Journal of Information Systems Engineering and Management
2025, 10(51s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

DJRT: A Model for Real-Time Detection & Mitigation of DDoS attacks in Software Defined Networks

[1]Deepak Kumar[0000-0002-1629-0132], [2]Jawahar Thakur[0000-0002-2832-5451]
[1,2]Department of Computer Science, Himachal Pradesh University, Summer-Hill, Shimla-171005 (India)
*Corresponding Author: *Deepak Kumar (Email-ID: deepak.cs339@gmail.com)*

ARTICLE INFO

Received: 18 Dec 2024

Revised: 10 Feb 2025

Accepted: 28 Feb 2025

ABSTRACT

Software-defined networks (SDN) offer significant flexibility, scalability, and dynamic management advantages. However, these networks are increasingly vulnerable to high-rate Distributed Denial of Service (DDoS) attacks. This study investigates the susceptibility of SDNs to such threats and presents a novel model, DJRT (Real-Time Detection and Mitigation of DDoS attacks). The model employs a dual strategy for both detection and mitigation. The objective is to detect multiple DDoS attacks causing elephant flow and mitigate them as the framework detects them, thereby enhancing the SDN security and resilience. The DJRT features a custom-developed script, ryu2m.js, for the real-time detection and mitigation, along with the elephant.py script to identify the route through which elephant flow occurs. The framework uses the sFlow flow-statistic tool to monitor the network traffic, and a mininet emulation virtual SDN environment consisting of 27 hosts, 13 openvswitches, and a RYU controller. The mininet dashboard application provides visualization of the topology used, the connection between switches, and visualization of real-time traffic patterns and performance. The framework's effectiveness was evaluated through three distinct scenarios: In the first scenario, regular traffic was assessed without any detection and mitigation, resulting in a top flow of approximately 22 kbps, with baseline performance metric recorded as an sFlow data rate of 61.5 kbps, 12.7 packet per second (pps), 1.78% CPU usage and 50.04% memory usage. The second scenario involved a single UDP DDoS attack occurring alongside regular traffic, leading to a top flow of about 105 mbps (attacker host 24 to victim host 10). This scenario saw a sharp traffic increase, characterized by a data rate of 25 mbps, a 2.32 kpps packet rate, 30.80% CPU usage, and 70.7% memory usage. However, the attack was successfully detected and mitigated. The third scenario included three simultaneous UDP DDoS attacks occurring with regular traffic. For these attacks, the top flows reached approximately 50 mbps (attacker host 21 to victim host 2), 100 mbps (attacker host 14 to victim host 18), and 150 mbps (attacker host 3 to victim host 27). This results in a significant traffic increase, with a data rate of 36.9 mbps and a packet rate of 3.43 kpps, overwhelming the system and causing CPU usage to peak at 96.90% and memory usage at 87.2%.

Nevertheless, these attacks were also successfully detected and mitigated. The DJRT model effectively detects and mitigates both single and multiple high-rate UDP DDoS attacks. The findings emphasize its effectiveness in mitigating congestion caused by attacks, indicating a potential for significant improvements in security and performance within practical SDN applications. Further investigation is necessary to assess the framework's scalability and its implementation in larger SDNs.

Keywords:DDoS, Elephant flow, High-rate , IDS, RYU, Security, sFlow.

INTRODUCTION

A new paradigm known as Software-Defined Networking (SDN) has emerged, bringing with it a variety of attack vectors and traffic patterns that can adversely affect network performance. To maintain the functionality of the network and to minimize the risk of congestion or network failures, it is essential to identify both elephant flows[1] (EFs refer to a network flow that carries a large amount of data over a significant period, that affects bandwidth, and is easy to predict). Mice flows[2] (MFs refer to smaller short-lived data transfers, and are difficult to forecast).

Journal of Information Systems Engineering and Management
2025, 10(51s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Research conducted within SDN-based data centers indicates that MFs make up approximately 80% of total flows[3], and they can cause jitter, which is responsible for delays. In comparison, 10-15% of EFs constitute the predominant share of total traffic. Research[4] indicates that EFs can rapidly overload network device buffers, resulting in a drop in the queueing delays and drop in packet delivery. EFs are the primary contributors to network congestion. So, efficiently managing EFs can help to identify the sources of problems more quickly. If we can spot MFs earlier, we can separate them from other real-time data traffic which will help reduce delays and improve network efficiency.

Therefore, accurate identification and scheduling of EFs are essential for addressing issues related to unbalanced link loads and low link utilization in data centers[5]. Studies suggest that cybercriminals leverage botnets to execute DDoS[6] attacks capable of generating traffic volumes that reach terabytes per second. This excessive traffic not only creates EFs but also consumes network resources designated for legitimate packets, rendering these resources unavailable and leading to network congestion.

To effectively detect EFs, it's suitable to use programmable interfaces for applications running on the application layer, taking advantage of the network's capabilities. OpenFlow[7] switches play a vital role in packet forwarding, utilizing flow entries in the flow table, which primarily include counters that simplify the collection of flow statistics for accurate detection of EFs. In recent years, network security threats have profoundly impacted SDN. Among these threats, Distributed Denial of Service (DDoS) attacks have become the most frequent. DDoS attacks take advantage of the table-miss mechanism in SDNs to overload both switches and controllers by generating extensive table-miss traffic [8].

In current SDNs, the major concern is about the variants of DDoS attacks, as these attacks are classified into two categories: high-rate[9] DDoS attack, and low-rate[9] DDoS attack which are are major reasons for the EFs in the network. Both high-rate DDoS and low-rate DDoS attacks both are capable of generating EFs, but their effects, mechanisms and mitigation techniques can be differ.

A. Elephant flow due to low-rate DDoS attack

In the realm of low-rate[10] DDoS attack, EFs occur when a small volume of attack traffic is intentionally transmitted over an extended period, with the aim to induce prolonged congestion within the network. The low-rate nature of DDoS attacks makes them challenging to detect, as the traffic may be less intense in terms of total volume but still generates EFs that are highly impactful. The low-rate DDoS attacks are persistent and are responsible for the increase in latency, reducing Quality of Service (QoS). The Slowloris attack is a suitable example of a low-rate DDoS attack. The mitigation techniques for low-rate DDoS attacks are: rate limiting, flow control and traffic shaping, Intrusion Detection Systems (IDS), and session timeout reduction protocol filtering.

B. Elephant flow due to high-rate DDoS attack

The high-rate[11] DDoS attack generates a high volume of attack traffic over a relatively short period. It can generate EFs more direct with the aim of overwhelming the target's network resources. These attacks flood the target with ICMP, UDP, or TCP SYN. Such attacks are responsible for bandwidth saturation, and device overload. The mitigation techniques for high-rate DDoS attacks are traffic scrubbing, rate limiting, traffic shaping, load balancing, and Anycast routing.

This paper tries to address the critical issue of elephant flows that arise from high-rate UDP DDoS attacks. In response, the research focuses on developing a unified framework for the real-time detection and mitigation of such attacks. The primary objective is to introduce a novel flow statistics-based unified framework designed for the real-time detection, mitigation and visualization of EFs resulting from high-rate UDP DDoS attacks. We proposed a model named DJRT, which operates in real-time and integrates the sFlow tool with openvswitch. The openvswitch acts as sFlow agent, and it continuously monitors and analyzes packet flow based on predefined threshold values. When the traffic flow through openvswitch(say switch s1) exceeds the predefined threshold, an sFlow agent promptly alerts the RYU controller regarding the high-rate UDP DDOS, and in response, the RYU controller makes the flow entries for these attacks, which means attacks are detected and blocked as per the attacker's source IP address. Moreover, the proposed model facilitates the visualization of network topology, allowing the identification

Research Article

of routes and switches that contribute to the generation of elephant flows(EFs). Integration of sFlow and openvswitch with the RYU controller greatly improves the unified detection and mitigation approach. It offers essential information about network activity during these events and this increases the trust in the framework's effectiveness in handling severe high-rate UDP DDoS attacks.

The key contributions of this article include:

-The proposed DJRT is a flow-based approach for detecting and mitigating of high-rate DDoS attacks. Implemented in a virtual SDN environment using the mininet emulator, it effectively detects and mitigates high-rate DDoS attacks in real time. Additionally, it offers a clear view of EFs through the sFlow-RT mininet dashboard application.

- This study aims to evaluate and compare the proposed DJRT model with previous research. Following this evaluation, we will present a comparative review that highlights the differences from earlier studies.

The research presented in this paper is highly relevant as it addresses the limitations of previous studies, which tend to focus either on the detection or mitigation of single DDoS attacks, rather than tackling multiple attacks simultaneously. Moreover, earlier studies often lack visual representation of both regular and high-rate UDP DDoS attack traffic, including key elements such as network topology, top flows, top port information, and other metrics like data rate, packet rate, and details of CPU and memory usage. This lack of comprehensive data complicates the analysis of real-time traffic. The proposed model overcomes these challenges and can address previous issues by providing real-time detection and mitigation of multiple high-rate UDP DDoS attacks. This integrated approach not only enhances detection and mitigation efforts but also effectively manages congestion caused by elephant flows (EFs).

The rest of this paper is organized in the following manner. Section 2 provides related work. In Section 3, we discuss DDoS attack types. Section 4 discusses different techniques for DDoS detection. An overview and introduction to the DJRT model and its components is discussed in Section 5. Section 6 discusses the methodology used. The results and discussion are presented in Section 7. Section 8, where we conduct a thorough comparative evaluation of DJRT with previous studies, will enhance credibility in the validity of our research. Implications and future insights are discussed in Section 9. The conclusion is discussed in Section 10.

RELATED WORK

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore Numerous studies have highlighted how DDoS attacks can significantly hinder network performance, particularly in an SDN environment. In Table 1, we present a comparison of various research efforts, detailing their specific focus area, methodologies, key findings and limitations. A recent study [12] introduced a new technique to improve security in Software-Defined Networking (SDN) by integrating IPsec with sFlow. This method was evaluated in a virtual SDN environment using the Mininet emulator, which was able to detect and respond to attacks rapidly. Another research effort [13] identified five categories of DDoS detection methods: (A) Deep Learning, (B) Entropy-based, (C) Flow Counting, and (D) Hybrid Approaches, showcasing various innovative strategies in this area. Additionally, a novel approach for preventing DDoS attacks was introduced, utilizing the sFlow-RT application. This technique enables multiple RYU controllers to collaborate and exchange strategies for DDoS mitigation through a Redis Simple Message Queue (RSMQ). Test results indicate that this approach reduces the load on the controllers, making it well-suited for various applications and enhancing reliability [14].
In a separate study [15], the author developed an effective method for network load balancing and QoS in SDN. By leveraging the sFlow and RYU, the system remains compatible with traditional multicast applications such as IGMP v2, confirming its deployment in a real environment. It improves path selection and optimizes bandwidth utilization while employing load-balancing techniques to minimize unwanted traffic, reducing latency and packet loss. This focus on practicality enforces the validity of these solutions. Furthermore, a real-time method for detecting and mitigating DDoS attacks that utilizes sFlow was proposed. This approach analyzes each incoming packet within the network and initiates a response to mitigate the attack, proving to be highly efficient in both detection and mitigation. The authors validated the strategy's effectiveness through simulations conducted in a virtual environment[16]. The significance of measurement-based monitoring for identifying network issues was

Research Article

emphasized. This method employs the STRIDE threat model to evaluate the security of sFlow and BigTap tools. The results indicate that the effectiveness of sFlow is dependent upon the configuration of its parameters, further supporting the usefulness of these approaches[17].

In [18], the authors have introduced a novel approach to flow events and employed the Sequential Probability Ratio Test for decision-making with a specified error rate. The approach's credibility is measured using DARPA intrusion-detection datasets and compared to prior studies with metrics such as count, entropy, etc. The results of this study are significant as they provide a new perspective on flow events and decision-making in network security, potentially leading to more effective anomaly detection systems. In [19], the authors conducted a groundbreaking study that assessed the implications of kill-chain models and cyber-attacks on network systems. The study introduces novel and advanced ensemble learning and deep learning approaches for Decision Engines (DEs) and datasets for practical training and validation. Network Anomaly Detection Systems (NADS) such as Cloud Computing, Data Centers, Fog, and IoT were explored—the study with experimental insights and for future research avenues. In [20], the authors improved the packet count tracer in the OpenFlow table by integrating the flow statistics technique and introducing an entropy-based DDoS detection model. This practical solution lightens the controller workload, and due to its minimal computational complexity, the algorithm is easily deployable in programmable switches.

In [21], the authors developed a model using support vector machines (SVMs) to detect anomalies in performance and identify bottlenecks in distributed applications. This technique combines one-class SVM and multiclass SVM to establish a baseline model without requiring prior knowledge. The results from simulations clearly demonstrate its effectiveness in detecting and identifying failures. When comparing this approach to statistical methods such as Decision Trees and Naive Bayes Classifiers, it shows superior performance. In [22], the authors propose an innovative method to mitigate DDoS attacks using an OpenFlow-enabled device. They introduce a flexible and modular framework that utilizes the programmatic feature of SDN within the context of Network Function Virtualization (NFV). The OpenFlow-enabled device plays a vital role by enabling dynamic and efficient network management, which enhances the network's resilience against DDoS attacks. The study in [23] provides a thorough analysis of the security challenges inherent in SDN, emphasizing IDS as a promising solution. The author reviews recent security solutions and assesses various IDS techniques based on both machine learning and deep learning. The research also explores future development and improvements in SDN security, offering a comprehensive overview of the current landscape.

