashi, M.S., Tanwar, S., Badotra, S., Aljebreen, M., Bharany, S. 2022b. A comparative study of software defined networking controllers using mininet. Electronics, 11, 2715.

Gupta, N., Tanwar, S., Badotra, S., Behal, S. 2022a. Performance analysis of SDN controller. International Journal of Performability Engineering, 18(8), 537–544.

Gupta, V., Kochar, A., Saharan, S., Kulshrestha, R. 2019. DNS amplification based DDoS attacks in SDN environment: Detection and mitigation. 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), 473–478.

Haider, S., Akhunzada, A., Mustafa, I., Patel, T.B., Fernandez, A., Choo, K.K.R., Iqbal, J. 2020. A deep CNN ensemble framework for efficient DDoS attack detection in software defined networks. IEEE Access, 8, 53972–53983.

Hu, D., Hong, P., Chen, Y. 2017. FADM: DDoS flooding attack detection and mitigation system in software-defined networking. 2017 IEEE global communications conference, 1–7.

Hyder, M.F., Fatima, T., Arshad, S. 2024. Towards adding digital forensics capabilities in software defined networking based moving target defense. Cluster Computing, 27, 893–912.

Jia, K., Liu, C., Liu, Q., Wang, J., Liu, J., Liu, F. 2022. A lightweight DDoS detection scheme under SDN context. Cybersecurity, 5, 27.

Jiang, S., Yang, L., Gao, X., Zhou, Y., Feng, T., Song, Y., Liu, K., Cheng, G. 2022. Bsd-guard: Acollaborative blockchain-based approach for detection and mitigation of SDN-targeted DDoS attacks. Security and Communication Networks, 2022, 1608689.

Joëlle, M.M., Park, Y.H. 2018. Strategies for detecting and mitigating DDoS attacks in SDN: A survey. Journal of Intelligent & Fuzzy Systems, 35, 5913–5925.

Jumani, A.K., Laghari, R.A. 2021. Review and state of art of fog computing. Archives of Computational Methods in Engineering, 1–13.

Kumar, C., Kumar, B.P., Chaudhary, A., Gupta, A., Dev, K., Sharma, A., Srivastava, S., Rajitha, B. 2020. Intelligent DDoS detection system in software-defined networking (SDN). 2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), 1–6.

Kumar, P., Baliyan, A., Prasad, K.R., Sreekanth, N., Jawarkar, P., Roy, V., Amoatey, E.T. 2022. Machine learning enabled techniques for protecting wireless sensor networks by estimating attack prevalence and device deployment strategy for 5G networks. Wireless Communications and Mobile Computing, 2022, 5713092.

Kumar, P., Tripathi, M., Nehra, A., Conti, M., Lal, C. 2018. SAFETY: Early detection and mitigation of TCP SYN flood utilizing entropy in SDN. IEEE Transactions on Network and Service Management, 15, 1545–1559.

Laghari, A.A., He, H., Khan, A., Laghari, R.A., Yin, S., Wan, J. 2022. Crowdsourcing platform for QoE evaluation for cloud multimedia services. Computer Science and Information Systems, 19, 1305–1328.

Laghari, A.A., Zhang, X., Shaikh, Z.A., Khan, A., Estrela, V.V., Izadi, S. 2023. A review on quality of experience (QoE) in cloud computing. Journal of Reliable Intelligent Environments, 1–15.

Latah, M., Toker, L. 2020. An efficient flow-based multi-level hybrid intrusion detection system for software-defined networks. CCF Transactions on Networking, 3, 261–271.

Lawal, B.H., Nuray, A.T. 2018. Real-time detection and mitigation of distributed denial of service (DDoS) attacks in software defined networking (SDN). 2018 26th Signal Processing and Communications Applications Conference (SIU), 1–4.

Lunagariya, D., Goswami, B. 2021. A comparative performance analysis of stellar SDN controllers using emulators. 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), 1–9.

Makuvaza, A., Jat, D.S., Gamundani, A.M. 2021. Deep neural network (DNN) solution for real-time detection of distributed denial of service (DDoS) attacks in software defined networks (SDNs). SN Computer Science, 2, 1–10.

Mani, S., Nene, M.J. 2021. Preventing distributed denial of service attacks in software defined mesh networks. 2021 International Conference on Intelligent Technologies (CONIT), 1–7.

Meti, N., Narayan, D.G., Baligar, V.P. 2017. Detection of distributed denial of service attacks using machine learning algorithms in software defined networks. 2017 international conference on advances in computing, communications and informatics (ICACCI), 1366–1371.

Mishra, A., Gupta, N. 2022. Supervised machine learning algorithms based on classification for detection of distributed denial of service attacks in SDN-enabled cloud computing. Cyber Security, Privacy and Networking: Proceedings of ICSPN 2021, 165–174.

Patidar, S., Singh, S. 2021. Information theory-based techniques to detect DDoS in SDN: A survey. 2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC), 529–534.

Pattanaik, A., Gupta, A., Kanavalli, A. 2019. Early detection and diminution of DDoS attack instigated by compromised switches on the controller in software defined networks. 2019 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), 1–5.

