Journal of Computer Science

Review

DDoS Mitigation Using Machine Learning in Software-Defined Networks

Ahmed Gaber Abu Abd-Allah, Noureldin Omar Mohamed, Mohamed Samir Mohamed, Mahytab Jaafar, Youssef Mohamed Yasser and Mohamed Saeed Taha

Computer Science, Canadian International College, Cairo, Egypt

Article history
Received: 20-04-2024
Revised: 19-08-2024
Accepted: 23-10-2024

Corresponding Author:
Ahmed Gaber Abu Abd-Allah
Computer Science, Canadian International College, Cairo, Egypt
Email: a_gaber_abuabdallah@cic-cairo.com

Abstract: With the escalating frequency and sophistication of Distributed Denial of Service (DDoS) attacks, the realm of cyber security has witnessed a paradigm shift towards innovative solutions. This review explores the potential of machine learning as a powerful defense mechanism in the field of DDoS mitigation within Software-Defined Networks (SDNs). The study methodically examines a variety of machine learning algorithms used for DDoS mitigation, from cutting-edge deep learning approaches to conventional statistical approaches. A key focus of this review is to provide a comparative analysis of different machine learning approaches, evaluating their efficacy in identifying and mitigating DDoS attacks within SDN environments. The discussion encompasses the strengths and limitations of each algorithm, shedding light on their applicability and performance metrics. By dissecting the nuanced differences between these methodologies, the review aims to guide practitioners and researchers toward informed decisions when implementing DDoS mitigation strategies in SDNs. Furthermore, this study addresses the main challenges faced by machine learning-based DDoS mitigation in SDNs. From issues related to real-time detection and adaptability to dynamic attack patterns to the impact of network scale and diversity, the review systematically outlines these challenges and proposes potential avenues for overcoming them. By understanding these hurdles, stakeholders in the field can proactively develop solutions that enhance the robustness and effectiveness of DDoS mitigation frameworks within SDNs. Conclusively, this review stands as an invaluable resource for cyber security professionals, researchers, and policymakers navigating the intricate terrain of DDoS mitigation in Software-Defined Networks. Through a meticulous exploration of machine learning techniques and a discerning analysis of associated challenges, the paper not only provides comprehensive insights but also lays the groundwork for the development of resilient and adaptive security measures against the ever-evolving landscape of cyber threats. By assimilating the knowledge gleaned from this review, stakeholders are empowered to make informed decisions and contribute to the ongoing refinement of DDoS mitigation strategies, ensuring the continued integrity and security of Software-Defined Networks in the face of emerging threats.

Keywords: SDN, Software Defined Networks, DDoS, DDoS Mitigating, Machine Learning, Cyber Security, DDoS Attacks

Introduction

The concept of software-defined networking is not new; This is a complete transformation. Rather, it is the result of collaboration, the development of ideas, and network research. In (Liu *et al*., 2024), three main cases of SDN development are identified: functional coupling (the mid-1990s to early 2000), separation of data and control planes (2001-2007), and OpenFlow API and NOS (2007-2010). All this is discussed below. The challenge for researchers to test new ideas on real architecture and the time, effort, and resources required to implement these ideas at the Internet Engineering Task Force (IETF) will result in some work being done

on network equipment. Active networking provides a programmable network interface, or API, that exposes users to the resources of each network (such as processing, memory resources, and packet processing) and includes the availability nature of identity for packets arriving between nodes. The need to use different models on nodes is the first step in network virtualization research and the development of methods or platforms for building applications on nodes. Active Network Architecture Framework v1.0 (Liu *et al*., 2024) consists of a shared Operating System (NodeOS), a layer of operational environments (Execution Environments (EEs)) and active applications (Active Applications (AA)). NodeOS manages shared resources, while EE defines virtual machines for package operations. AA operates on Energy Efficiency and provides end-to-end services. The separation of packets sent to each EE depends on the pattern in the packet header of the incoming object. This model is used in the PlanetLab platform, where researchers perform experiments in a virtual operating environment and packets are parsed to each virtual destination based on their headers. These developments are especially important in examining network architectures, platforms, and programming models. However, their commercial use is limited and has been criticized mainly for their performance and security limitations. The work proposed by (Li and Louri, 2021) tries to achieve a high level of collaboration, while the secure active network environment architecture (Rose *et al*., 2020) tries to increase its security.

Improve network connectivity, management process and connection (train engineering), traffic forecasting, reaction and rapid response to network problems, etc. This creates the need to use the best management resources, such as management cycle methods. However, the development of this technology is strictly limited by the tight coupling between the hardware and software of the network equipment. It also means that the connection speed (branch networks) constantly improves, and all transmitted packets (packet forwarding) are destined for hardware, controlled separately, or network management for software applications. These applications run best on servers that have higher performance and better memory than a network device. In this sense, the IETF (RFC 3746) standard CES (Separation of Transport and Control Objects) project (Yang *et al*., 2024) creates an interface between data and control planes in network nodes. SoftRouter (Islam *et al*., 2021) Use this software interface to configure messages sent on the router's data plane. In addition, the Routing Control Platform (RCP) (Zhang, 2020) project proposes logical central control of the network, promoting network management, innovation capabilities, and programming. RCP is immediately available because it uses the existing management system Border Gateway Protocol (BGP) to improve entries in routers' routing tables. The separation of the data plane and control plane allows the development of "new" architectures such as 4D Project (Badotra and Panda, 2020) and Ethernet (Abuarqoub, 2020). The architecture shows a four-layer architecture in function: Data plane, plane detection, plane communication, and decision plane. In addition, the Ethanet project (Abuarqoub, 2020) proposed a centralized management system for business connections. However, the need for a switch based on Linux, OpenWrt, and NetFPGA and supporting the Ethan protocol makes it difficult to implement the project. Currently, the OpenFlow protocol (Eliyan and Di Pietro, 2021) is the most widely used protocol in the scientific community and has become the basis of different projects. Companies like Cisco have also called for a new architecture called Cisco Open Network Environment (Cisco ONE).

The term software-defined communication, which was easy to define in the past, shows some changes today. First, the data plane and the control plane are separated or decoupled, ensuring independent development and evolution. Second, it proposes a centralized control plane so that there is a global view of the network. Finally, SDN creates an open connection between the control plane and the data plane. The differences between these architectures are shown in Fig. (1). The network programmability provided by SDN is comparable to mobile applications running on operating systems (Android and Windows Mobile). These applications use the mobile phone's resources (GPS, accelerometer, and memory) through APIs provided by the operating system. Likewise, network administrators can manage and manage network resources through APIs (private or open) available on the controller according to user needs (Scaranti *et al.*, 2020).