In [24], the author presents an entropy-driven method for enhanced DDoS detection, highlighting practical applications. By utilizing both Shannon and Renyi entropy in a simplified form, they successfully detect the distributed nature of DDoS traffic and assist the RYU framework in filtering out illegitimate traffic. Their snort-based signature detection, applied to data plane traffic collection, effectively minimizes network resource impact. They analyze the processed data to enhance the detection capabilities of Convolutional Neural Networks (CNN) and Stacked Auto-Encoder models, showcasing high accuracy rates in real-world scenarios. The authors in [25] introduce an innovative flow-aware strategy for detecting elephant flows in SDN. This approach deploys two classifiers within switches and controllers, which significantly improves accuracy, F-measure, processing time, and recall performance. By reducing reliance on excessive signaling overhead and classification requests, this method addresses the challenges associated with EF's detection in SDN. In [26], the authors investigate the challenges related to IT security implementation in light of high-speed data transmission resulting from heavy network traffic. They adopt the use of machine learning to enhance detection efforts while minimizing error rates. The study also examines the impact of virtual networking protocols such as GRE, GENVE, and VxLAN on anomaly detection and highlights the importance of data preprocessing, particularly in adverse scenarios.

In [27], the authors introduce an information security-based defense mechanism for the SDN environment. This mechanism integrates the OpenFlow protocol with a sFlow collector to construct an effective Network Intrusion Detection System (NIDS) that addresses DDoS attacks. The findings illustrate how this mechanism can significantly reduce packet loss and strengthen defense capabilities. The research presented in [28] outlines an innovative network monitoring, detection, and mitigation approach, which integrates the sFlow at the controller level, enabling anomaly detection based on user-defined rules and policies. The study uses a mininet emulator,

Journal of Information Systems Engineering and Management
2025, 10(51s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

OpenDaylight controller, and OpenFlow protocol. The author highlights the limitations of early detection approaches in managing DDoS attacks. The machine learning model, which has proven to be a more reliable approach for DDoS detection, instills confidence in its effectiveness. However, the lack of a unified solution to opt for a suitable algorithm remains a challenge. Data sets include data preprocessing, such as data cleaning, transformation, and normalization. Another significant issue is the dependence on outdated data sets such as CAIDA, DARPA, and KDD-99. These data sets are suited for traditional network environments and must offer the dynamism and diversity of today's network traffic. In [29], the author introduces an entropy-based approach for detecting both low-rate DDoS and high-rate DDoS attacks on SDN controllers. The emphasis is on the susceptibility of SDNs to DDoS attacks, which can disrupt network operations. This approach analyzes packet headers to assess traffic randomness and identify unusual patterns. Simulations show that this approach is especially practical for high-rate DDoS attacks.

Table 1: Literature review of previous related work

Ref.	Key Features	Detection Mechanism	Mitigation capability	Attack Detection Focus	Real-Time Performance	Scalability	Limitations
[12]	Integration of IPsec with sFlow for SDN security	Fast detection and response	Prompt mitigation of attacks	DDoS attacks	Real-Time detection with immediate response	Efficient in SDN environment	May be limited by the complexity of Integration between sFlow and IPsec
[13]	Identification of five DDoS detection methods	Deep Learning, Entropy-based, Flow Counting	Various innovative strategies	DDoS detection methods	Classifies different methods	Scalable across various model	No active mitigation or response mechanism
[14]	sFlow-RT application with RYU controllers cooperation	Collaborative DDoS mitigation	Reduces controller burden	DDoS attacks	Real-Time response to DDoS attacks	Scalable, reduces controller overhead	Dependent on network controller configuration
[15]	Network load balancing and QoS in SDN	sFlow and RYU based optimization	Reduces packet loss and delays	Network efficiency	Works effectively for real-time network management	Efficient in high traffic scenario	May not fully address security concerns in QoS management
[16]	Real-time detection and mitigation for DDoS attacks	Incoming packet examination	Effective in countering DDoS attacks	DDoS attacks	Real-time detection and response	Scalable in SDN environment	Detection may miss subtle or evolving attack patterns
[17]	Measurement-based monitoring using STRIDE threat model	Security appraisal of sFlow tools	Identifies network problems	Network security	Real time monitoring based on parameter setting	Dependent on setup configuration	Limited by manual configuration and setup complexity
[18]	Sequential Probability Ratio Test for	Anomaly detection with DARPA	Improved flow event handling	Flow events	Analyzes flow events based on error rates	Applicable in anomaly based systems	May be less effective for real-time or

Journal of Information Systems Engineering and Management
2025, 10(51s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

	flow events decision-making	datasets					large scale environments
[19]	Ensemble learning and deep learning for Decision Engines and NADS	Novel network anomaly detection techniques	Powerful insights for future research	Various network systems	Advanced machine learning methods for anomaly detection	Scalable, handles various data centers, cloud and IoT environments	Require high computational sources and may be slow for real time detection
[20]	Integrated entropy-based DDoS detection model	Packet count tracer with flow statistics	Lightens controller workload	DDoS detection	Effective with less complexity	Highlyscalable in programmable switches	May struggle with false positives in diverse traffic scenarios
[21]	Anomaly detection using support vector machines	One-class and multiclass SVM techniques	Detects performance bottlenecks	Performance issues	Simulation based detection	Efficient with less data but computationally intensive	High computational cost for large datasets
[22]	Flexible framework using OpenFlow within NFV	Dynamic network management	Enhances resilience against DDoS	DDoS attacks	Dynamic defense against attacks	Scalable in SDN environment	Limited by OpenFlow protocol constraints and compatability
[23]	Review of IDS techniques leveraging machine and deep learning	Comprehensive security assessment	Potential developments in SDN security	SDN security	Emphasizes advanced IDS strategies	Scalable with advanced IDS systems	Requires Integration with existing IDS systems may be complex to deploy
[24]	Entropy-driven approach utilizing Shannon and Renyi entropy	Effective DDoS detection	Assists RYU in managing illegitimate traffic	DDoS detection	Effective in Real world scenario	Scalable in SDN environment	Limited by false positives due to signature based detection
[25]	Flow-aware approach for elephant flow detection	Two classifier deployment	Improves detection accuracy	Elephant flow detection	Highly efficient with reduced overhead	Scalable across SDN environments	May not address all types of network traffic anomalies
[26]	\Examination of IT security challenges with machine learning	Machine Learning models for anomaly detection	Lower error rate in anomaly detection	High speed data environments	Consolidates detection with lower error rates	Potential for scalability across diverse network protocols	Requires continuous retaining of model for optimal performance

Journal of Information Systems Engineering and Management
2025, 10(51s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

[27]	Information Security Defense Mechanism integrating OpenFlow with sFlow	Network Intrusion Detection System (NIDS)	Effective DDoS mitigation	DDoS attacks and NIDS functionality	Real-Time DDoS detection and mitigation	Highly scalable for SDN based environments	May face performance degradation in highly dynamic environment
[28]	Anomaly detection using sFlow collector at controller level	User-defined rule-based detection	Mitigates DDoS based on machine learning	DDoS detection	Real Time detection through Machine Lerning	Challenges in dynamic data handling	Requires significant data for training and may be affected by data quality issues
[29]	Entropy based approach for detecting DDoS attack	Entropy collection based on packet header features	Focus on mitigation, no mitigation strategy	Both low-rate DDoS and high-rate DDoS	Effective for high rate DDoS	Scalable to SDN environments, adaptable to varying traffic rates	Struggles with detecting low-rate DDoS attacks targeting multiple victims

Research on DDoS detection and mitigation in Software-Defined Networks (SDNs) reveals several gaps. Many methods focus solely on detection without real-time mitigation, which is crucial for managing active attacks. Most studies address single attack types like UDP or ICMP floods, ignoring the growing prevalence of multi-vector attacks. While some models provide network monitoring and anomaly detection, they often lack traffic flow visualization and insights into attack sources, complicating real-time threat management. Scalability poses a challenge, with many solutions tested in small environments that struggle in large-scale SDNs. Additionally, reliance on predefined thresholds can hinder the detection of subtle, evolving attacks. Machine learning approaches, while promising, frequently encounter high computational costs and false positives, reducing their real-time effectiveness. Many methods separate detection from mitigation, resulting in delayed responses. Although some research on elephant flow detection exists, it is rarely linked to dynamic traffic management, highlighting the need for more adaptive, scalable, and integrated solutions for real-time DDoS defense in SDNs.

CLASSIFICATION OF DDOS ATTACKS

DDoS attacks can be categorized into three main types: volumetric attacks, state-exhaustion attacks, and application attacks, as illustrated in Figure 1.

3. 1. Volumetric attacks

Volumetric attacks [30] are intended to overwhelm a target system by sending a large amount of traffic, which can exhaust its resources and disrupt services. These attacks are generally divided into two main types: **flooding attacks**, which directly flood the target with traffic, and **amplification attacks**, which make use of other systems to increase the volume of traffic hitting the target.

3. 1. 1. Flooding attacks

a. UDP flood

Journal of Information Systems Engineering and Management
2025, 10(51s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

In these types of attacks, the attacker floods random ports on a target host using User Datagram Protocol (UDP) packets directed at the controller. This method exploits the connection less nature of UDP [31].

b. ICMP flood

In Internet Control Message Protocol (ICMP) flood attacks, the attacker sends many echo requests to overwhelm the victim's computer. ICMP [32] attacks can greatly disrupt the availability of the victim's services.

c. Spoofed packet flood

Attackers carry out this kind of DDoS attack by flooding their target with fake data packets [33], which aims to slow down the target's ability to respond quickly.

3. 1. 2. Amplification attacks

The term 'amplification attack' describes a strategy where an attacker increases their impact by using an amplification factor, allowing them to disrupt a wider array of targets while using fewer resources. Key examples include the UDP amplification (Fraggle Attack) and DNS amplification.

a. DNS amplification

This attack exploits open DNS [34] resolver to flood a target server with high traffic volumes through reflective techniques, making it inaccessible and disrupting its services.

b. UDP amplification

Fraggle attacks cause denial-of-service by bombarding a router's broadcast address with spoofed UDP[35] traffic. This method resembles a Smurf attack but primarily uses spoofed ICMP traffic instead of UDP.

3. 2. State exhaustion attacks

These types of attacks, often referred to as protocol-based attacks, focus on disrupting network devices like firewalls and load balancers. Attackers may also target TLS endpoints, which can hinder legitimate users from accessing the service.

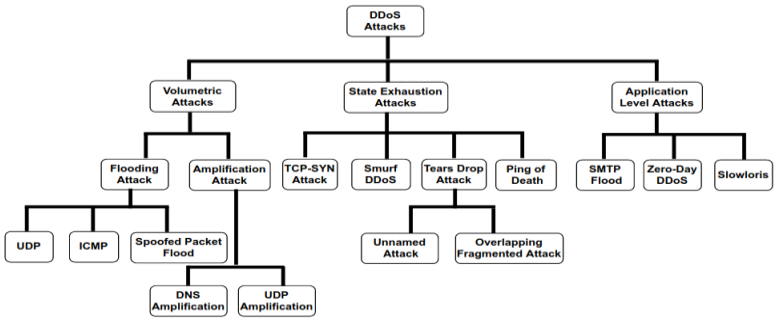

Figure 1: Classification of DDoS attacks

3. 2. 1. TCP-SYN

TCP-SYN [36] attacks utilize the TCP 3-way handshake process to overload the target server's resources, rendering it unresponsive. In this type of attack, the attacker floods the targeted machine with a high volume of TCP connection requests.

3. 2. 2. Smurf

Smurf [37] attacks incapacitate computer networks by exploiting weaknesses in the Internet Protocol (IP) and Internet Control Message Protocol (ICMP). This attack overwhelms the network, causing it to become non-functional.

3. 2. 3. Tear Drop

Journal of Information Systems Engineering and Management
2025, 10(51s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

A tear drop [38] attack takes advantage of how IP packet fragments are reassembled. It manipulates the fragment offset field in the IP header, which indicates where a fragmented packet should be placed during reassembly.

a. Unnamed

Unnamed [39] DDoS attacks refer to Distributed Denial of Service attacks where the attacker does not disclose their identity or the means of the attack. This obscurity can result in system instability or crashes, complicating the victim's ability to identify and respond to the attack. Attackers may utilize botnets, amplification techniques, or application-layer assaults.

b. Overlapping fragmented

Overlapping fragmented [40] attacks take advantage of how network protocols process fragmented IP packets. Attackers send fragmented packets with overlapping content, targeting routers and firewalls to cause disruptions.

3. 2. 4. Ping of Death

In Ping of Death attacks [41], over-sized packets are sent to the target machine, potentially causing it to freeze or crash. Though such attacks are less common today, there has been a rise in ICMP flood attacks that share similar characteristics.

3. 3. Application attacks

Application attacks [42] aim to exploit weaknesses in applications and operating systems. Rather than merely blocking specific applications from functioning, these attacks attempt to saturate the network's bandwidth, rendering the system inoperable and leading to crashes.

3. 3. 1. SMTP floods

An SMTP [43] flood is a type of Denial of Service attack aimed at email servers. By overwhelming the server's resources, this attack can cause the server to crash or become unresponsive.