Rodriguez, A., Quiñones, J., Iano, Y., Barra, M.A. 2022. A comparative evaluation of ODL and ONOS controllers in software-defined network environments. 2022 IEEE XXIX International Conference on Electronics, Electrical Engineering and Computing (INTERCON), 1–4.

Ruchel, L.V., Turchetti, R.C., de Camargo, E.T. 2022. Evaluation of the robustness of SDN controllers ONOS and ODL. Computer Networks, 219, 109403.

Sai, A.D., Tilak, B.H., Sanjith, N.S., Suhas, P., Sanjeetha, R. 2022. Detection and mitigation of low and slow DDoS attack in an SDN environment. 2022 International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), 106–111.

Santos, R., Souza, D., Santo, W., Ribeiro, A., Moreno, E. 2020. Machine learning algorithms to detect DDoS attacks in SDN. Concurrency and Computation: Practice and Experience, 32, e5402.

Shalini, P.V., Radha, V., Sanjeevi, S.G. 2021. DDoS attack detection in SDN using CUSUM. Proceedings of International Conference on Computational Intelligence and Data Engineering: ICCIDE 2020, 301–309.

Shah, S. Q. A., Khan, F. Z., Ahmad, M. 2022. Mitigating TCP SYN flooding based EDOS attack in cloud computing environment using binomial distribution in SDN. Computer Communications, 182, 198–211.

Shakil, M., Mohammed, A.F.Y., Arul, R., Bashir, A.K., Choi, J.K. 2022. A novel dynamic framework to detect DDoS in SDN using metaheuristic clustering. Transactions on Emerging Telecommunications Technologies, 33, e3622.

Sheibani, M., Konur, S., Awan, I. 2022. DDoS attack detection and mitigation in software-defined networking-based 5G mobile networks with multiple controllers. 2022 9th International Conference on Future Internet of Things and Cloud (FiCloud), 32–39.

Singh, A., Kaur, N., Kaur, H. 2022. An extensive vulnerability assessment and countermeasures in open network operating system software defined networking controller. Concurrency and Computation: Practice and Experience, 34, e6978.

Singh, J., Behal, S. 2020. Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges and future directions. Computer Science Review, 37, 100279.

Smida, K., Tounsi, H., Frikha, M., Song, Y.Q. 2020. Efficient SDN controller for safety applications in SDN-based vehicular networks: POX, floodlight, ONOS or OpenDaylight? 2020 IEEE Eighth International Conference on Communications and Networking (ComNet), 1–6.

Sritharan, K., Elagumeeharan, R., Nakkeeran, S., Mohamed, A., Ganegoda, B., Yapa, K. 2022. Machine learning based distributed denial-of-services attacks detection and mitigation testbed for SDN-enabled IoT devices. 2022

13th International Conference on Computing Communication and Networking Technologies (ICCCNT), 1–6.

Swami, R., Dave, M., Ranga, V. 2023. IQR-based approach for DDoS detection and mitigation in SDN. Defence Technology, 25, 76–87.

Tayfour, O.E., Marsono, M.N. 2020. Collaborative detection and mitigation of distributed denial-of-service attacks on software-defined network. Mobile Networks and Applications, 25, 1338–1347.

Uddin, R., Monir, M.F. 2020. Evaluation of four SDN controllers with firewall modules. Proceedings of the International Conference on Computing Advancements, 1–8.

Valdovinos, I.A., Pérez-Díaz, J.A., Choo, K.K.R., Botero, J.F. 2021. Emerging DDoS attack detection and mitigation strategies in software-defined networks: Taxonomy, challenges, and future directions. Journal of Network and Computer Applications, 187, 103093.

Valizadeh, P., Taghinezhad-Niar, A. 2022. DDoS attacks detection in multi-controller based software defined network. 2022 8th International Conference on Web Research (ICWR), 34–39.

Varghese, J.E., Muniyal, B. 2021. Trend in SDN architecture for DDoS detection-a comparative study. 2021 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), 170–174.

Vishnu Priya, A. 2019. Reinforcement learning-based DoS mitigation in software defined networks. ICCCE 2018: Proceedings of the International Conference on Communications and Cyber Physical Engineering 2018, 393–401.

Vishnu Priya, A., Singh, H.K. 2021. Mitigation of ARP cache poisoning in software-defined networks. Advances in Smart System Technologies. 85–94.

Wang, X., Yin, S., Li, H., Wang, J., Teng, L. 2020. A network intrusion detection method based on deep multi-scale convolutional neural network. International Journal of Wireless Information Networks, 27, 503–517.

Yaser, A.L., Mousa, H.M., Hussein, M. 2022. Improved DDoS detection utilizing deep neural networks and feedforward neural networks as autoencoder. Future Internet, 14, 240.

Yin, S., Li, H., Teng, L., Laghari, A.A., Estrela, V.V. 2023. Attribute-based multiparty searchable encryption model for privacy protection of text data. Multimedia Tools and Applications, 1–22.

Yin, S., Li, H., Laghari, A.A., Karim, S., Jumani, A.K. 2021. A bagging strategy-based kernel extreme learning machine for complex network intrusion detection. EAI Endorsed Transactions on Scalable Information Systems, 8(33), e8–e8.