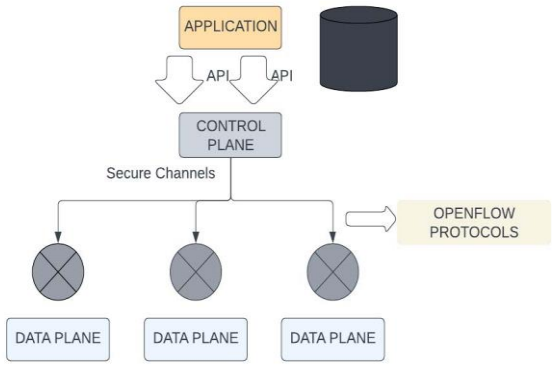

Fig. 1: SDN overview

OpenFlow

OpenFlow (Eliyan and Di Pietro, 2021) was initially proposed as an alternative to the universities' experimental protocol, allowing new algorithms to be tested without disrupting or interfering with a common process among other users' traffic. Today, the Open Networking Foundation (ONF) (Yang *et al*., 2004) is the organization responsible for publishing the OpenFlow protocol and other regulations such as OF Config (Shaghaghi *et al*., 2020) for SDN. The advantage of OpenFlow over the SDN protocol is that it uses hardware and functionality found in most network devices. These elements include reading headers, forwarding packets to ports, sending packets, etc. These are routing tables with various functions such as: OpenFlow exposes concepts and functions; so these can be controlled externally. This means that by updating the firmware, real devices can support OpenFlow. These companies do not need to change their hardware to use SDN in their products and services (Ahmad *et al*., 2020). OpenFlow architecture defines the existence of the controller, OpenFlow switch, and secure communication protocol. These elements are shown in Fig. (2). Each OpenFlow switch has a flow table managed by a controller. Each flow table has three components: Packet header, processing, and statistics. The packet header is like a mask that selects the packets the switch will process. The domain used for comparison can be from layers 2, 3, or 4 of the TCP/IP architecture. This means that there is no distinction between layers as in current architectures. All packets processed by the switch are filtered by this method. The number of zones the exchanger can manage depends on the version of the OpenFlow protocol. OpenFlow v1.0 (Costa *et al*., 2021) (the most used version) has 12 domains, while the latest version of OpenFlow v1.3 means there are 40 domains, including IPv6 support. When the header of the incoming packet matches the packet header of the routing table, the switch matches this mask. There are important actions and choices to be made. The main operations are as follows: Send the packet to a specific port, package the packet send it to the controller, and then release the packet. Some of the processing options are: Sending packets over the connected port (queuing process) or 802.1D processing. If the header of the incoming packet does not match the packet header of the routing table, the switch (depending on its configuration) forwards the packet to the controller for verification and processing. Finally, the statistics file uses counters to collect statistics for management (Phan *et al*., 2016). The OpenFlow protocol defines the following terms related to switches and controllers: Controller-to-switch, symmetric, and asynchronous. The switch message type controller checks the status of the switch. The corresponding message is sent by the controller or switch to initiate a connection or message exchange.

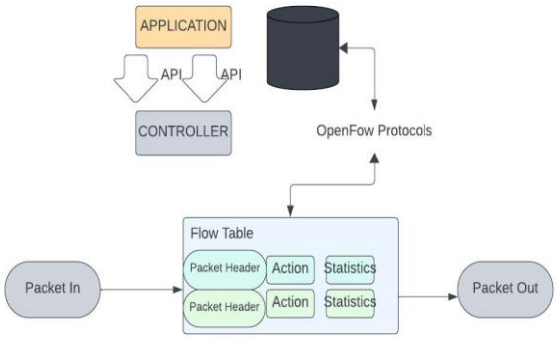

Fig. 2: OpenFlow architecture

Asynchronous messages update network event handling and state transitions. Similarly, OpenFlow generates two types of switches: OpenFlow only and OpenFlow enabled. Only OpenFlow switches use the OpenFlow protocol to process packets. On the other hand, OpenFlow-enabled switches can still use traditional switching or routing algorithms to process packets (von Rechenberg *et al*., 2021). The controller receives data from each variable and adjusts the variable table at that time. Here users can allow the behavior of the network. Active mesh, unlike active mesh, proposes a "node operating system". OpenFlow opens up the concept of Network Operating Systems (NOS). In this case, (Anerousis *et al*., 2021), NOS is defined as the software installed in the network switch that controls the logic and usage status of the network behavior. In recent years, NOS has continued to evolve according to the needs and uses of researchers and network administrators. The concept of a Network Operating System (NOS) is based on the functioning of the operating system in the computer. That is, the operating system allows users to create applications using high levels of data, resources, and hardware. In SDN, some authors (von Rechenberg *et al*., 2021; Hamdan *et al*., 2021) divide the network resources experience into southern and northern (Fig. 3). The function of the south side is to understand the function of the programmable switch and connect to the control software. An obvious example of the south side is OpenFlow. You run the network operating system on the south interface. NOX is an example of NOS (Wang, 2022; Qin, 2021) and others. The north side allows applications or advanced network requirements to be easily created and these functions are transferred to the Network Operating System (NOS). Examples of these interfaces include Frenetic (Foster *et al*., 2016), Procera (Foster *et al*., 2019), (Voellmy *et al*., 2012), Netcore (Monsanto *et al*., 2017) and McNettle (Voellmy and Wang, 2012). They were later detected in the main NOS and the northern part (Polat *et al*., 2020). SDN has become an alternative to traditional security systems due to the easy network

management it provides. However, if the SDN framework is compromised, the security of the system is also compromised. Ordinary controllers also create the same point of failure. Therefore, an attack on the controller will cause the entire network to collapse (Abhiroop *et al*., 2018). The main security issues in SDN are rogue controllers (intrusion), man-in-the-middle attacks, and policy changes that alter packets. Other related issues include malicious packets hijacking controllers, denial of service due to a flood of changes, and communications and configuration issues. Distributed Denial of Service (DDoS) is one of the most dangerous and feared threats designed to completely prevent traffic from reaching the controller. This attack works by inserting more malicious packets into the controller than it can handle, causing the controller to fail. The attack is done using various bots to create malicious packets. The attacker creates a botnet, a group of bots, from switches connected to a controller and then takes control of the entire network after disabling the controller.

Background and Related Works

The authors performed a systematic literature review (SLR) to examine current methodologies for detecting and mitigating distributed denial of service (DDoS) attacks in software-defined networking (SDN) environments (Phan *et al*., 2017). They followed a predefined SLR protocol to search, select, and analyze 70 primary studies published from 2014-2022 that used machine learning (ML), deep learning (DL), or hybrid methods to address the DDoS problem in SDN. They classified the existing approaches into three categories: ML-based, DL-based, and hybrid-based, and discussed their strengths, weaknesses, and limitations. They also examined the evaluation metrics, datasets, network simulators, hacking tools, and experimental platforms used in the literature. Moreover, they identified the challenges, open issues, and future research directions for DDoS detection in SDN networks. The main contribution of this study is to provide a comprehensive and critical overview of the state-of-the-art approaches and to highlight the research gaps and opportunities in this area.

Table (1) shows that most of the methods in the literature are in the hybrid category, followed by the machine learning category, and then by the integrated machine learning category. In addition, while most researchers used self-generated real-world data to evaluate and train their planning, due to the lack of benchmark data for SDN DDoS attacks, few other researchers used it for the Untruth collection. It is public information. In addition, most studies use a special selection process to select the best features to improve to ensure the accuracy and distribution of network connections. However, some studies such as (Nadeem *et al*., 2022; Dong and Sarem, 2020) do not use them. At the same time, most work continues on SDN controllers and imposes unnecessary overhead on the controller, e.g., (Sahoo *et al*., 2020; Alamri and Thayananthan, 2020; De Assis *et al*., 2018; Nurwarsito and Nadhif, 2021). In addition, some researchers have opined that SDN controllers can reduce the load and overhead, especially during DDoS attacks, for example (Perez-Diaz *et al*., 2020; Cui *et al*., 2016). In contrast, some studies do not provide detailed information on where their methods were used, for example (Santos *et al*., 2020; Oo *et al*., 2020; Sahoo *et al*., 2018). Furthermore, the majority of techniques either target or mitigate DDoS attacks, whereas very few do both (Sahoo *et al*., 2018; Hannache and Batouche, 2020). Furthermore, the majority of machine learning techniques are only capable of identifying or thwarting DDoS attacks (Ahuja *et al*., 2021; Tonkal *et al*., 2021; Tan *et al*., 2020), which may be accomplished with great precision because of the volume of malicious traffic. However, only a few machine learning techniques, like (Swami *et al*., 2021; Cui *et al*., 2016), are able to identify low-level DDoS attacks in SDN networks (Maheshwari *et al*., 2022; Firdaus *et al*., 2020; Ahuja *et al*., 2021; Sangodoyin *et al*., 2021).