3. 3. 2. Zero Day

Zero-day [44] attacks exploit undiscovered vulnerabilities in network systems, whether in software or hardware. These attacks allow unauthorized access to system resources and can lead to data theft. Since they often go undetected until after a breach, they can be particularly challenging to defend against.

3. 3. 3. Slowloris

Slowloris [45] attacks, often referred to as slow HTTP header attacks, are a stealthy form of application-layer DDoS attack. In this scenario, a single computer establishes connections with a web server using incomplete HTTP requests, which can exhaust the server's resources. Unlike other DDoS attacks, Slowloris primarily affects the targeted web server without impacting other services or ports.

DIFFERENT TECHNIQUES FOR DDOS DETECTION

DDoS attacks involve employing multiple approaches that utilize diverse methodologies, each demonstrating varying levels of effectiveness. Below, we present some commonly used techniques for detecting DDoS attacks:

4. 1. Entropy based

Recent research shows that entropy-based techniques for DDoS detection are particularly effective for identifying anomalies. Entropy measures the level of randomness and uncertainty in incoming packets or data sets over a specific time frame, with higher randomness corresponding to greater entropy. Detection mechanisms using entropy algorithms require monitoring all packets and rely on features like SOURCE_IP, DESTINATION_IP, and PORT_NUMBER to identify abnormal network behavior and compute entropy values. Predefined thresholds aid in anomaly detection within SDN [46].

4. 2. Information theory based

The approach utilizes probabilistic theory and statistics, drawing on principles from information theory to identify

and address DDoS attacks. It focuses on analyzing network traffic data to detect anomalies and differentiate between legitimate and malicious traffic. These methods show flexibility and performance capabilities that do not rely on specific assumptions about data probabilistic distributions. However, their effectiveness can be influenced significantly by the choice of the information theory criterion, which may only become evident when anomalies are present in the data. In [47], researchers explore the use of information theory techniques for DDoS detection, applying concepts like entropy and mutual information to identify suspicious traffic patterns associated with DDoS attacks. In [48], the authors assess the efficiency of information-theory-based entropy and discrepancy measures for DDoS detection and propose a NetFlow methodology, which they validate using real benchmark datasets. Their findings indicate that divergence-based information theory metrics offer greater accuracy in identifying different attack flows compared to entropy metrics.

4. 3. Flow count based

Flow-based anomaly detection methods are recognized for their ability to identify a wide range of network anomalies, regardless of whether they are malicious or benign. In Software Defined Networking (SDN), the controller evaluates network flows by gathering statistics on PACKET_COUNT, BYTE_COUNT, IP_ADDRESS, PORT_NO., and PROTOCOL_INFO. By employing a flow counting technique, the controller continuously monitors incoming packets to analyze their behavior. Concurrently, the intrusion detection system updates the status of network components and assesses anomaly rates by comparing them to historical device statuses. An alert is generated if the anomaly rate exceeds a specified threshold. In [49], the author presents an adaptive flow counting scheme for anomaly detection in SDNs. This scheme accounts for factors such as network overhead, response time, and controller workload, improving existing methods by removing the need for flags on aggregated data. The algorithm effectively handles minor flow entries and accommodates high-volume network traffic with lower complexity. In [50], another approach is introduced, featuring an anomaly detection technique that incorporates a controller along with two main modules: a preprocessing module and an anomaly detection module. It processes flow feature vectors obtained from the SDN controller using the double P-value of transductive confidence schemes for the KNN algorithm, which significantly enhances the accuracy and efficiency of anomaly detection.

4. 4. Machine learning based

Machine learning aims to automate anomaly detection in network traffic by utilizing algorithms like supervised, unsupervised, semi-supervised, and reinforcement learning. These algorithms learn from network data, discerning normal and malicious traffic patterns. The process involves gathering and preprocessing data, training machine learning models using labeled or unlabeled datasets containing normal and attack attributes, and then classifying incoming traffic based on learned features in real-time. In [51], the authors developed a DDoS detection system using the C4.5 algorithm, combining algorithmic and signature-based detection to identify signature attacks in DDoS flooding effectively. Alternative machine-learning techniques were also considered and compared for system performance validation. In [52], the authors propose a novel mathematical model using Logistic Regression and Naive Bayes for DDoS attack detection. The model is evaluated on the CAIDA 2007 Dataset, implemented on the Weka data mining platform, and compared with alternative approaches using different machine-learning algorithms. In [53], the authors contribute novel features for DDoS detection, creating an openly accessible dataset stored in a CSV file. Using a hybrid machine learning model (SVC-RF), they achieve an impressive 98.8% testing accuracy with minimal false alarms. In [54], researchers introduced a novel method for securing network communication using machine learning to detect DDoS attacks in an SDN-WISE IoT controller system. The tested algorithms, including Naive Bayes (97.4%), SVM (96.1%), and DT (98.1%), demonstrated high accuracy. This innovation enhances security in SDN-WISE setups, preventing unauthorized access, as empirical findings from experimental research validate.

4. 5. Deep learning based

Deep learning enhances DDoS detection in SDN by leveraging algorithms to autonomously learn features, improving accuracy and efficiency in detecting and mitigating attacks. In [55], the authors introduced a specialized DDoS detection system tailored for SDNs, employing deep learning for streamlined feature selection and traffic organization. Likewise, in [56], they suggested a deep-learning approach for anomaly detection in flow-based data

Journal of Information Systems Engineering and Management
2025, 10(51s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

within SDNs, developing a deep neural network (DNN) model for an intrusion detection system (IDS). The model was trained on the NSL-KDD dataset by opting for six features from the primary collection of forty-one features. In [57], the authors propose a framework integrating deep learning and cookie analysis for real-time detection and mitigation of web attacks. The Convolutional Neural Network (CNN) is trained on HTTP request parameters and a cookie analytical module to identify abnormal cookies and potential privacy breaches. The framework efficiently reduces processing time for real-time web application protection deployment. In [58], the authors propose an LSTM-based model for DDoS threat detection, achieving up to 98% accuracy on the 'CICDDoS2019' dataset. The study highlights the superior performance of deep learning, particularly LSTM, in accurately identifying DDoS attacks, emphasizing its efficacy in network security.

4. 6. Hybrid based

Previous studies have shown that hybrid-detection approaches can be strengthened by integrating anomaly-based, feature-based, signature-based, and volume-based techniques. The intelligent trust model (ITM) presented by the author in [59] follows extreme learning machines (ELMs) to detect diverse DDoS attacks in the SDN environment. ITM enables rapid updates and real-time response to various DDoS attacks. It helps achieve an accurate detection rate with a low false positive rate. In [60], a novel hybrid DDoS detection technique combines feature volume-driven approaches. Upon evaluation, it was found that the proposed technique performs better than the existing techniques. In [61], the author introduces a novel network intrusion detection system that employs deep learning techniques to improve its ability to detect potential security risks. Through extensive training on datasets such as CICIDS2018 and Edge_IoT, the proposed technique achieves 99.99% accuracy in multiclass classification and 99.64% in dataset testing. In [62], the author presents the novel approach GA-DT, a new technique for detecting DDoS attacks. It combines a genetic algorithm and a decision tree algorithm to improve detection accuracy.

4. 7. Integration of Snort & OpenFlow

In some studies, researchers integrated an Intrusion Detection System (IDS) tool like SNORT with the OpenFlow protocol to dynamically monitor and manage malicious activities across network resources, enabling real-time detection of anomalies or attacks. In [63], the author developed an early DDoS detection tool by integrating SNORT IDS with popular SDN controllers. The tool underwent testing in five network scenarios using Mininet emulation and four penetration tools, demonstrating timely detection of fast DDoS attacks with improved SDN controller performance. In [64], the authors utilized a signature-based Snort IDS to detect and monitor suspicious activity by mirroring traffic in OpenvSwitch (OVS).

DJRT MODEL

The proposed model introduces an advanced solution for the real-time monitoring, detection, and mitigation of UDP DDoS attacks in an SDN environment. Figure 2 illustrates the proposed framework. This methodology relies on flow statistics analysis and encompasses three primary components: detection and mitigation, sFlow, and elephant flow. By utilizing the sFlow tool, which employs sampling techniques to collect flow statistics, this framework effectively identifies and mitigates high-rate DDoS attacks in the SDN environment. It integrates the capabilities of both OpenFlow and sFlow with a centralized RYU controller to orchestrate network management. OpenFlow-enabled switches, such as openvswitch, function as agents that transmit sampled data to the flow statistics collector known as sFlow-RT [65]. The primary role of sFlow-RT is to perform a comprehensive analysis of the incoming samples and accurately detect Elephant Flows (EFs) that signal the presence of DDoS attacks based on the analyzed data.

Journal of Information Systems Engineering and Management
2025, 10(51s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

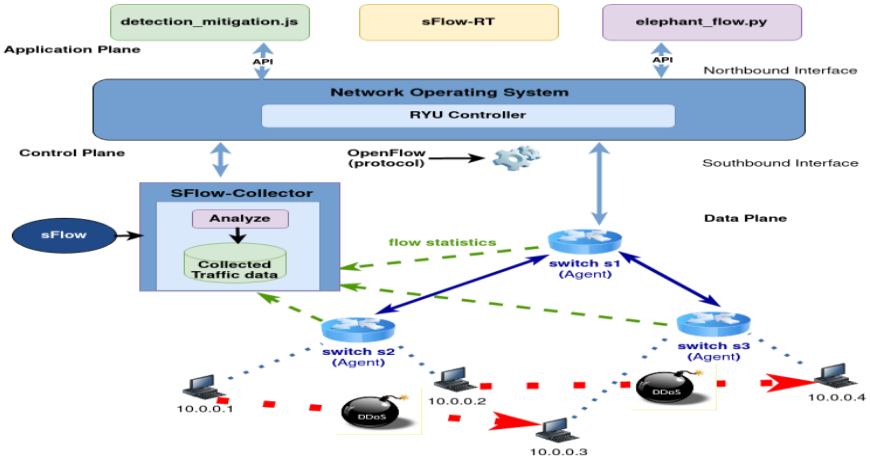

Figure 2. Proposed model architecture (DJRT)

The sFlow-RT application receives a continuous stream of sFlow measurements from the switch agents and swiftly detects EFs in real-time. This systematic approach enhances the resilience of SDN architectures against high-rate DDoS attacks. Figure 2 illustrates the anomaly detection and alleviation process, which operates in three stages. The components of the DJRT model are as follows:

5. 1. sFlow

The sFlow technology represents a pioneering and open-source network monitoring and sampling approach. As stated in [66], sFlow has been identified as a suitable technology for monitoring real-time collection and analysis of traffic data at high speeds. Capturing samples of network packets offers valuable insights into network performance, traffic patterns, and potential security threats. Various devices can be used to collect sFlow samples, including physical switches, virtual switches, hosts, routers, etc. With reduced overhead, it is possible to configure sFlow monitoring on all devices' interfaces. Based on monitoring rules, sampling rates can be established for individual links. The sFlow agent within network devices performs random sampling by adhering to predetermined polling and sampling rates. The sFlow tool includes the sFlow agent, the sFlow-RT collector, and the sFlow analyzer.

- *sFlow agent:* It can be integrated into various network devices such as switches, routers, and hubs. It analyzes network traffic and collects statistics from every incoming and outgoing packet.
- *sFlow-RT collector:* In SDN, the sFlow-RT collector is crucial for gathering and interpreting statistical data from network devices (sFlow agent). The collected sFlow samples are evaluated to offer a detailed insight into various aspects such as performance, security concerns, and traffic flow.
- *sFlow analyzer:* It provides comprehensive insights into network traffic in real-time and historical context. The comprehensive analysis enables the identification of anomalies due to DDoS attacks, malware infection, and other potential security issues.

5. 1. 1. Flow sampling

The total packet and sample count are initialized to zero at the start. A variable 'skip' stores the value (rate) of 'next_skip' and waits for every incoming packet. On each packet arrival, it is either accepted or rejected. If rejected, the system continues waiting for the next packet. If the packet is received, it is forwarded to the designated port or interface. The 'skip' count decreases each step while the total packet count increases. When the 'skip' value (skip != 0) is not zero, the packet is forwarded to the destination port, and if the 'skip' value (skip = 0) is zero, then the 'skip' value is updated with the current rate of 'next_skip' and the sample count is also updated. Finally, all information is sent to the switch agents.

5. 1. 2. Flow identification

Journal of Information Systems Engineering and Management
2025, 10(51s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

sFlow makes it easier for network operators to capture and analyze essential traffic details, which can be instantly processed and analyzed in real-time. Previous studies have proposed and evaluated sFlow-based algorithms and methodologies and shown how they can help detect anomalies, patterns, and other security threats. Moreover, integrating sFlow with machine learning techniques may improve the detection accuracy.

5. 2. Modules for DDoS detection and mitigation

Figure 3 shows the general structure of anomaly detection and alleviation. The first stage involves the data collection techniques module, which gathers traffic data from OpenFlow[67] switches functioning as sFlow agents. Specific algorithms run in the background to identify anomalies in the detection stage. Furthermore, stage three focuses on the measures to mitigate destructive network traffic after detecting an attack.