Table 1: Current Approaches and Methodologies

Ref.	ML. based approach	Realistic dataset	Feature selection technique	Deployment of detection approach	DDoS attack techniques	Rate of attack	Detection accuracy	Limitations
	E or H or S			In or Out	D or M	High or Low		
Phan *et al*. (2017)	E	No	Yes	In	D, M	High	High	The suggested methodology was assessed using implausible datasets that failed to represent the attributes of the SDN network. Which did not reflect the characteristics of the SDN network

Nadeem *et al*. (2022)	E	No	Yes	In	D	High	High	The proposed approach was assessed using datasets that lacked realism and did not accurately represent the characteristics of the SDN
Dong and Sarem (2020)	E	Yes	Yes	In	D	High	High	Testing the suggested model's ability to identify such assaults is not as good as putting it into practice on an actual SDN network
Sahoo *et al*. (2020)	E	No	Yes	In	D	High	Low	Unrealistic datasets that did not accurately represent the properties of the SDN network were used to evaluate the suggested methodology. The overall strategy performed poorly
Alamri and Thayananthan (2020)	H	Yes	Yes	In	D	High	High	The strategy is restricted to heavy DDoS attacks, which can be accurately predicted because of high frequency
De Assis *et al*. (2018)	H	No	Yes	In	D	High	High	An artificial dataset that does not accurately represent the nature of the SDN network environment was used to train and test the method
Nurwarsito and Nadhif (2021)	H	Yes	Yes	Out	D	High	High	Since DDoS attacks on the SDN controller have an impact on a global scale, the DAD is restricted to handling SYN DDoS flood attacks on data plans. We used a small dataset to train and test the proposed model
Perez-Diaz *et al*. (2020)	H	No	Yes	In	D, M	High	High	Realistic datasets were used for testing and training the proposed model, although these datasets do not represent the properties of SDN networks
Cui *et al*. (2016)	H	Yes	Yes	In	D	High, low	High	This method can only be used to counter TCP-SYN flood attacks. A small dataset was used to examine the proposed approach

Santos *et al.* (2020)	H	Yes	Yes	In	D	High	High	Tests should be conducted on a genuine SDN testbed, as this would be the preferred method. It is restricted to high-rate DDoS attacks, which are easily identifiable due to the heavy network traffic flow
Oo *et al.* (2020)	H	Yes	Yes	In	D	High	High	Because ML is most effective when control parameters or hyper-parameters are fine-tuned or optimized, it was trained using the default values
Sahoo *et al.* (2018)	H	Yes	Yes	In	D, M	High	High	The method does not evaluate its outcomes against alternative methods. The proposed system operates on the controller, introducing additional load and overhead to it
Hannache and Batouche (2020)	H	No	Yes	—	D	High	High	The residual algorithms exhibit suboptimal performance. The suggested method was assessed using an artificial dataset that fails to represent the characteristics of the SDN network environment
Ahuja *et al.* (2021)	H	No	Yes	In	D	High	Low	The residual algorithms exhibit suboptimal performance. The suggested method was assessed using a non-representative dataset that fails to embody the attributes of the SDN network environment
Tonkal *et al.* (2021)	H	Yes	Yes	In	D	High	High	The remaining machine learning classifiers have comparatively low performance in terms of detection accuracy. The proposed approach's false positive rate is unspecified
Tan *et al.* (2020)	H	No	Yes	Out	D, M	Low	Low	The remaining machine learning

								classifiers have comparatively low performance in terms of detection accuracy. The proposed approach's false positive rate is not disclosed
Swami *et al*. (2021)	H	Yes	Yes	In	D, M	High	High	Due to the large volume of forgeries, high-rate DDoS attacks are easier to spot and this is where the framework is tested. The controller's job is increased by the suggested architecture
Myint Oo *et al*. (2019)	H	No	Yes	In	D, M	High	High	The controller is subjected to a superfluous burden and overhead as a result of the proposed system's operation. The proposed method is burdened by a substantial amount of processing and communication overhead
Maheshwari *et al*. (2022)	H	Yes	Yes	In	D, M	High	High	The proposed method applied to the SDN controller introduces an additional challenge in the context of DDoS attacks
Firdaus *et al*. (2020)	H	Yes	No	–	D	High	Low	The suggested method continues to demonstrate inferior performance in detecting DDoS attacks and requires implementation on an actual SDN network to evaluate its efficacy in identifying these attacks
Sangodoyin *et al*. (2021)	H	No	Yes	In	D	High	Low	The suggested method demonstrates inadequate performance and requires enhancement. The suggested methodology was assessed using an artificial dataset that fails to represent the attributes of the

								SDN network
Bendale *et al*. (2018)	H	Yes	Yes	–	D	High	High	The proposed method demonstrated that the controller DDoS assault yielded the lowest classification results for SVM and MLP, with accuracy rates below 90% compared to flow-table and bandwidth attacks. The proposed approach's false positive rate is unspecified
Xu *et al*. (2019)	H	No	Yes	Out	D, M	High	High	A dataset that does not represent the features of an SDN network environment was used for the defense system's evaluation, testing, and training
Dayal and Srivastava (2017)	H	Yes	Yes	In	D, M	High	High	There is more work and impose added by the proposed model since it operates on the controller. A dataset that does not represent the features of an SDN network environment was used for the defense system's evaluation, testing, and training
Caraguay *et al*. (2016)	H	No	Yes	In	D	High	High	The proposed method employed at the SDN controller elevates the controller's overhead during a DDoS attack
Kokila *et al*. (2014)	S	Yes	Yes	In	D, M	High	High	The suggested method operates on the SDN controller as an application system, introducing superfluous burden and overhead, especially during DDoS assaults on the controller
Deepa *et al*. (2019)	S	Yes	Yes	–	D	High	Low	The ASVM approach demonstrates suboptimal efficacy in identifying DDoS attacks
Ali *et al*. (2023)	S	Yes	Yes	In	D	Low	High, Low	The suggested method operates on the SDN controller

								as an application system, introducing superfluous burden and overhead, especially during DDoS assaults on the controller
Musumeci *et al*. (2022)	S	Yes	Yes	In	D	High	Low	Implementing the proposed solution at the controller results in an increase in load and overhead that is not warranted. In addition to that, it needs to be improved in terms of the accuracy of detection
Gonzalez and Charfadine (2023)	S	No	No	–	D	High	High, Low	The proposed approach demonstrated low accuracy in its performance. The proposed approach was tested and trained on a dataset that does not ac accurately represent the characteristics of the SDN network environment
Fernandes *et al*. (2019)	S	Yes	Yes	IN	D	High	Low	Due to the fact that the scheme needs to be implemented on each and every switch, the suggested solution suffers from a scalability issue. This is because the implementation of the scheme is required on every switch
Freytsis *et al*. (2024)	S	Yes	Yes	In	D, M	High	High	The suggested approach has processing and communication overhead at the SDN controller. The suggested technique was assessed and trained using an unrealistic NSL-KDD dataset, which fails to represent the characteristics of the SDN network environment
Ajaeiya *et al*. (2017)	S	Yes	No	In	D	High	High	The proposed model operates at the controller, resulting in increased load and overhead on the

								SDN controller during DDoS attacks. Information regarding the testing and training datasets is insufficient
Firouzi and Rahmani (2022)	S	Yes	Yes	In	D, M	High	Low	The suggested method exhibits a detection accuracy of 96.13%, attributable to significant false-positive and false-negative rates
Satheesh *et al*. (2017)	S	No	Yes	–	D	High	High	The proposed methodology has been assessed using an unrealistic dataset that fails to represent the attributes of the SDN network environment
Nain *et al*. (2014)	S	No	Yes	–	D	High	High	It takes additional time to examine and classify all packet flows that enter via the data layer's OpenFlow switches since they feature an attack-detection module
Hesamifard *et al*. (2018)	S	Yes	Yes	Out	D, M	High	High	Due to the SDN controller collecting all flows from the switches for detection reasons, this system still has overhead in heavy DDoS assault flows, leading to congestion and a reduction of response time

DDoS Attacks

DDoS is a serious attack that has a major impact on network operations. It affects the traffic between the switch and the controller. An attacker can manipulate the key and engage the controller with many queries (Bendale *et al*., 2018). The attack is depicted below in Fig. (3). Within the context of this assault, the attacker targets the server by bombarding it with a large number of requests, which ultimately results in the server crashing. The server is unable to receive requests from users who are authorized to access it, which leads to a distributed denial of service assault.