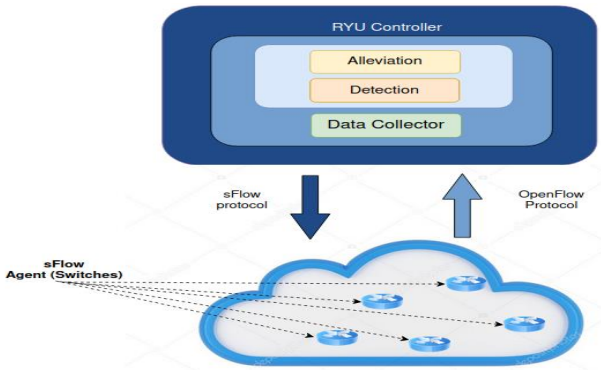

Figure 3. Modules for DDoS detection and mitigation

5. 2. 1 Collector module

The collector module serves a vital function in the acquisition of traffic data, which is essential for identifying anomalies in network flow. It systematically gathers flow-related information and relays this data to the anomaly detection module for comprehensive analysis. Within the context of the SDN framework, this module plays a critical role in collecting and processing network flow data. It retrieves traffic information from various network devices while managing the flow of packets throughout the network. By capturing and scrutinizing network traffic, the collector module generates valuable insights that enhance effective network management. Overall, it is indispensable for monitoring and analyzing network flows within an SDN environment.

5. 2. 2 Detection algorithm

The detection algorithm shown in *Algorithm 1* plays a crucial role in identifying attack when the threshold exceeded the predefined limit. It utilizes the flow metric (udp_reflection) to effectively monitor and analyze traffic patterns. The detection module detect malicious activities of attack patterns based on threshold value, which is used to detect specific type of attack, in this case, UDP reflection attack. The threshold is based on a metric UDP reflection that tracks the number of frames in a UDP reflection flow. If the metric (no. of frames) exceeds a set threshold value, an attack is considered to be occurring and the system triggers to call an event handler (mitigation algorithm) that reacts when the threshold value is exceeded, which indicates the presence of an attack.

A crucial aspect of detection algorithm is the prediction value, which acts as input and guides its actions. A prediction value 1 indicates that DDoS traffic is present, while a value 0 represents normal traffic. When the detection module produces a prediction value of 1, it triggers a REST API message that is sent to the controller to block the DDoS traffic. In contrast, if the prediction value is 0, no message is sent to the controller. In case where multiple attacks are identified, several REST messages will be dispatched to the controller to mitigate DDoS traffic.

Set Flow sets the flow for tracking UDP reflection attack based on the source and destination IP addresses and the UDP source port, while the value:'frames' represents the metric being tracked (number of frames in the flow).

SetFlow('udp_reflection', { keys: 'ipsource, ipdestination, udpsourceport', value: 'frames'});

Journal of Information Systems Engineering and Management
2025, 10(51s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Set Threshold is set to trigger an event if the metric exceeds more than 100 frames with in a certain time period (timeout set to 2) as:

setThreshold ('udp_reflection_attack', {metric : 'udp_reflection', value:100, byFlow: True, timeout: 2});

The **even handler** listen for when the threshold is triggered (when the udp_reflection_attack event occurs), indicating a potential DDoS attack . It then sends a flow rule to the RYU controller to block the attack. The **SetIntervalHandler** periodically checks whether the threshold is still triggered and delete the flow rules once the attack ceases.

Algorithm 1

Detection Algorithm

1. Initialize empty dictionary 'controls'
2. Set threshold value for UDP reflection attacker
3. For each incoming event 'evt':
a. Check if the event involves inter-switch link; if so, skip this event
b. Retrieve port info rom event data
i. If no port info available, skip event
ii. If no OpenFlow info available (dpid, ofport), skip event
c. If a flow for this event already exists in 'controls', skip event
d. Extract Source IP, Destination IP, and UDP source port from 'evt.flowkey'
e. Contruct flow rules to block attack with these parameters:
- Source IP
- Destination IP
- UDP Source Port
- OpenFlow port (ofport)
f. Send flow rules to RYU controller to block the attack via an HTTP request
g. Store control information in 'controls' dictionary with:
- Flow key
- Time of detection
- Threshold ID
- Event data
4. Log the detection of attack for monitoring
End algorithm

5. 2. 3 Mitigation algorithm

The mitigation algorithm as shown in *Algorithm 2*, handles the response to a detected attack. It involves sending the flow rules to the RYU controller to block malicious traffic and the event handler checks periodically whether the threshold is still triggered (i.e. the attack is ongoing). If the attack is no longer triggered, then remove the flow rule from RYU controller, indicating the attack has ended.

Journal of Information Systems Engineering and Management
2025, 10(51s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Algorithm 2

Alleviation/Mitigation Algorithm

1. Initialize current time as 'now'

2. For each entry 'key' in controls

a. Retrieve control record 'rec' for the given flow key 'key'

b. If the control record was created less than 10 seconds ago,

continue

checking the next record

c. If the threshold for this record is still triggered (attack is

ongoing):

- Continue to next record

d. If the threshold is no longer triggered (attack is no longer

ongoing):

i. Send HTTP request to RYU controller to remove the flow

entry using

DELETE operation at '/stats/flowentry/delete'

ii. Delete the record from 'controls'

iii. Lof that the attack has been unblocked for the given flow

key

3. Continue checking other flow records

End algorithm

3. Elephant Flow detection algorithm

The Elephant Flow Detection System, developed using a Python script called Elephant.py, successfully identified key traffic flows associated with several DDoS attacks. This detection was further confirmed through visual analysis using the sFlow-RT Mininet dashboard application, which provides a real-time visualization of the affected routes and switches, as well as elephant flows (EFs) in topology as shown in Figures 29 and 30. Timely detection and management of elephant flows are critical for effectively responding to large-scale attacks and minimizing their impact on the overall network.

Journal of Information Systems Engineering and Management
2025, 10(51s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Algorithm 3

Elephant Flow Detection Algorithm

1. Initialize REST API base URL:
rt ← "http://127.0.0.1
2. Define flow configuration:
flow.keys ← "link:inputifindex, ipsource, ipdestination"
flow.value ← "bytes"
3. Send HTTP PUT request to rt + "/flow/pair/json" with JSON(flow)
4. Define threshold configuration:
threshold.metric ← "pair"
threshold.value ← "1000000 / 8" //byte threshold
threshold.byFlow ← TRUE
threshold.timeout ← 1 //seconds
5. Send HTTP PUT request to rt + "/threshold/elephant/json" with JSON (threshold)
6. Construct event polling URL:
eventurl ← rt + "events/json?

thresholdID=elephant*maxEvents=10&timeout=60"
7. Initialize eventID ← -1
8. Loop indefinitely:
a. Make HTTP GET request to eventurl + "&eventID" = + eventID
Let response result of GET request
b. If response.status_code != 200 then
Exit loop
c. Parse JSON response into events list
d. If events liste is empty then
Continue to next iteration (poll again)
e. Update eventID ← events[0].eventID //most recent eventID
f. Reverse events list to process oldest to newest
g. For each event e in events do:
Output e.flowkey
9. End loop

This algorithm interacts with a network monitoring REST API to detect and output significant network flow events often called as elephant flows due to their large volume of data. It initialize REST API base url as rt ← "http://127.0.0.1:8008" this sets root endpoint of the REST API server that manages network flow data. The address 127.0.0.1 is a local address, 8008 is the network port where the service listens for API requests. The flow.keys and flow.value defines how the network flow will be identified and measured as shown ing *Algorithm 3*. The program sends an HTTP PUT request to the API endpoint /flow/pair/json submitting the "flow" configuration as a JSON object. This registers or updates the monitoring system to track flows matching the specified keys and

Journal of Information Systems Engineering and Management
2025, 10(51s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

value metric. The following fields shown in Table 2 set the criteria to identify elephant flows exceeding a threshold in bytes.

Table 2: Threshold Configuration

Sr. No.	Field	Explanation
1.	metric: "pair"	This indicates the threshold value applies to source to destination flow pair
2.	Value: "1000000 / 8"	It sets the thresold at 125000 bytes (1000000 bits converted to bytes)
3.	ByFlow: TRUE	It restrict threshold application to individual flows rather than aggregated data
4.	Timeout: 1	Timeout is in second configures how long a flow exceeding the threshold is considered active or relevant

An HTTP PUT request is sent to /threshold/elephant/json with the threshold configuration that registers the threshold under the name "elephant" in the monitoring system. The eventurl is used to request up to 10 events from the API where flows exceed the "elephant" threshold. The timeout=60 configures the maximum waiting time for the server to respond with events - eventID ← -1 respresents no events have yet been processed. The program then regularly enters a continuous loop to fetch new flow events exceeding the threshold. To make an HTTP GET request, it sends a request to "eventurl" with the current eventID as a parameter. The server then returns events with IDs greater than eventID (i.e., new events). If the HTTP status request is not 200, the loop terminates, signaling an error or shutdown. The server response is parsed as a JSON list of events. If the event list is empty, the loop continues to poll again, waiting for new events. The program updates "eventID" to the latest event's ID (events[0].eventID), which ensures the subsequent request only fetches events that occurred after this. Events are reversed so they are processed from oldest to newest, preserving temporal order. For each event, it prints the flow identifier (flowkey) representing the source and destination interface. The process continues until API returns an error or the program is manually stopped. This elephant flow detection algorithm serves as an elephant flow anomaly detector.

METHODOLOGY

The research aims to develop a real-time detection and mitigation model for high-rate DDoS attacks (DJRT), a novel model designed to detect and mitigate multiple high-rate DDoS attacks accurately. By effectively integrating the sFlow statistical monitoring tool with data plane devices, the framework facilitates the comprehensive monitoring of network activities, including traffic flow statistics, thereby enabling the identification of malicious network behaviors. This approach offers network security professionals a valuable capability to visualize both regular and high-volume traffic flows across diverse network topologies and their interconnections, thereby enhancing their understanding of network dynamics and overall behavior. The study is conducted through an emulation-based experiment within a virtual lab environment built using Mininet, employing tools such as the Mininet emulator, openvswitch, the sFlow-RT tool, x-term terminal, hping3 [68], the RYU[69] controller, as well as custom-developed scripts Elephant.py and ryu2m.js. Network traffic is monitored via traffic flow, which is statistically collected by an sFlow collector integrated into openvswitch, serving as an sFlow agent. The detection framework operates across the three layers of SDN—the data plane, control plane, and application plane—utilizing the sFlow tool within openvswitch in the data plane and communicating with the RYU controller on the control plane through the OpenFlow protocol. The custom scripts run in the application plane, aiming to achieve real-time detection and mitigation of high-rate DDoS attacks. These are executed using the hping3 tool among virtual hosts and are visually displayed through the Mininet dashboard application of the sFlow-RT tool.

RESULTS AND DISCUSSION

This section presents the findings from the experimental evaluation of a defense mechanism for the high-rate DDoS

Journal of Information Systems Engineering and Management
2025, 10(51s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

attacks within a virtual SDN environment. This study investigates the network response to both single and multiple simultaneous high-rate DDoS attacks using a flow-based statistics approach for detection and mitigation. The experimental setup is based on a Tree topology consisting of 27 hosts, 13 switches, and a RYU controller, emulated via mininet.

7. 1 Scenario 1: Normal Network Traffic

In the first scenario SDN was subjected to normal network traffic without any malicious interference. The traffic involved 27 hosts pinging each other, providing a baseline for comparison. The network metric obtained during normal operations indicated a stable state with an sFlow data rate of 61.5 kbps and an sFlow packet rate of 12.7 pps. CPU utilization was found to be low at 1.78%, and memory utilization was moderate at 50.04%, as shown in Figure 4. The results align with typical network conditions, highlighting the ability of the SDN to maintain optimal performance under standard usage. Real-time monitoring through the mininet dashboard and sFlow-RT GUI (Figures 5, 6, and 7) allowed visualization of the network flow data, top switch ports, and overall topology. These visual aids provide an in-depth understanding of how the network topology and flow distributions performed under normal conditions, contributing to the effectiveness of the SDN architecture in monitoring and managing traffic.