One of the most significant security concerns with SDN is the possibility of Distributed Denial of Service (DDoS) attacks. The network's functionality will be severely compromised by this assault. By intercepting network traffic, malicious actors are able to attack services, denying service to innocent consumers. DDoS attacks in SDN can compromise both the data and control systems of airplanes. The controller will be occupied with requests during an attack on the control plane. Xu *et al*. (2019), adversely affecting the computation time. The controller is the nerve center of every network, therefore any assault on it can have far-reaching consequences. We will not accept legal claims. If the controller notices an unusually high volume of traffic coming from only one location, it may be a sign of a denial-of-service attack. Downtime is easily detectable. The continual monitoring and updating of regulations on SDN makes fraud less prevalent. Using the key to launch a Man in the Middle attack is possible. Forging the key is another way that attackers can get inside the system. This controller needs to be able to recognize each switch before letting it into the network in order to circumvent SDN. Two main types of distributed denial of service attacks are the data plane and control plane distributed denial of service attacks (Dayal and Srivastava, 2017; Caraguay *et al*., 2016).

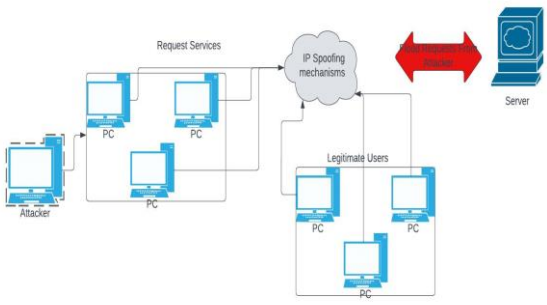

Fig. 3: Spoofing (DDoS) attack

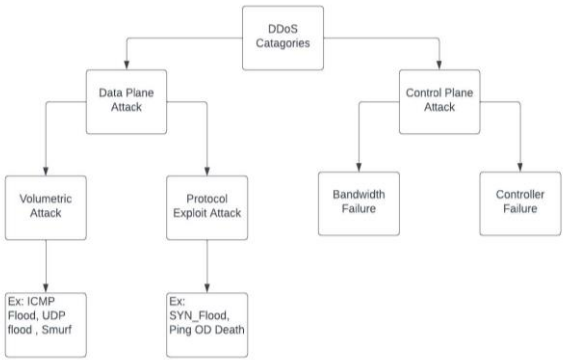

Fig. 4: Categories of DDoS Attack

Data Plane DDoS

There are two categories as shown in Fig. (4), in these categories: Volumetric attacks and protocol attacks. In a volumetric attack, the attacker sends a large number of requests to the victim. Therefore, the victim tries to complete these requests and eventually, the victim gets overwhelmed by the packets received and therefore loses the request and cannot complete it. Essentially, this attack makes a large number of requests to the victim causing the server to fail. Examples of this attack are Internet Control Message Protocol (ICMP) Flood, User Datagram Protocol (UDP) Flood, and Smurf attack. The purpose of using the protocol is tools resources and applications, that is, the main action of the victim will affect its work as follows: Consumes memory, bandwidth, and other resources. The Synchronized (SYN) flood assault is an example of this type of attack, in which the attacker sends SYN packets repeatedly without waiting for an acknowledgment (ACK) from the recipient. Eventually, the server allocates memory for SYN requests and runs out of memory, which leads to network failure. The death ping is another instance of this kind. However, the activity of the program uses resources to generate connection requests, which finally overwhelms the victim and prevents an extended request from being approved (Kokila *et al*., 2014).

Control Plane DDoS

Such attacks pose a threat to aircraft control. Two things cannot accomplish what the attacker is trying to do sending a flood of random streams to the switch until it misses the attack and sends a Packet-In message. You must process the message or else the data bundle will be incomplete. An additional form of attack impacts the OpenFlow protocol's bandwidth. Because the controller has no control over the request, a Packet-In message is sent in this instance. As a result, the network will become unresponsive due to a lack of available bandwidth (Shaghaghi *et al*., 2020).

DDoS Mitigation

Due to the fact that normal users might occasionally be confused with attackers, it is impossible to totally eliminate distributed denial of service assaults. It is necessary to take preventative measures after a distributed denial of service assault has been identified. In order to protect against distributed denial of service attacks, it is possible to identify the attacker and stop them from sending any additional requests. We are able to add policies thanks to SDN. In addition to the modifications, legal notifications have been included, which enables us to restrict people. IP and MAC addresses can be blocked with the use of firewalls. For instance, when the client needs to create a TCP connection to the server, it can expressly limit the number of SYN packets that can be transmitted to the server at the same time during the connection establishment process. It is possible to ban nodes that exceed these restrictions by creating firewall rules that are based on IP and MAC addresses. For an extended period of time, the node will be prevented from issuing requests.

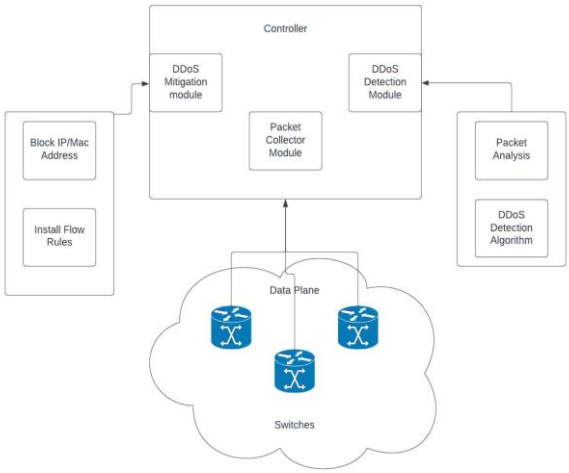

Fig. 5: DDoS mitigation architecture

The above Fig. (5) shows the basic functionality of the control plane's DDoS attack mitigation architecture. There is a module in the control plane that collects traffic from the network. This traffic data is transferred to the DDoS detection module, where data analysis and detection algorithms are used to see if the network is under attack. If it is detected that the network is attacked, the interrupt is informed about the attack and the module takes the necessary actions for the attack. These actions include blocking suspicious hosts.