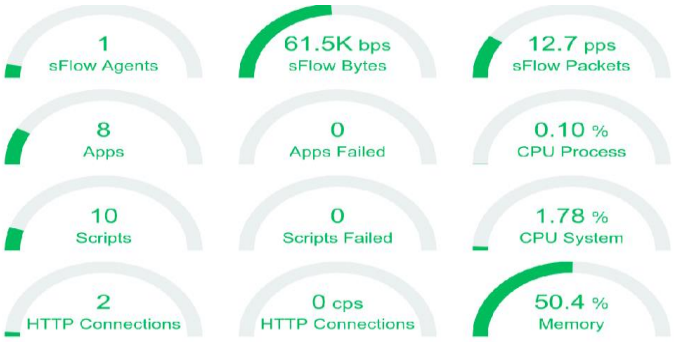

Figure 4: sFlow-RT status normal traffic

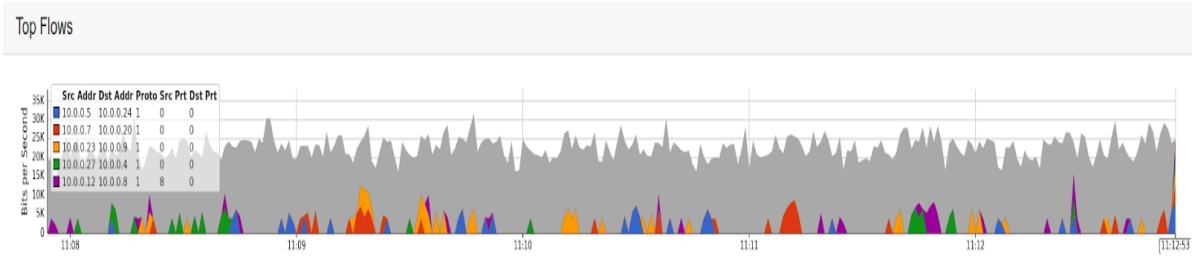

Figure 5 : Top flows for normal traffic due to 27 host ping among each other

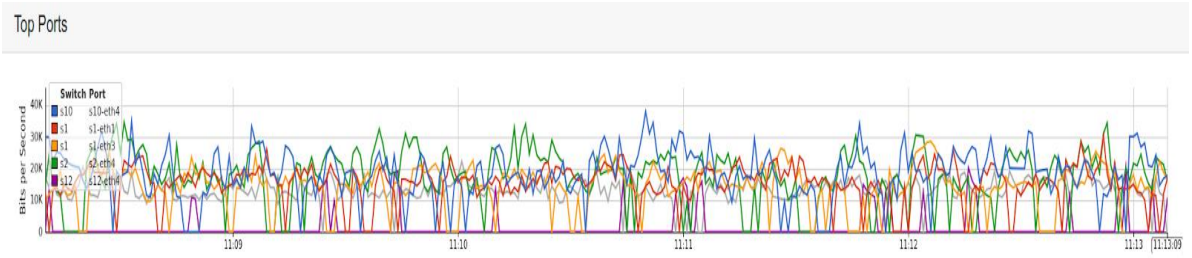

Figure 6: Top switch ports for normal traffic due to 27 host pinging among each other

Journal of Information Systems Engineering and Management
2025, 10(51s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

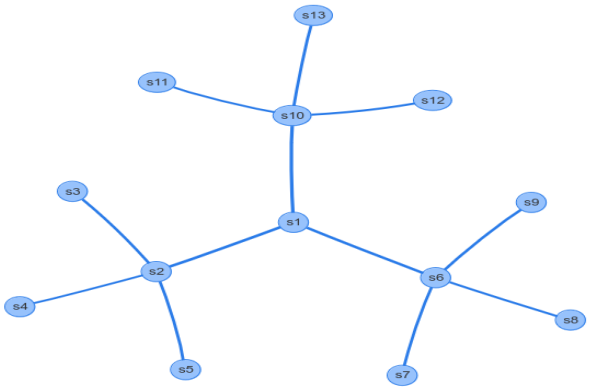

Figure 7: Tree topology

7. 2 Scenario 2: Single DDoS attack

In the second scenario, a single UDP DDoS attack was simulated using the hping3 tool, targeting host h10 from host h24 (Figure 8). As shown in Figure 9, the module successfully detected and blocked the attack by adding a flow entry that filtered traffic from the attacking host. This dynamic response underscores the effectiveness of the hybrid detection and mitigation system. This attack significantly affected the network's behavior, as evidenced by the results shown in Figure 10. The sFlow data rate increased drastically to 25.0 mbps, and the sFlow packet rate rose to 2.32 kpps, indicating the presence of high traffic volume due to the attack. CPU utilization surged to 30.80%, and memory utilization reached 70.7%, highlighting the strain on the resources induced by the attack. The impact of the DDoS attack was also reflected in the Top Flows and Top Ports charts, as depicted in Figures 11 and 12. The primary traffic flow, previously dominated by standard host communication, was now heavily influenced by the malicious UDP traffic generated by the attack. The top port analysis revealed that the most significant traffic was observed at port three on switch 12 (s12-eth3), correlating with the UDP DDoS attack from h24 to h10, which indicates that the specific ports were targeted and congested, affecting the system's vulnerability. Furthermore, the sFlow test results (Figures 13, 14, and 15) demonstrated the marked increase in flow rates with 80 mbps, 250K packets per second (pps), and 25K samples per second (sps), underscoring the substantial load imposed by the single attack. The real-time detection and mitigation process via the ryu2m.js module played a crucial role in managing the attack.

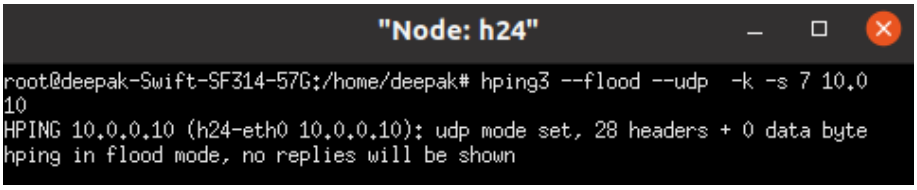

Figure 8: DDoS attack from h24 to h10

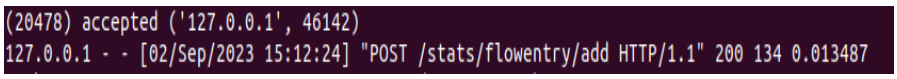

Figure 9: Flow entry made by controller for DDoS attack detection

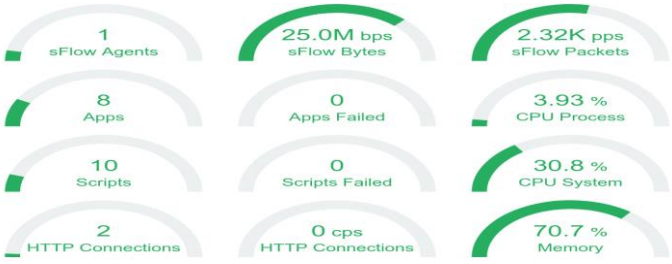

Figure 10: sFlow-RT status (Single UDP DDoS attack)

Journal of Information Systems Engineering and Management
2025, 10(51s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

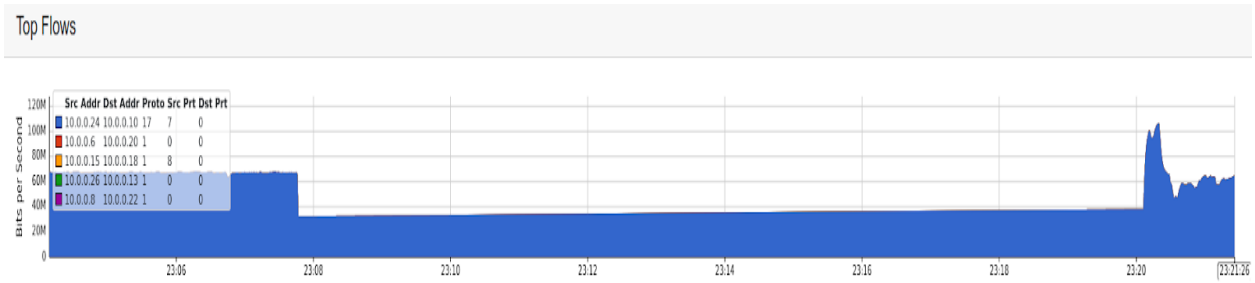

Figure 11: Top flows (anomalous traffic due to single UDP DDoS attack in M bps)

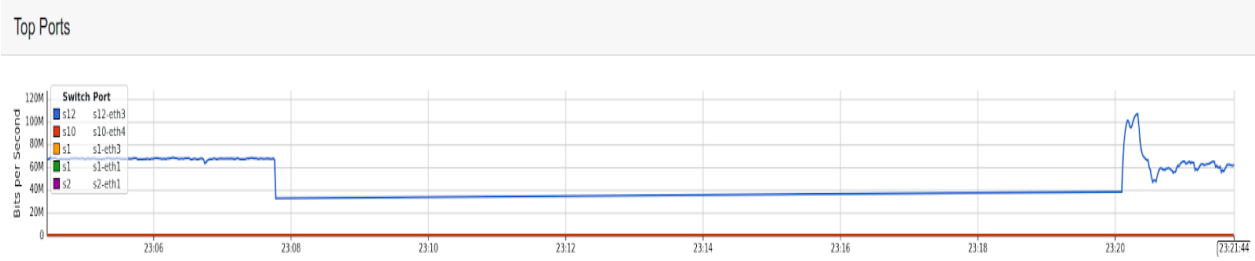

Figure 12: Top switch flows (anomalous traffic due to UDP DDoS attack in M bps)

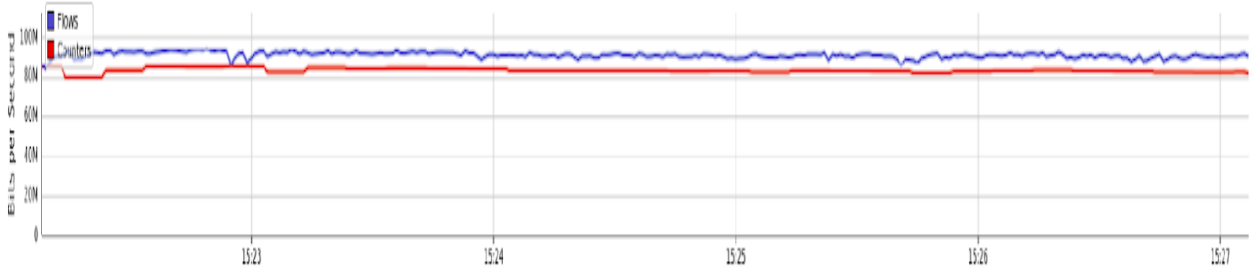

Figure 13: Single UDP DDoS attack impact on sFlow-test as Bits per Second

Figure 14: Single UDP DDoS attack impact on sFlow-test as Packets per Second

Figure 15: Single UDP DDoS attack impact on sFlow-test as Samples per second

7. 3. Scenario 3: Multiple Simultaneous DDoS attacks

In the third scenario, three simultaneous UDP DDoS attacks were launched from hosts h3, h14, and h21, targeting hosts h27, h18, and h2, respectively (Figures 16, 17, and 18). As soon as these attacks are made, they are successfully detected and blocked, as illustrated in Figure 19. The controller then updated the flow entries accordingly, as shown in Figures 20, 21, and 22, respectively. This scenario presented a significant challenge due to the concurrent nature

Journal of Information Systems Engineering and Management
2025, 10(51s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

of the attacks. Metrics from sFlow-RT showed a marked increase in network traffic, with the sFlow data rate peaking at 36.9 Mbps and the packet rate reaching 3.43 Kpps (Figure 23). Consequently, CPU utilization surged to 96.90%, while memory utilization increased to 87.2%, highlighting the system's susceptibility to multiple simultaneous DDoS attacks. These findings suggest that the network resources become quickly overwhelmed when faced with attacks from various sources. The Top Flows chart (Figure 24) illustrated a substantial volume of traffic brought on by the DDoS attacks, with the peak flow resulting from the collective impact of the three UDP floods and regular ping traffic among 30 hosts.

Additionally, the Top Ports chart (Figure 25) pointed out the specific ports that were most affected by the attack traffic, showcasing the areas of the network that bore the brunt of the malicious activities. The sFlow-test results (Figures 26, 27, and 28) further revealed the extent of the impact from the multiple attacks. Flow rates were recorded at approximately 175-180 Mbps, with 500 K packets per second and 49 K samples per second, indicating significant performance degradation in the network due to the coordinated DDoS effort.

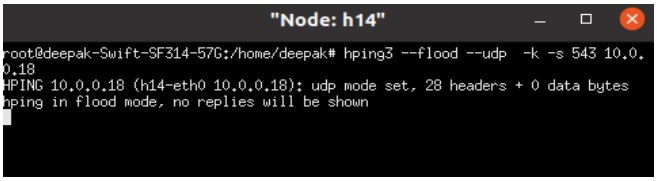

Figure: 16 UDP flood attack from h14 to h18

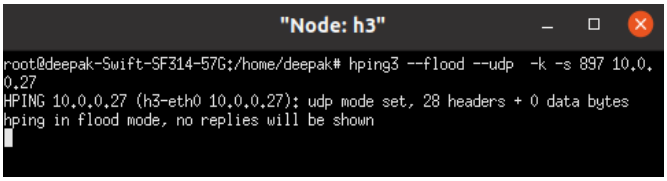

Figure 17: UDP flood attack from h3 to h27

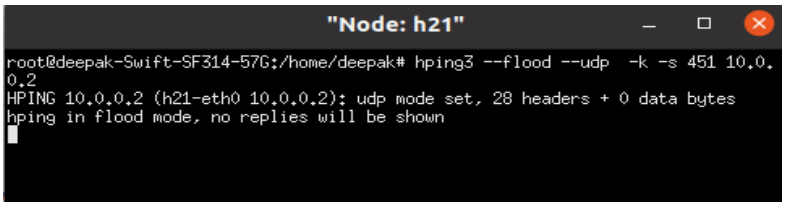

Figure 18: UDP flood attack from h21 to h2

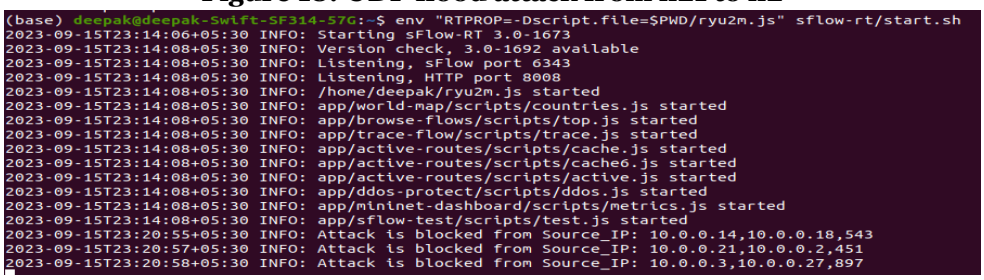

Figure 19: Multiple DDoS attack detection from h3, h14, and h21

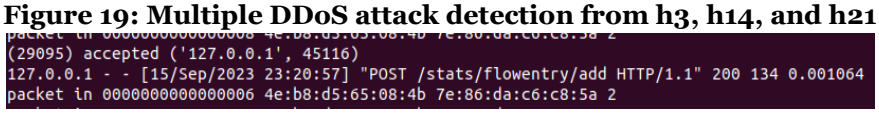

Figure 20: First flow entry added by controller for attack from h14 to h1

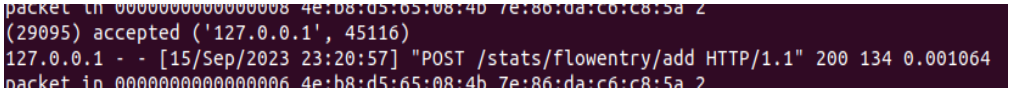

Figure 21: Second flow entry added by controller for attack from h21 to h2

Journal of Information Systems Engineering and Management
2025, 10(51s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

```
(29095) accepted ('127.0.0.1', 45128)
127.0.0.1 - - [15/Sep/2023 23:20:58] "POST /stats/flowentry/add HTTP/1.1" 200 134 0.000941
packet in 0000000000000003 f6:c4:13:bb:1c:4b ae:92:30:37:53:9f 3
```

Figure 22: Third flow entry added by controller for attack from h3 to h27

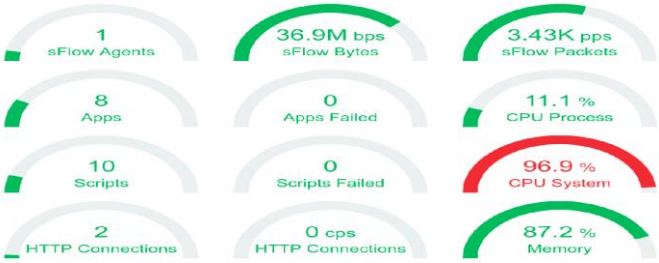

Figure 23 : Status of sFlow-RT (For multiple DDoS attacks)

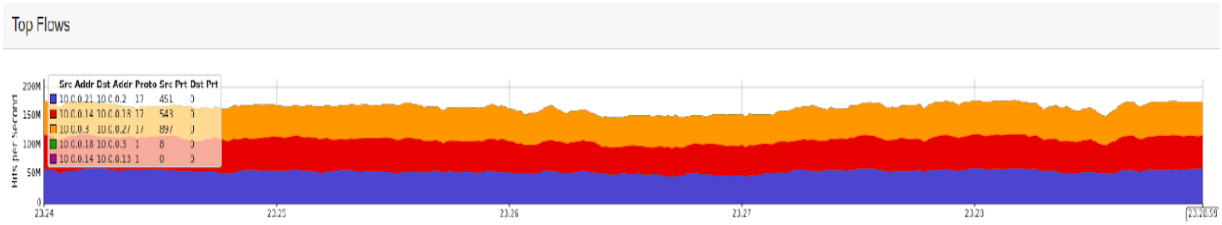

Figure 24: Top flows (anomalous traffic due to 30 host ping and 3 UDP DDoS attack (High-rate) in Mbps)

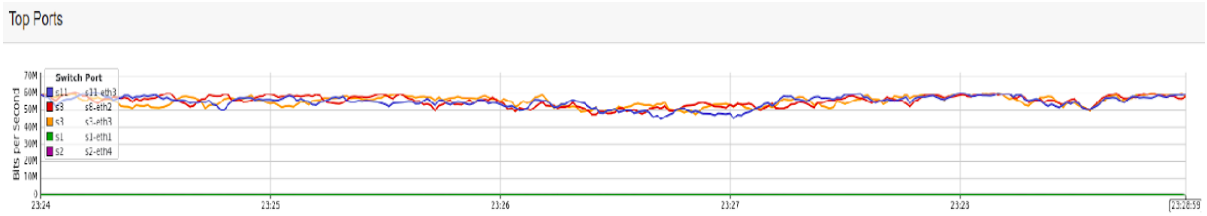

Figure 25: Top port (anomalous traffic due to 27 host ping and 3 UDP DDoS attack in M bps)

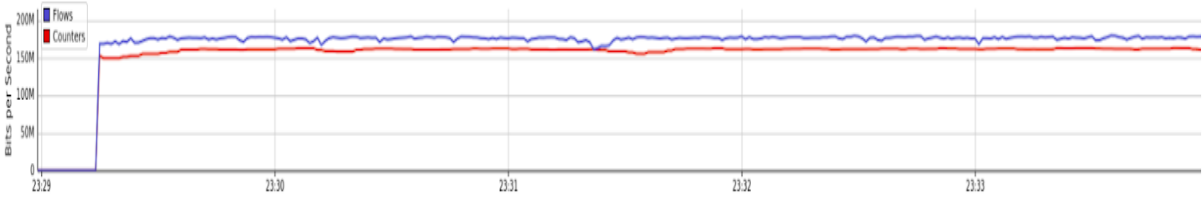

Figure 26: Multiple UDP DDoS attack impact on sFlow-test as Bits per Second

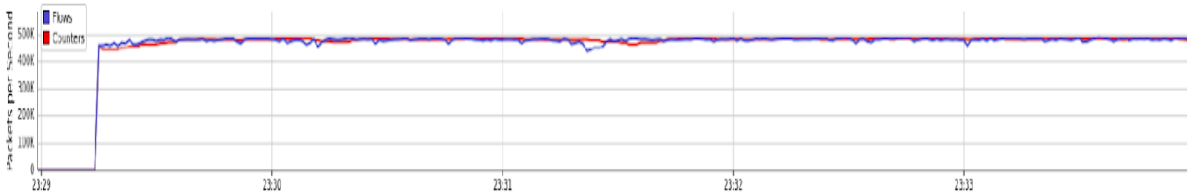

Figure 27: Multiple UDP DDoS attack impact on sFlow-test as Packets per Second

Journal of Information Systems Engineering and Management
2025, 10(51s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Figure 28: Multiple UDP DDoS attack impact on sFlow-test as Samples per Second

COMPARATIVE ANALYSIS OF DJRT MODEL WITH PREVIOUS STUDIES

In Table 3 comparison highlights how DJRT model is better than previous techniques for real-time DDoS detection, especially when we have deal with high-rate DDoS attacks. We have performed comparative evaluation based on following points:

8. 1 Detection and mitigation of multiple simultaneous DDoS attacks

8. 1. 1. DJRT: It is distinguished by its capability to detect and mitigate multiple simultaneous high-rate DDoS attacks in real time. This feature is especially critical in contemporary distributed denial-of-service (DDoS) scenarios, where attackers often employ various sources and attack vectors simultaneously to overwhelm their targets. It continuously analyzes traffic using sFlow sampling, ensuring that it can identify low-rate attacks in real-time, even those attempting to evade detection by blending in with normal traffic. On detecting the attack, DJRT implement mitigation or blocking the attackers IP address immediately, preventing the damage before the attack can escalate.

8. 1. 2 Previous studies: Most studies in the table focus mainly on detection methods and do not include an integrated strategy for mitigating attacks. Additionally, those that do address mitigation often concentrate on individual types of attacks and lack real-time response capabilities. An example of this would be the TRW-CB methods, as referenced in sources [70] and [69]. While these methods are effective at identifying attacks, their primary objective is detection rather than providing real-time automated responses to mitigate those threats. Also, SVM-based approaches, as seen in studies [67] and [68], can be effective for detection purposes. However, they frequently lack direct integration with real-time mitigation strategies and may struggle to address multiple simultaneous attacks. Often, these methods necessitate further post-processing or manual intervention to implement effective mitigation measures.

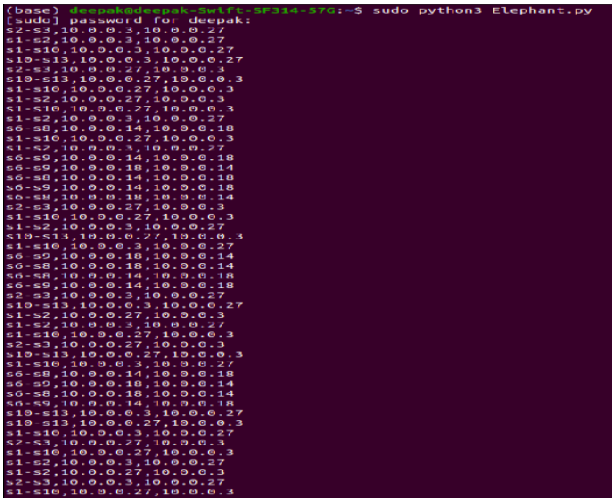

Figure 29: Elephant flow detection across routes and switches

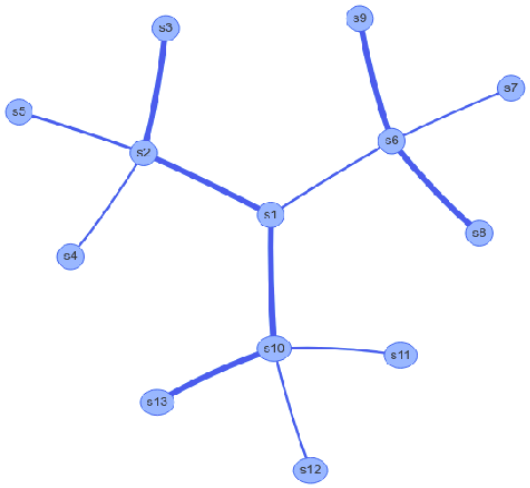

Figure 30: Visualization of elephant flows in used topology

Journal of Information Systems Engineering and Management
2025, 10(51s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Table 3: Comparative evaluation with previous studies

Reference	Methods	Flow Statistics Collector	ML / DL	Types of attack detected	Attacks Detected		Mitigation		Elephant Flow Detection	High-rate DDoS	N/w size
					Single	Multiple	Single	Multiple			
[49], 2016	sFlow + OpenFlow	Flow statistics & features	None	DDoS / Probe	Yes	No	No	No	No	No	200 mbps
[70], 2016	Openflow	TRW-CB	None	DDoS / Worm propagation	Yes	No	No	No	No	No	200 mbps
[71], 2017	Openflow	TRW-CB	DL	DDoS	Yes	No	No	No	No	No	200 mbps
[72], 2016	Netflow + Openflow	TRW-CB	DL	DDoS	Yes	No	No	No	No	No	200 mbps
[73], 2016	Openflow	Flow statistics & features	None	DDoS / Probe	Yes	No	No	No	No	No	100 mbps
[74], 2016	Openflow	TRW-CB	None	DDoS / Worm propagation	Yes	No	No	No	No	No	100 mbps
[75], 2017	Openflow	TRW-CB	None	DDoS / DoS	Yes	No	No	No	No	No	200 mbps
[76], 2017	Openflow	TRW-CB	SVM	DDoS / Link flooding	Yes	No	No	No	No	No	200 mbps
[77], 2017	Openflow	TRW-CB	None	DDoS / DoS	Yes	No	No	No	No	No	200 mbps
[78], 2018	Openflow	TRW-CB	SVM	DoS / DDoS	Yes	No	No	No	No	No	100 mbps

Journal of Information Systems Engineering and Management
2025, 10(51s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

[79], 2018	Openflow	TRW-CB	SVM	DoS / DDoS	Yes	No	No	No	No	No	100 mbps
[80], 2018	Openflow	TRW-CB	DL	DDoS	Yes	No	No	No	No	No	200 mbps
[81], 2019	Netflow + Openflow	TRW-CB	None	DoS / Probe	Yes	No	No	No	No	No	100 mbps
Proposed DJRT	**sFlow + RYU Openflow**	**Flow statistics & features**	**None**	**UDP DDoS**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**100 mbps**

8. 2 Reat-Time Detection and Mitigation

8. 2. 1 DJRT: The DJRT model operates in real-time, allowing for immediate responses without waiting for attack data accumulation. Through an sFlow-based approach, it continuously samples network traffic to promptly detect anomalies. Once a high-rate DDoS attack is detected, DJRT quickly activates measures to protect the network, like filtering or slowing down the incoming traffic.

8. 2. 2 Previous studies: While various studies in the Table 2 utilize artificial intelligence techniques such as machine learning and deep approaches to identify attacks, many of them do not prioritize real-time mitigation solutions. Machine learning and deep learning approaches, as highlighted in references [73] and [75], primarily focus on improving classification and detection accuracy. However, many of these methods do not emphasize the importance of real-time response capabilities. For instance, detection models like Support Vector Machines, as cited in [78] and [79], often fail to operate in real-time or provide the immediate response necessary to mitigate the effects of high-rate DDoS attacks, particularly in scenarios lacking an integrated mitigation mechanism.

8. 3 High-rate DDoS attack detection

8. 3. 1 DJRT: DJRT is designed specifically to address high-rate DDoS attacks, which can be less challenging to identify than low-rate attacks. Attacks like UDP may produce significant high spikes in traffic volume, rendering them similar to normal traffic and difficult to detect. One of DJRT's strengths is its use of sFlow sampling, which facilitates detailed, real-time analysis of flow-based data. This ability to analyze network traffic in real time is essential for spotting high-rate DDOS that standard detection systems might overlook, highlighting the critical nature of this issue.