Machine Learning Approaches for DDoS Detection and Mitigation

Flow-Based Anomaly Detection

Network flows: A flow essentially represents a unidirectional stream of data between two network endpoints. Anomaly detection: Anomaly detection involves identifying patterns or behaviors that deviate from what is considered normal or expected. In the context of network flows, anomalies may indicate potential DDoS attacks. Unsupervised machine learning: Unsupervised machine learning algorithms are often employed for flow-based anomaly detection. Unlike supervised learning, unsupervised learning doesn't rely on labeled training data but instead attempts to identify patterns or anomalies without prior knowledge of what is "normal." Training phase: During the training phase, the system observes the characteristics of normal network behavior by analyzing historical flow data. Features such as the number of packets per flow, the duration of flows, and other relevant attributes are considered. Model building: The unsupervised machine-learning algorithm builds a model of normal behavior based on the observed features. Detection phase: In the detection phase, the system continuously monitors incoming network flows in real time. As new flows are observed, the system compares their characteristics to the learned model of normal behavior. Anomaly identification: If a flow exhibits characteristics significantly different from the learned normal behavior (Deepa *et al*., 2019), it may be flagged as an anomaly. The degree of deviation from the norm can be quantified using statistical measures. Alert or Mitigation: When anomalies are identified, the SDN controller or security system can trigger an alert or take automated mitigation actions. Mitigation actions may include rerouting traffic, blocking specific flows, or dynamically adjusting security policies. Adaptability: Flow-based anomaly detection is adaptive and can respond to changes in network traffic patterns over time. This adaptability is crucial for effectively identifying new and evolving DDoS attack strategies. Limitations: It's important to note that while flow-based anomaly detection is effective, it may have limitations in detecting subtle or low-rate DDoS attacks. Combining flow-based detection with other techniques can enhance overall DDoS mitigation capabilities. Flow-based anomaly detection leverages unsupervised machine learning to analyze network flows, identify deviations from normal behavior and trigger timely responses to potential DDoS attacks in SDN environments (Zhang, 2020; Ali *et al*., 2023).

Statistical Analysis for DDoS Mitigation in SDN

Monitoring Statistical Metrics: In this approach, various statistical metrics related to network traffic are continuously monitored. These metrics include packet rates, traffic volume, protocol distribution, and other statistical characteristics of the network. Supervised or unsupervised machine learning: Statistical analysis for DDoS mitigation can utilize both supervised and unsupervised machine learning techniques, depending on the specific requirements and available data. Training phase (for supervised learning): In the case of supervised learning, the system is trained on labeled datasets that include examples of normal and malicious network behavior. The model learns to differentiate between the two based on the provided labels. Feature selection: Relevant features, such as packet rates, traffic volume per protocol, or distribution of source/destination IP addresses, are selected for analysis. Feature selection is crucial to focus on the most informative aspects of the data. Model building (for supervised learning): Supervised machine learning algorithms, like support vector machines (SVM) or neural networks, are trained to recognize patterns associated with normal and malicious network behavior. The model aims to generalize from the training data to make accurate predictions on new, unseen data (Musumeci *et al*., 2022). Real-time monitoring: In the deployment phase, the system continuously monitors real-time network traffic, collecting statistical metrics as new data flows through the SDN infrastructure. Deviation detection (for unsupervised learning): In the case of unsupervised learning, the system analyzes the statistical metrics without pre-existing labels. Deviations from established statistical norms are identified as potential anomalies or signs of a DDoS attack. Alerting or mitigation actions: When significant deviations or anomalies are detected, the SDN controller or security system can trigger alerts or automated mitigation actions. Mitigation actions may include traffic redirection, rate limiting, or the dynamic adjustment of security policies. Dynamic adaptation: Statistical analysis allows for dynamic adaptation to changing network conditions. As DDoS attacks evolve, the statistical model can be updated to account for new patterns of malicious behavior. Combination with other techniques: Statistical analysis is often used in conjunction with other machine learning and DDoS mitigation techniques to provide a comprehensive defense strategy. Statistical analysis leverages machine learning techniques to continuously monitor and analyze

statistical metrics of network traffic, identifying deviations from normal patterns and enabling timely responses to potential DDoS attacks in SDN environments (Zhang, 2020; Ali *et al*., 2023).

Traffic Classification for DDoS Mitigation in SDN

Distinguishing Legitimate and Malicious Traffic: Traffic classification involves the process of distinguishing between legitimate and potentially malicious traffic based on various characteristics, such as packet size, protocol type, source/destination IP addresses, and other relevant attributes. Supervised machine learning: Supervised machine learning techniques are commonly employed for traffic classification. During the training phase, the system is provided with labeled datasets containing examples of both normal and malicious traffic. Feature extraction: Relevant features, such as the size of packets, source and destination IP addresses, and protocol types, are extracted from the network traffic data. These features serve as input for the machine-learning model. Training phase: The system uses the labeled training data to train a machine learning model, such as a support vector machine (SVM), decision tree, or neural network. The model learns to associate specific patterns of features with either normal or malicious traffic. Real-time traffic analysis: During the deployment phase, the trained model is applied to real-time network traffic. As new traffic flows through the SDN infrastructure, the model classifies each flow as either normal or potentially malicious. Thresholds and decision-making: The model may employ predefined thresholds or decision boundaries to classify traffic. If a flow exhibits characteristics indicative of a DDoS attack (e.g., high packet rates, unusual patterns), it may be classified as malicious. Alerting or mitigation actions: When malicious traffic is identified, the SDN controller or security system can trigger alerts or automated mitigation actions. Mitigation actions may include blocking or redirecting the identified malicious flows. Adaptability: Supervised machine learning models can be adaptive, allowing for updates to the training data and retraining of the model to adapt to evolving DDoS attack strategies. Ensemble methods: Ensemble methods, which combine multiple machine learning models, can enhance the accuracy and robustness of traffic classification. Different models may focus on different aspects of the traffic characteristics. Integration with SDN policies: The results of traffic classification can be integrated with SDN policies to dynamically adjust routing or apply security measures based on the identified classification of traffic. Traffic classification leverages supervised machine learning to distinguish between normal and potentially malicious traffic in real time. By training models on labeled datasets, the system can make informed decisions about the nature of network traffic and take appropriate actions to mitigate the impact of DDoS attacks in SDN environments (Zhang, 2020; Ali *et al*., 2023).

Behavioral Analysis for DDoS Mitigation in SDN

Observing behavioral patterns: Behavioral analysis involves the continuous observation of the behavior of network entities, such as end-hosts, applications, or communication patterns between different components within the SDN infrastructure. Machine learning models: Machine-learning models, such as neural networks, decision trees, or clustering algorithms, can be employed for behavioral analysis. These models are trained to recognize normal patterns of behavior based on historical data. Feature extraction: Relevant features representing behavioral aspects are extracted from the network data. These features may include communication patterns, traffic volume, frequency of interactions, and other behavioral indicators. Training phase: During the training phase, the machine-learning model learns to associate specific patterns of features with normal behavior. This involves using labeled datasets where normal and abnormal behaviors are identified. Real-time behavioral monitoring: In the deployment phase, the system continuously monitors the real-time behavior of network entities. As new data flows through the SDN infrastructure, the model assesses whether the observed behavior aligns with what was learned during the training phase. Anomaly detection: Deviations from the learned normal behavior are flagged as potential anomalies. Behavioral analysis is particularly effective in identifying subtle and evolving DDoS attacks that may not exhibit easily detectable signature patterns. Alerting or Mitigation Actions: When anomalies indicative of a potential DDoS attack are detected, the SDN controller or security system can trigger alerts or automated mitigation actions. Mitigation actions may include isolating the affected entities, rerouting traffic, or dynamically adjusting security policies. Adaptive learning: Behavioral analysis is adaptive and can learn from new behavioral patterns over time. This adaptability is crucial for effectively identifying emerging DDoS attack strategies that may evolve or change their behavior. Combination with other techniques: Behavioral analysis is often used in conjunction with other DDoS mitigation techniques to provide a comprehensive defense strategy. Combining multiple techniques can enhance the overall accuracy and robustness of the DDoS detection system. Granular Insights: Behavioral analysis provides granular insights into the dynamics of network behavior, allowing for a more nuanced understanding of normal and abnormal activities within the SDN infrastructure. Behavioral analysis leverages machine learning models to continuously monitor and analyze the behavioral patterns of network entities, enabling the detection of anomalies that may indicate DDoS attacks. The adaptive

nature of behavioral analysis makes it well-suited for identifying evolving threats in SDN environments (Zhang, 2020; Ali *et al*., 2023).