8. 3. 2 Previous studies: Previous research has mainly concentrated on conventional DDoS attacks or probing attacks, as seen in works referenced by [49], [70], and [75]. However, these studies often do not specifically cater to high-rate attacks. For example, methodologies that rely on flow statistics (referenced in [72] and [74]) risk overlooking low-rate attacks due to their limited data traffic, which differs from the high-volume traffic usually associated with traditional DDoS incidents. This gap underscores the necessity for a more effective solution like DJRT. The machine learning methods discussed in studies [79] and [80] have the potential to identify low-rate attacks. However, they are limited by the absence of a real-time detection and mitigation system, which affects how effectively they can respond to issues quickly.

8. 4. Unified approach to detection and mitigation:

Journal of Information Systems Engineering and Management
2025, 10(51s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

8. 4. 1. DJRT: It stands out from conventional systems by combining detection and mitigation into one unified approach, greatly enhancing its capability against real-time DDoS attacks. Upon detecting an attack DJRT not only alerts the user but also implements proactive strategies to neutralize the threat, such as regulating traffic flow and applying rate limits. Such a comprehensive approach is essential for minimizing the impact of high-rate DDoS attacks.

8. 4. 2 Previous studies: Many of the studies mentioned in the Table such as those referenced [49], [70], [75], primarily concentrate on either detecting or mitigating of issues independently. Although, some suggest mitigation methods, like the one in [72] using SVM, they fall short of offering a unified approach that seamlessly integrates real-time attack detection with immediate mitigation efforts.

FUTURE SCOPE

The DJRT model greatly enhances security in SDN by effectively addressing high-rate DDoS attacks, which pose a significant challenge to network security. This study introduces a novel statistical framework using the sFlow tool to detect and mitigate high-rate DDoS attacks in real-time, emphasizing its vital role in protecting networks. One of the distinguishing advantages of the DJRT model is its capability to handle multiple attacks in real-time. This feature allows for a more flexible and proactive defense compared to traditional methods. The framework continuously monitors and analyzes network traffic, providing ongoing protection against new threats.

Additionally, the DJRT model enables network administrators to visualize traffic flows, making it easier to identify significant patterns like elephant flows (EFs). These long-lasting large flows aid in better management and early detection of potential problems. The framework's effectiveness has been demonstrated through tests using a mininet emulator, showcasing its performance in virtual SDN environment. At the same time DJRT provides a scalable and effective defense against high-rate DDoS attacks. There are still opportunities for improvement. As it stands, the framework primarily addresses UDP DDOS attacks, ICMP Flood, and TCP SYN flood. Expanding its capabilities in these areas would strengthen its overall effectiveness.

CONCLUSION

This research introduces a novel approach for the real-time detection and mitigation of high-rate DDoS attacks within the SDN environment. It employs sFlow technology, a flow statistics-based monitoring for comprehensive traffic analysis. What sets this work apart is its dual capability to detect and mitigate these attacks while actively blocking them as they occur. The DJRT was assessed using a mininet emulator, which provided a virtual SDN environment with an inbuilt host, switches, controller, and connetion among them. The DJRT comprises components- **sFlow**: this statistical tool gathers traffic information from any switch or network device with which it is integrated. It monitors each packet flow through the switch and stores the data in a module known as the collector module. **Collector module**: Its primary function is to collect traffic flow statistics data from network devices, which act as sFlow agents. **Detection Module:** The information collected by the collector module is forwarded to the detection module, which identifies abnormal traffic patterns based on predefined threshold values. **Alleviation module**: This module executes appropriate mitigation actions based on the detection module's output. The crucial part of this research is the custom-built JavaScript, ryu2m.js, designed for the real-time detection and mitigation of multiple high-rate DDoS attacks based on predefined thresholds.

Additionally, a Python script was developed to find the routes where elephant flow occurs, which can also be monitored through the mininet GUI dashboard, an extension of the sFlow-RT application. The proposed model contrasts with previous studies that primarily focused on detecting singular DDoS attacks and lacked effective mitigation measures. The DJRT model not only detects and mitigates multiple high-rate DDoS attacks in a dynamic SDN environment but also continues to monitor network traffic patterns even after blocking the attacker's IP address. Furthermore, incorporating machine learning techniques can enhance the DJRT's ability to manage previously unidentified traffic or attack patterns. By more accurately distinguishing between regular and suspicious traffic, this model can improve the security of the SDN environment.

In conclusion, the proposed model serves as a practical approach in addressing the issue of high-rate DDoS attacks in SDNs. It enhances the security of network infrastructure and supports the development of new technologies aimed at automating SDN security. Its adaptability to evolving threats and ability to improve protection in SDN

environments make it an essential resource for network security and its management, making it a better approach.

Declarations

Availability of data and materials Yes on demand, contact at email: deepak.cs339@gmail.com

Competing interests No conflict of interest among authors.

Funding This research is not funded by any agency or organization.

Authors' contributions DK: Conceptualization, Methodology, Software, Validation, Data curation, Writing – original draft, Writing – review & editing, Investigation. JT: Writing-review and editing, Supervision.

Acknowledgment Not applicable

Ethics statement Not applicable

REFERENCES

[1]. Zaw, H. T., & Maw, A. (2019). Traffic management with elephant flow detection in software defined networks (SDN). *International Journal of Electrical and Computer Engineering (IJECE)*, *9*(4), 3203. https://doi.org/10.11591/ijece.v9i4.pp3203-3211

[2]. Amezquita-Suarez, F., Estrada-Solano, F., da Fonseca, N. L., & Rendon, O. M. (2019). An efficient mice flow routing algorithm for data centers based on software-defined networking. *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 1–6. https://doi.org/10.1109/icc.2019.8761552

[3]. Abdollahi, S., Deldari, A., Asadi, H., Montazerolghaem, A., & Mazinani, S. M. (2021). Flow-aware forwarding in SDN datacenters using a Knapsack-PSO-based solution. *IEEE Transactions on Network and Service Management*, *18*(3), 2902–2914. https://doi.org/10.1109/tnsm.2021.3064974

[4]. Kiran, M., & Chhabra, A. (2019). Understanding flows in high-speed scientific networks: A Netflow Data Study. *Future Generation Computer Systems*, *94*, 72–79. https://doi.org/10.1016/j.future.2018.11.006

[5]. Sharma, A., Tokekar, S., & Varma, S. (2023). A comprehensive survey on Network Resource Management in SDN Enabled Data Centre Network. *6G Enabled Fog Computing in IoT*, 333–353. https://doi.org/10.1007/978-3-031-30101-8_14

[6]. Gelgi, M., Guan, Y., Arunachala, S., Samba Siva Rao, M., & Dragoni, N. (2024). Systematic literature review of IOT botnet DDOS attacks and evaluation of detection techniques. *Sensors*, *24*(11), 3571. https://doi.org/10.3390/s24113571

[7]. Yan, B., Liu, Q., Shen, J., & Liang, D. (2022). Flowlet-level multipath routing based on graph neural network in OpenFlow-based SDN. *Future Generation Computer Systems*, *134*, 140–153. https://doi.org/10.1016/j.future.2022.04.006

[8]. Huang XB, Xue KP, Xing YT et al. An efficient scheme to defend data-to-control-plane saturation attacks in software-defined networking. journal of computer science and technology 37(4): 839–851 July 2022. DOI:10.1007/s11390-022-1495-0

[9]. Karnani, S., Agrawal, N., & Kumar, R. (2023). A comprehensive survey on low-rate and high-rate ddos defense approaches in SDN: Taxonomy, research challenges, and opportunities. *Multimedia Tools and Applications*, *83*(12), 35253–35306. https://doi.org/10.1007/s11042-023-16781-0

[10]. Rios, V. D., Inacio, P. R., Magoni, D., & Freire, M. M. (2022). Detection and mitigation of low-rate denial-of-service attacks: A survey. *IEEE Access*, *10*, 76648–76668. https://doi.org/10.1109/access.2022.3191430

[11]. Hoque, N., & Bhattacharyya, D. K. (2018). HLR_DDoS: A low-rate and high-rate ddos attack detection method using $$\alpha $$α-divergence. *Lecture Notes in Networks and Systems*, 655–662. https://doi.org/10.1007/978-981-10-6890-4_63

[12]. Lawal, B. H., & At, N. (2018). Improving software defined network security via sflow and IPSec protocol. *Anadolu University Journal of Science and Technology-A Applied Sciences and Engineering*, 1–2. https://doi.org/10.18038/aubtda.421939

[13]. Jafarian, T., Masdari, M., Ghaffari, A., & Majidzadeh, K. (2020). A survey and classification of the security anomaly detection mechanisms in software defined networks. *Cluster Computing*, *24*(2), 1235–1253. https://doi.org/10.1007/s10586-020-03184-1

[14]. 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*(4), 1338–1347. https://doi.org/10.1007/s11036-020-01552-0

[15]. Huang, L., Zhi, X., Gao, Q., Kausar, S., & Zheng, S. (2016). Design and implementation of Multicast Routing

System over SDN and sflow. *2016 8th IEEE International Conference on Communication Software and Networks (ICCSN)*. https://doi.org/10.1109/iccsn.2016.7586578

[16]. 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)*. https://doi.org/10.1109/siu.2018.8404674

[17]. Dauer, P., Khondoker, R., Marx, R., & Bayarou, K. (2015). Security analysis of software defined networking applications for monitoring and measurement. *The 10th International Conference on Future Internet*. https://doi.org/10.1145/2775088.2775104

[18]. Dong, P., Du, X., Zhang, H., & Xu, T. (2016). A detection method for a novel ddos attack against SDN controllers by vast new low-traffic flows. *2016 IEEE International Conference on Communications (ICC)*. https://doi.org/10.1109/icc.2016.7510992

[19]. Moustafa, N., Hu, J., & Slay, J. (2019). A holistic review of Network Anomaly Detection Systems: A comprehensive survey. *Journal of Network and Computer Applications*, *128*, 33–55. https://doi.org/10.1016/j.jnca.2018.12.006

[20]. Wang, R., Jia, Z., & Ju, L. (2015). An entropy-based distributed ddos detection mechanism in software-defined networking. *2015 IEEE Trustcom/BigDataSE/ISPA*. https://doi.org/10.1109/trustcom.2015.389

[21]. M. El-Shamy, A., A. El-Fishawy, N., Attiya, G., & A. A. Mohamed, M. (2021). Anomaly detection and Bottleneck Identification of the distributed application in Cloud Data Center using software–defined networking. *Egyptian Informatics Journal*, *22*(4), 417–432. https://doi.org/10.1016/j.eij.2021.01.001

[22]. Giotis, K., Androulidakis, G., & Maglaris, V. (2015). A scalable anomaly detection and mitigation architecture for Legacy Networks via an openflow middlebox. *Security and Communication Networks*, *9*(13), 1958–1970. https://doi.org/10.1002/sec.1368

[23]. Hande, Y., & Muddana, A. (2020). A survey on intrusion detection system for Software Defined Networks (SDN). *Research Anthology on Artificial Intelligence Applications in Security*, 467–489. https://doi.org/10.4018/978-1-7998-7705-9.ch023

[24]. Ujjan, R. M., Pervez, Z., Dahal, K., Khan, W. A., Khattak, A. M., & Hayat, B. (2021). Entropy based features distribution for anti-ddos model in SDN. *Sustainability*, *13*(3), 1522. https://doi.org/10.3390/su13031522

[25]. Hamdan, M., Mohammed, B., Humayun, U., Abdelaziz, A., Khan, S., Ali, M. A., Imran, M., & Marsono, M. N. (2020). Flow-aware elephant flow detection for software-defined networks. *IEEE Access*, *8*, 72585–72597. https://doi.org/10.1109/access.2020.2987977

[26]. Spiekermann, D., & Keller, J. (2020). Impact of virtual networks on anomaly detection with machine learning. *2020 6th IEEE Conference on Network Softwarization (NetSoft)*. https://doi.org/10.1109/netsoft48620.2020.9165325

[27]. LIN, H.-C., & WANG, P. (2016). Implementation of an SDN-based security defense mechanism against ddos attacks. *DEStech Transactions on Economics and Management*, (iceme–ebm). https://doi.org/10.12783/dtem/iceme-ebm2016/4183

[28]. Leal, A., Botero, J. F., & Jacob, E. (2018). Improving early attack detection in networks with sFlow and Sdn. *Communications in Computer and Information Science*, 323–335. https://doi.org/10.1007/978-3-030-00353-1_29

[29]. Aladaileh, M. A., Anbar, M., Hintaw, A. J., Hasbullah, I. H., Bahashwan, A. A., Al-Amiedy, T. A., & Ibrahim, D. R. (2023). Effectiveness of an entropy-based approach for detecting low- and high-rate ddos attacks against the SDN controller: Experimental Analysis. *Applied Sciences*, *13*(2), 775. https://doi.org/10.3390/app13020775

[30]. Zhang, M., Li, G., Wang, S., Liu, C., Chen, A., Hu, H., Gu, G., Li, Q., Xu, M., & Wu, J. (2020). Poseidon: Mitigating volumetric ddos attacks with programmable switches. *Proceedings 2020 Network and Distributed System Security Symposium*. https://doi.org/10.14722/ndss.2020.24007

[31]. Tung, Y.-H., Wei, H.-C., Ti, Y.-W., Tsou, Y.-T., Saxena, N., & Yu, C.-M. (2020). Counteracting UDP Flooding Attacks in SDN. *Electronics*, *9*(8), 1239. https://doi.org/10.3390/electronics9081239