Feature Extraction and Dimensional Reduction for DDoS Mitigation in SDN

Feature extraction: Feature extraction involves selecting and transforming relevant information, or features, from the raw data. In the context of DDoS mitigation, features may include attributes such as packet rates, traffic volume, protocol distribution, and other characteristics of network flows. Machine learning models: Machine-learning models, such as decision trees, neural networks, or clustering algorithms, benefit from having a set of informative features to make accurate predictions or classifications. Complexity of data: Network data can be complex, containing a large number of attributes. Some of these attributes may be redundant, irrelevant, or noisy. Feature extraction aims to capture the most important information while reducing the dimensionality of the data. Dimensional reduction techniques: Dimensional reduction techniques, such as Principal Component Analysis (PCA) or autoencoders, are applied to further reduce the number of features. These techniques help in retaining the most significant information while discarding less critical aspects. Training phase: During the training phase, the selected features and reduced-dimensional representations are used to train machine-learning models. This process helps the models learn the patterns associated with normal and malicious network behavior. Real-time processing: In the deployment phase, the system continuously processes real-time network data using the extracted features and reduced-dimensional representations. This enables efficient and quick analysis, crucial for the timely detection of DDoS attacks. Anomaly detection or classification: The machine learning models leverage the extracted features to detect anomalies or classify network flows as normal or potentially malicious. Dimensional reduction contributes to computational efficiency and prevents issues associated with the curse of dimension. Alerting or mitigation actions: When anomalies are identified or specific classifications are made, the SDN controller or security system can trigger alerts or automated mitigation actions. Mitigation actions may include rerouting traffic, blocking specific flows, or dynamically adjusting security policies. Adaptability: Feature extraction and dimensional reduction contribute to the adaptability of the DDoS mitigation system by focusing on the most relevant information. This adaptability is essential for handling variations in network traffic and attack strategies. Integration with other techniques: Feature extraction and dimensional reduction are often integrated into a broader DDoS mitigation strategy that may include other techniques, such as flow-based anomaly detection or traffic classification. The combination of these techniques enhances the overall effectiveness of the defense mechanism. Feature extraction and dimensional reduction are critical processing steps in DDoS mitigation, enabling machine-learning models to efficiently analyze network data, identify relevant patterns, and make informed decisions regarding the presence of DDoS attacks in SDN environments (Zhang, 2020; Ali *et al*., 2023).

Reinforcement Learning for DDoS Mitigation in SDN

Decision-making through interaction: Reinforcement learning involves training an algorithm to make decisions through interactions with an environment. The algorithm learns by receiving feedback in the form of rewards or penalties based on the actions it takes. Dynamic and adaptive responses: In the context of DDoS mitigation in SDN, reinforcement learning enables the system to dynamically and adaptively respond to evolving threats by learning optimal strategies over time. Training environment: The SDN environment serves as the training ground for the reinforcement-learning algorithm. The algorithm interacts with the SDN infrastructure, making decisions related to DDoS mitigation, and receives feedback based on the effectiveness of its actions. State Representation: The state of the SDN environment, including network traffic patterns, system resources, and security policies, is represented by the reinforcement-learning algorithm. This representation helps the algorithm understand the current conditions and make informed decisions. Action space: The algorithm has an action space, representing the set of possible actions it can take in response to the observed state. Actions may include adjusting security policies, rerouting traffic, or dynamically allocating resources to mitigate the impact of DDoS attacks. Rewards and penalties: The algorithm receives rewards or penalties based on the consequences of its actions. For example, successful mitigation actions may yield positive rewards, while ineffective responses may result in penalties. Learning policy: The reinforcement-learning algorithm iteratively adjusts its policy, a mapping from states to actions, to maximize the cumulative reward over time. This learning process enables the algorithm to discover effective strategies for mitigating DDoS attacks. Adaptation to changing threats: Reinforcement learning excels in adapting to changing and dynamic environments. As DDoS attack strategies evolve, the algorithm can learn and update its policy to counter new threats effectively. Real-time decision-making: Reinforcement learning facilitates real-time decision-making. The trained algorithm can quickly assess the current state of the SDN environment and take appropriate actions to mitigate DDoS attacks without requiring extensive manual intervention. Integration with

other techniques: Reinforcement learning can be integrated with other machine learning and DDoS mitigation techniques to form a comprehensive defense strategy. It complements other methods by providing adaptive and autonomous decision-making capabilities. Reinforcement learning enables autonomous decision-making in response to DDoS attacks by training algorithms to learn optimal strategies through interactions with the SDN environment. This approach enhances the adaptability and effectiveness of DDoS mitigation in SDN environments (Zhang, 2020; Ali *et al*., 2023).

Real-Time Analysis and Decision-Making for DDoS Mitigating in SDN

Immediate response requirements: DDoS attacks can have rapid and severe impacts on network performance, necessitating quick detection and mitigation. Real-time analysis focuses on processing data and making decisions promptly to respond to ongoing threats. Continuous monitoring: The SDN system continuously monitors incoming network traffic and other relevant parameters in real time. This constant surveillance allows for immediate awareness of changes in network behavior. Machine learning models suitable for real-time: To facilitate real-time analysis, lightweight machine learning models are often chosen. These models are designed for efficiency and quick processing, making them suitable for timely decision-making in dynamic environments. Feature extraction and simplified models: Techniques such as feature extraction and simplified machine learning models contribute to real-time capabilities. Extracting essential features and reducing model complexity helps in processing data swiftly without compromising accuracy. Thresholds and trigger mechanisms: Real-time analysis involves setting thresholds or trigger mechanisms that, when surpassed, signal potential DDoS attacks. These thresholds are determined based on the expected behavior of the network under normal conditions. Automated Alerting Systems: When anomalies or potential DDoS attack patterns are detected, automated alerting systems can promptly notify administrators or trigger predefined mitigation actions. Alerts provide timely information to initiate a response. Automated mitigation actions: Automated mitigation actions, such as rerouting traffic, blocking malicious flows, or dynamically adjusting security policies, can be triggered in real time based on the analysis of incoming data. These actions aim to minimize the impact of DDoS attacks as quickly as possible. Adaptability to changes: Real-time analysis requires systems to adapt swiftly to changes in network conditions or attack strategies. The use of adaptive machine learning models and continuous updates to detection mechanisms contribute to this adaptability. Minimizing detection latency: The goal of real-time analysis is to minimize detection latency the time it takes to identify a potential DDoS attack. Quick detection enables faster initiation of mitigation measures, reducing the overall impact on network performance. Integration with SDN controllers: Real-time analysis is integrated with SDN controllers, allowing for seamless communication between the detection system and the SDN infrastructure. This integration ensures that mitigation actions can be implemented directly within the SDN environment. Real-time analysis and decision-making involve continuous monitoring, quick processing of data, and automated responses to potential DDoS attacks. This approach is essential for minimizing the impact of attacks in SDN environments where rapid adaptation to changing conditions is crucial (Zhang, 2020; Ali *et al*., 2023).

Collaborative and Distributed Defense for DDoS Mitigation in SDN

Shared Information among Components: Collaborative and distributed defense involves the sharing of information and coordination among multiple components within the SDN infrastructure. This collaboration enhances the overall effectiveness of DDoS mitigation. SDN controllers and network devices: SDN controllers and network devices, such as switches and routers, work together to detect and mitigate DDoS attacks. Collaboration ensures that information about potential threats is communicated seamlessly across the network. Shared threat intelligence: The various components exchange threat intelligence, including information about ongoing DDoS attacks, patterns of malicious traffic, and effective mitigation strategies. This shared intelligence enhances the collective defense posture. Coordinated response: In the event of a detected DDoS attack, the SDN controllers and network devices collaborate to formulate a coordinated response. This may involve dynamically adjusting routing tables, redistributing traffic, or activating specific security policies. Dynamic policy adjustments: Collaboration enables the dynamic adjustment of security policies across the SDN infrastructure. Policies can be updated in real time based on the shared threat intelligence and the evolving nature of the DDoS attack landscape. Load distribution and redundancy: Collaborative defense mechanisms can involve redistributing network traffic across multiple paths to balance the load and ensure redundancy. This approach helps prevent congestion at specific points targeted by DDoS attacks. Communication protocols: Effective communication protocols are established to facilitate information exchange among SDN components. These protocols allow for seamless coordination and quick dissemination of threat information. Adaptive decision-making: Collaboration enables adaptive decision-making. As DDoS attack strategies evolve, the collaborative defense system can

collectively adapt by sharing insights and implementing new mitigation measures. Scalability: Collaborative and distributed defense mechanisms are scalable. This scalability is crucial for protecting large-scale SDN environments. Resilience against sophisticated attacks: By working together, SDN controllers and network devices can create a more resilient defense against sophisticated DDoS attacks that may attempt to exploit vulnerabilities or bypass traditional security measures. Collaborative and distributed defense in SDN involves the cooperative efforts of SDN controllers and network devices to share threat intelligence, coordinate responses, and collectively defend against DDoS attacks. This collaborative approach enhances the overall resilience and adaptability of the DDoS mitigation system in SDN environments (Zhang, 2020; Ali *et al*., 2023).

Challenges and Proposed Solutions

Evolving Attack Techniques

Description: DDoS attackers constantly develop new methods to evade detection. Traditional ML models trained on static datasets struggle to adapt. Solutions: Adaptive learning: Implement algorithms that can learn and update themselves based on real-time network traffic. This could include mechanisms like remote education. (Gonzalez and Charfadine, 2023) Focus on anomaly detection: Shift from signature-based detection to anomaly detection. Train models to identify deviations from normal traffic patterns, regardless of the specific attack type (Fernandes *et al*., 2019).

False Positives and Negatives

ML models can misclassify normal traffic as malicious (false positives) or miss actual attacks (false negatives). This disrupts legitimate traffic flow and leaves the network vulnerable.

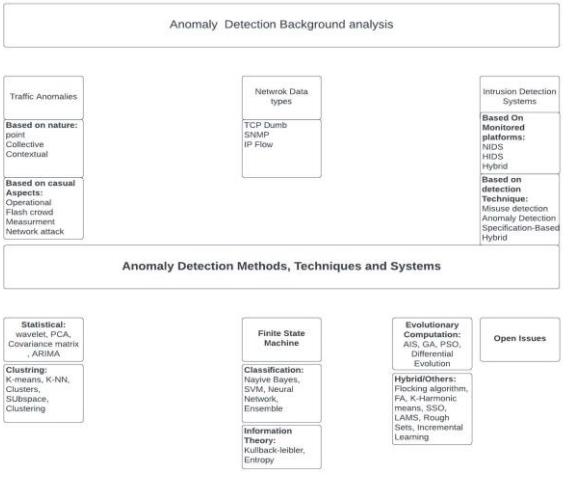

Fig. 6: Anomaly detection summary

Solutions: Feature engineering: Carefully select and engineer relevant network traffic features that effectively distinguish normal from malicious traffic. This work has presented two tree-based approaches to detect anomalies in the presence of irrelevant features. As shown in Fig. (6), Anomaly detection methods are already starting to be used in LHC analyses. Since BDT-based methods are already used in experimental analyses, the hope is that our methods would be readily able to be adopted and calibrated for experimental use. They first considered a CWoLa-inspired method and showed that boosted decision trees are more robust to irrelevant features compared to neural networks. By exploiting the inherent feature selection of decision trees, the BDT-based classifier sustained strong performance despite the inclusion of significantly more irrelevant than discriminating auxiliary features (Freytsis *et al*., 2024). Hybrid detection systems: Combine ML-based detection with other techniques like signature-based methods or honeypots for a more robust approach. They introduced a lightweight flow-based Intrusion Detection System (IDS) that periodically gathers statistical information about flows from SDN OpenFlow forwarding devices and inspects traffic patterns by extracting and aggregating a set of features. The proposed IDS system proved to be accurate with a high detection rate of 0.98 measured by the F1-score of the classification model and a relatively low false alarm rate (Ajaeiya *et al*., 2017).

Scalability and Performance

Description: DDoS attacks can generate massive amounts of traffic overwhelming the SDN controller and switches. Real-time processing of network data for ML models can be computationally expensive. Solution: Distributed Learning: Train ML models in a distributed manner across multiple SDN controllers to handle large datasets efficiently. Hundreds of billions of things are estimated to be deployed in the rapidly advancing IoT paradigm, resulting in enormous amounts of data. Transmitting all these data to the cloud has recently proven to be a performance blockage, as it motivates many network challenges. We deploy Federated Learning Applications in our distributed SDN-based architecture, using the gateways to provide distributed intelligence at the edge of the network and conduct a comprehensive and detailed evaluation of the system from several perspectives (Firouzi and Rahmani, 2022) Resource optimization: Optimize the ML algorithms and network infrastructure to minimize processing overhead without compromising accuracy. The proliferation of mobile devices and the increasing use of networked applications have generated enormous data that require real-time processing and low-latency responses. There is a massive growth of data generated at the web edge, but its limited computing resources and boundary dynamics pose

significant challenges. For proficient resource utilization and network management, Software-Defined Networks (SDN) integration with EC can provide control and programmability in edge networking management. Besides (Satheesh *et al*., 2020), the utilization of SDN in EC addresses both centralized and distributed infrastructure while further facilitating data processing of data nearer to its source and supporting the increasing need for efficient and systematic evaluation of the current systems. The review of challenges, benefits, and various proposed approaches for utilizing capabilities in Edge and SDN-based Edge environments are investigated. This study offers key findings of the present situation and upcoming trends of Edge, SDN-based EC systems, which can assist organizations in designing and deploying more efficient resource consumption solutions (Nain *et al*., 2024).

Data Privacy

Description: Network traffic data, which may contain sensitive information, is necessary for machine learning models to function. Data privacy and security must be balanced. Solution: Data anonymization: Before supplying network traffic data to the machine learning model, anonymize it. Techniques like tokenization and traffic aggregation can be used to accomplish this. Deep Neural Network (NN) based machine learning algorithms have produced impressive results and are widely applied across several areas. The theoretical underpinnings for applying deep neural network algorithms in an encrypted environment are presented in this research, along with strategies for integrating neural networks within realistic constraints of existing homomorphic encryption systems. They demonstrate that using encrypted data to train neural networks, producing encrypted predictions, and returning the suggested Crypto DL show its applicability across numerous datasets. The empirical findings demonstrate that it offers precise training and classification while protecting privacy (Hesamifard *et al*., 2018).

Federated learning: Execute federated learning by training models on local devices while refraining from transmitting raw data to a central server. The proliferation of zero-day exploits has elevated privacy concerns since IoT devices generate and send sensitive data via the conventional internet. This research proposes the utilization of a Deep Neural Network (DNN) and Federated Learning (FL) within an IoT network, alongside Mutual Information (MI) as an efficient anomaly identification technique. A key advantage of combining Federated Learning (FL) with Deep Learning (DL) is that only modified weights are transmitted to the centralized FL server, whereas the data remains on local IoT devices for model training. The evaluation utilizes the IoT-Botnet 2020 dataset. Research indicates that the DNN-based Network Intrusion Detection System (NIDS) outperforms deep learning models, demonstrated by enhanced model accuracy and a reduction in the False Alarm Rate (FAR) (Wang *et al*., 2023).