[32]. Shehabat, M. M., & Shurman, M. M. (2024). Detection and mitigation of ICMP-based ddos in software defined networks. *2024 15th International Conference on Information and Communication Systems (ICICS)*, 1–6. https://doi.org/10.1109/icics63486.2024.10638300

[33]. Gao, D., Liu, Z., Liu, Y., Foh, C. H., Zhi, T., & Chao, H.-C. (2018). Defending against packet-in messages flooding attack under SDN context. *Soft Computing*, *22*(20), 6797–6809. https://doi.org/10.1007/s00500-018-3407-3

[34]. 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. https://doi.org/10.1109/ccoms.2019.8821716

[35]. Chang Liu, Gang Xiong, Jie Liu, & Gaopeng Gou. (2015). Detect the reflection amplification attack based on

UDP protocol. *2015 10th International Conference on Communications and Networking in China (ChinaCom)*, 260–265. https://doi.org/10.1109/chinacom.2015.7497948

[36]. Ravi, N., Shalinie, S. M., Lal, C., & Conti, M. (2021). AEGIS: Detection and mitigation of TCP SYN Flood on SDN controller. *IEEE Transactions on Network and Service Management*, *18*(1), 745–759. https://doi.org/10.1109/tnsm.2020.3037124

[37]. Hasan, A., Iqbal, T., Naseer, M., Sarwar, N., Ali, A., & Shabir, M. (2024). Advanced detection and mitigation of Smurf attacks using AI and SDN. *2024 International Conference on Decision Aid Sciences and Applications (DASA)*, 1–6. https://doi.org/10.1109/dasa63652.2024.10836590

[38]. Balaji Bharatwaj, M., Aditya Reddy, M., Senthil Kumar, T., & Vajipayajula, S. (2022). Detection of DOS and ddos attacks using Hidden Markov model. *Lecture Notes in Networks and Systems*, 979–992. https://doi.org/10.1007/978-981-16-5529-6_74

[39]. Khamaiseh, S. Y., Al-Alaj, A., & Warner, A. (2020). Flooddetector: Detecting unknown DOS flooding attacks in SDN. *2020 International Conference on Internet of Things and Intelligent Applications (ITIA)*, 1–5. https://doi.org/10.1109/itia50152.2020.9312310

[40]. Shannon, C., Moore, D., & claffy, k. (2001). Characteristics of fragmented IP traffic on internet links. *Proceedings of the First ACM SIGCOMM Workshop on Internet Measurement Workshop - IMW '01*. https://doi.org/10.1145/505213.505214

[41]. Yihunie, F., Abdelfattah, E., & Odeh, A. (2018). Analysis of ping of death dos and ddos attacks. *2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, 1–4. https://doi.org/10.1109/lisat.2018.8378010

[42]. Yungaicela-Naula, N. M., Vargas-Rosales, C., & Perez-Diaz, J. A. (2021). SDN-based architecture for transport and application layer ddos attack detection by using machine and Deep Learning. *IEEE Access*, *9*, 108495–108512. https://doi.org/10.1109/access.2021.3101650

[43]. Chovanec, M., Havrilla, M., Hasin, M., & Chovancova, E. (2022). Analysis of attacks on mail service infrastructure based on application logs and NetFlow protocol. *2022 IEEE 20th Jubilee World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 000077–000082. https://doi.org/10.1109/sami54271.2022.9780690

[44]. Al-Rushdan, H., Shurman, M., Alnabelsi, S. H., & Althebyan, Q. (2019). Zero-day attack detection and prevention in software-defined networks. *2019 International Arab Conference on Information Technology (ACIT)*, 278–282. https://doi.org/10.1109/acit47987.2019.8991124

[45]. Salatino, F., Spina, M. G., Tropea, M., & De Rango, F. (2024). Detecting ddos attacks through AI driven SDN intrusion detection system. *2024 IEEE 21st Consumer Communications & Networking Conference (CCNC)*. https://doi.org/10.1109/ccnc51664.2024.10454798

[46]. 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*(2), 425–441. https://doi.org/10.1007/s13369-017-2414-5

[47]. 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)*. https://doi.org/10.1109/icsccc51823.2021.9478096

[48]. Singh, J., Jyoti, N., & Behal, S. (2021). On the use of information theory metrics for detecting ddos attacks and Flash events: An empirical analysis, comparison, and Future Directions. *Kuwait Journal of Science*, *48*(4). https://doi.org/10.48129/kjs.v48i4.10612

[49]. Garg, G., & Garg, R. (2016). Security of networks using efficient adaptive flow counting for anomaly detection in SDN. *Advances in Intelligent Systems and Computing*, 667–674. https://doi.org/10.1007/978-81-322-2656-7_61

[50]. Peng, H., Sun, Z., Zhao, X., Tan, S., & Sun, Z. (2018). A detection method for anomaly flow in software defined network. *IEEE Access*, *6*, 27809–27817. https://doi.org/10.1109/access.2018.2839684

[51]. Zekri, M., Kafhali, S. E., Aboutabit, N., & Saadi, Y. (2017). DDoS attack detection using machine learning techniques in cloud computing environments. *2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)*. https://doi.org/10.1109/cloudtech.2017.8284731

[52]. Kumari, K., & Mrunalini, M. (2022). Detecting denial of service attacks using machine learning algorithms. *Journal of Big Data*, *9*(1). https://doi.org/10.1186/s40537-022-00616-0

[53]. Ahuja, N., Singal, G., Mukhopadhyay, D., & Kumar, N. (2021). Automated DDOS attack detection in software defined networking. *Journal of Network and Computer Applications*, *187*, 103108. https://doi.org/10.1016/j.jnca.2021.103108

[54]. Bhayo, J., Shah, S. A., Hameed, S., Ahmed, A., Nasir, J., & Draheim, D. (2023). Towards a machine learning-based framework for DDOS attack detection in software-defined IOT (SD-IOT) networks. *Engineering Applications of Artificial Intelligence*, *123*, 106432. https://doi.org/10.1016/j.engappai.2023.106432

[55]. Niyaz, Q., Sun, W., & Javaid, A. Y. (2017). A deep learning based ddos detection system in software-defined networking (SDN). *ICST Transactions on Security and Safety*, *4*(12), 153515. https://doi.org/10.4108/eai.28-12-2017.153515

[56]. Tang, T. A., Mhamdi, L., McLernon, D., Zaidi, S. A., & Ghogho, M. (2016). Deep Learning Approach for network intrusion detection in software defined networking. *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*. https://doi.org/10.1109/wincom.2016.7777224

[57]. Shahid, W. B., Aslam, B., Abbas, H., Khalid, S. B., & Afzal, H. (2022). An enhanced deep learning based framework for web attacks detection, mitigation and attacker profiling. *Journal of Network and Computer Applications*, *198*, 103270. https://doi.org/10.1016/j.jnca.2021.103270

[58]. Kumar, D., Pateriya, R. K., Gupta, R. K., Dehalwar, V., & Sharma, A. (2023). DDoS detection using Deep Learning. *Procedia Computer Science*, *218*, 2420–2429. https://doi.org/10.1016/j.procs.2023.01.217

[59]. Kushwah, G. S., & Ranga, V. (2020). Voting extreme learning machine based distributed denial of service attack detection in cloud computing. *Journal of Information Security and Applications*, *53*, 102532. https://doi.org/10.1016/j.jisa.2020.102532

[60]. Hnamte, V., & Hussain, J. (2023). DCNNBILSTM: An efficient hybrid deep learning-based Intrusion Detection System. *Telematics and Informatics Reports*, *10*, 100053. https://doi.org/10.1016/j.teler.2023.100053

[61]. Preamthaisong, P., Auyporntrakool, A., Aimtongkham, P., Sriwuttisap, T., & So-In, C. (2019). Enhanced ddos detection using hybrid genetic algorithm and decision tree for SDN. *2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE)*. https://doi.org/10.1109/jcsse.2019.8864216

[62]. Ahuja, N., Singal, G., Mukhopadhyay, D., & Kumar, N. (2021). Automated DDOS attack detection in software defined networking. *Journal of Network and Computer Applications*, *187*, 103108. https://doi.org/10.1016/j.jnca.2021.103108

[63]. Badotra, S., & Panda, S. N. (2020). Snort based early ddos detection system using Opendaylight and open networking operating system in software defined networking. *Cluster Computing*, *24*(1), 501–513. https://doi.org/10.1007/s10586-020-03133-y

[64]. Ujjan, R. M., Pervez, Z., & Dahal, K. (2018). Suspicious traffic detection in SDN with collaborative techniques of Snort and Deep Neural Networks. *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. https://doi.org/10.1109/hpcc/smartcity/dss.2018.00152

[65]. Shayegan, M. J., & Damghanian, A. (2024). A method for ddos attacks prevention using SDN and NFV. *IEEE Access*, *12*, 108176–108184. https://doi.org/10.1109/access.2024.3438538

[66] Ujjan, R. M., Pervez, Z., Dahal, K., Bashir, A. K., Mumtaz, R., & González, J. (2020). Towards sflow and adaptive polling sampling for deep learning based ddos detection in SDN. *Future Generation Computer Systems*, *111*, 763–779. https://doi.org/10.1016/j.future.2019.10.015

[67]. Krishnan, P., Jain, K., Aldweesh, A., Prabu, P., & Buyya, R. (2023). OpenStackDP: A scalable network security framework for SDN-based OpenStack Cloud Infrastructure. *Journal of Cloud Computing*, *12*(1). https://doi.org/10.1186/s13677-023-00406-w

[68]. Ohri, P., Arockiam, D., Neogi, S. G., & Muttoo, S. K. (2024). Intrusion detection and prevention system for early detection and mitigation of ddos attacks in SDN environment. *2024 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, 1–6. https://doi.org/10.1109/sceecs61402.2024.10481906

[69]. Jaiswal, A. K. (2022). *DOS Attack Network Traffic Monitoring in Software Defined Networking Using Mininet and Ryu Controller*. https://doi.org/10.21203/rs.3.rs-2282189/v1

[70]. Ha, T., Kim, S., An, N., Narantuya, J., Jeong, C., Kim, J., & Lim, H. (2016). Suspicious traffic sampling for intrusion detection in software-defined networks. *Computer Networks*, *109*, 172–182. https://doi.org/10.1016/j.comnet.2016.05.019

[71]. Niyaz, Q., Sun, W., & Javaid, A. Y. (2017a). A deep learning based ddos detection system in software-defined networking (SDN). *ICST Transactions on Security and Safety*, *4*(12), 153515. https://doi.org/10.4108/eai.28-12-2017.153515

[72]. Tang, T. A., Mhamdi, L., McLernon, D., Zaidi, S. A., & Ghogho, M. (2016a). Deep Learning Approach for network intrusion detection in software defined networking. *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*. https://doi.org/10.1109/wincom.2016.7777224

[73]. Pang, C., Jiang, Y., & Li, Q. (2016). Fade: Detecting forwarding anomaly in software-defined networks. *2016 IEEE International Conference on Communications (ICC)*. https://doi.org/10.1109/icc.2016.7510990

[74]. Cui, Y., Yan, L., Li, S., Xing, H., Pan, W., Zhu, J., & Zheng, X. (2016). SD-anti-ddos: Fast and efficient ddos

Research Article

defense in software-defined networks. *Journal of Network and Computer Applications*, *68*, 65–79. https://doi.org/10.1016/j.jnca.2016.04.005

[75]. Carvalho, L. F., Fernandes, G., Rodrigues, J. J., Mendes, L. S., & Proenca, M. L. (2017). A novel anomaly detection system to assist network management in SDN environment. *2017 IEEE International Conference on Communications (ICC)*. https://doi.org/10.1109/icc.2017.7997214

[76]. Lee, S., Kim, J., Shin, S., Porras, P., & Yegneswaran, V. (2017). ATHENA: A framework for scalable anomaly detection in software-defined networks. *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. https://doi.org/10.1109/dsn.2017.42

[77]. He, D., Chan, S., Ni, X., & Guizani, M. (2017). Software-defined-networking-enabled Traffic anomaly detection and mitigation. *IEEE Internet of Things Journal*, *4*(6), 1890–1898. https://doi.org/10.1109/jiot.2017.2694702

[78]. Carvalho, Luiz Fernando, Abrão, T., Mendes, L. de, & Proença, M. L. (2018). An ecosystem for anomaly detection and mitigation in software-defined networking. *Expert Systems with Applications*, *104*, 121–133. https://doi.org/10.1016/j.eswa.2018.03.027

[79]. Peng, H., Sun, Z., Zhao, X., Tan, S., & Sun, Z. (2018a). A detection method for anomaly flow in software defined network. *IEEE Access*, *6*, 27809–27817. https://doi.org/10.1109/access.2018.2839684

[80]. Dey, S. K., & Rahman, Md. M. (2018a). Flow based anomaly detection in software defined networking: A deep learning approach with feature selection method. *2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT)*. https://doi.org/10.1109/ceeict.2018.8628069

[81]. Garg, S., Kaur, K., Kumar, N., & Rodrigues, J. J. (2019). Hybrid deep-learning-based anomaly detection scheme for suspicious flow detection in SDN: A Social Multimedia Perspective. *IEEE Transactions on Multimedia*, *21*(3), 566–578. https://doi.org/10.1109/tmm.2019.2893549 .