Conclusion

In this comprehensive study, we conduct a qualitative investigation of the main machine learning techniques used in DDoS mitigation in software-defined networking (SDN) environments. Our research goes beyond this approach and includes identifying the need for recurring problems in the cybersecurity environment. Recognizing the importance of solving these problems, we are trying to propose solutions for everyone in order to reach a useful agreement for continuing efforts to improve DDoS mitigation systems. With this effort, we seek to deepen our understanding of the interplay between machine learning and SDN for DDoS prevention, highlighting the importance of real, flexible, real-time strategies in the face of changing threats.

Acknowledgment

We would like to express our sincere gratitude to Dr. Ahmed Gaber for his invaluable guidance, insightful feedback, and continuous support throughout the development of this research. His expertise in network security and machine learning has greatly contributed to the refinement of our study on mitigating DDoS attacks in Software-Defined Networking (SDN) using machine-learning techniques.

His constructive discussions and encouragement have been instrumental in shaping the direction of this work. We deeply appreciate his dedication and the time he has generously devoted to reviewing our findings and providing valuable recommendations. Thank you, Dr. Ahmed Gaber, for your mentorship and unwavering support.

Funding Information

This research received no specific grant from any funding agency in the public commercial.

Author's Contributions

Ahmed Gaber Abu Abd-Allah: Participated in all experiments, coordinated the data-analysis and designed the research plan and organized the study.

Noureldin Omar Mohamed: Participated in all experiments, coordinated the data-analysis.

Mohamed Samir Mohamed: Coordinated the data-analysis.

Mahytab Jaafar and Youssef Mohamed Yasser: Coordinated the data-analysis.

Mohamed Saeed Taha: Texting and wording.

Ethics

This research does not involve any human participants, personal data, or ethical concerns related to privacy, bias, or harm. The study focuses on the technical aspects of mitigating DDoS attacks in SDN using machine learning. However, ethical considerations arise in the potential misuse of the proposed review. While the intention is to enhance cybersecurity, adversaries could attempt to exploit similar techniques for malicious purposes. Researchers and practitioners should ensure responsible implementation, align with ethical cybersecurity standards, and comply with relevant regulations.

References

Abhiroop, T., Babu, S., & Manoj, B. S. (2018). A Machine Learning Approach for Detecting DoS Attacks in SDN Switches. *2018 Twenty Fourth National Conference on Communications (NCC)*, 1–6. https://doi.org/10.1109/ncc.2018.8600196

Abuarqoub, A. (2020). A Review of the Control Plane Scalability Approaches in Software Defined Networking. *Future Internet*, *12*(3), 49. https://doi.org/10.3390/fi12030049

Ahmad, A., Harjula, E., Ylianttila, M., & Ahmad, I. (2020). Evaluation of Machine Learning Techniques for Security in SDN. *2020 IEEE Globecom Workshops (GC Wkshps)*, 1–6. https://doi.org/10.1109/gcwkshps50303.2020.9367477

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

Ajaeiya, G. A., Adalian, N., Elhajj, I. H., Kayssi, A., & Chehab, A. (2017). Flow-based Intrusion Detection System for SDN. *2017 IEEE Symposium on Computers and Communications (ISCC)*, 787–793. https://doi.org/10.1109/iscc.2017.8024623

Alamri, H. A., & Thayananthan, V. (2020). Bandwidth Control Mechanism and Extreme Gradient Boosting Algorithm for Protecting Software-Defined Networks Against DDoS Attacks. *IEEE Access*, *8*, 194269–194288. https://doi.org/10.1109/access.2020.3033942

Ali, T. E., Chong, Y.-W., & Manickam, S. (2023). *Machine Learning Techniques to Detect a DDoS Attack in SDN: A Systematic Review*. Applied Sciences. https://doi.org/10.3390/app13053183

Anerousis, N., Chemouil, P., Lazar, A. A., Mihai, N., & Weinstein, S. B. (2021). The Origin and Evolution of Open Programmable Networks and SDN. *IEEE Communications Surveys & Tutorials*, *23*(3), 1956–1971. https://doi.org/10.1109/comst.2021.3060582

Badotra, S., & Panda, S. N. (2020). Software-defined networking: A novel approach to networks. *Handbook of Computer Networks and Cyber Security: Principles and Paradigms*, 313–339. https://doi.org/10.1007/978-3-030-22277-2_13

Bendale, S. P., & Rajesh Prasad, J. (2018). Security Threats and Challenges in Future Mobile Wireless Networks. *2018 IEEE Global Conference on Wireless Computing and Networking (GCWCN)*, 146–150. https://doi.org/10.1109/gcwcn.2018.8668635

Caraguay, Á. L. V., Peral, A. B., López, L. I. B., & Villalba, L. J. G. (2016). SDN: Evolution and Opportunities in the Development IoT Applications. *International Journal of Distributed Sensor Networks*, *10*(5), 735142. https://doi.org/10.1155/2014/735142

Costa, L. C., Vieira, A. B., de Britto e Silva, E., Macedo, D. F., Vieira, L. F. M., Vieira, M. A. M., da Rocha Miranda, M., Batista, G. F., Polizer, A. H., Gonçalves, A. V. G. S., Gomes, G., & Correia, L. H. A. (2021). OpenFlow data planes performance evaluation. *Performance Evaluation*, *147*, 102194. https://doi.org/10.1016/j.peva.2021.102194

Cui, Y., Yan, L., Li, S., Xing, H., Pan, W., Zhu, J., & Zheng, X. (2016). SD-Anti-DDoS: Fast and efficient DDoS defense in software-defined networks. *Journal of Network and Computer Applications*, *68*, 65–79. https://doi.org/10.1016/j.jnca.2016.04.005

Dayal, N., & Srivastava, S. (2017). Analyzing behavior of DDoS attacks to identify DDoS detection features in SDN. *2017 9th International Conference on Communication Systems and Networks (COMSNETS)*, 274–281. https://doi.org/10.1109/comsnets.2017.7945387

De Assis, M. V. O., Novaes, M. P., Zerbini, C. B., Carvalho, L. F., Abrao, T., & Proenca, M. L. (2018). Fast Defense System Against Attacks in Software Defined Networks. *IEEE Access*, *6*, 69620–69639. https://doi.org/10.1109/access.2018.2878576

Deepa, V., Sudar, K. M., & Deepalakshmi, P. (2019). Design of Ensemble Learning Methods for DDoS Detection in SDN Environment. *2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*, 1–6. https://doi.org/10.1109/vitecon.2019.8899682

Dong, S., & Sarem, M. (2020). DDoS Attack Detection Method Based on Improved KNN with the Degree of DDoS Attack in Software-Defined Networks. *IEEE Access*, *8*, 5039–5048. https://doi.org/10.1109/access.2019.2963077

Eliyan, L. F., & Di Pietro, R. (2021). DoS and DDoS attacks in Software Defined Networks: A survey of existing solutions and research challenges. *Future Generation Computer Systems*, *122*, 149–171. https://doi.org/10.1016/j.future.2021.03.011

Fernandes, G., Rodrigues, J. J. P. C., Carvalho, L. F., Al-Muhtadi, J. F., & Proença, M. L. (2019). A comprehensive survey on network anomaly detection. *Telecommunication Systems*, *70*(3), 447–489. https://doi.org/10.1007/s11235-018-0475-8

Firdaus, D., Munadi, R., & Purwanto, Y. (2020). DDoS Attack Detection in Software Defined Network using Ensemble K-means++ and Random Forest. *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 164–169. https://doi.org/10.1109/isriti51436.2020.9315521

Firouzi, R., & Rahmani, R. (2022). A Distributed SDN Controller for Distributed IoT. *IEEE Access*, *10*, 42873–42882. https://doi.org/10.1109/access.2022.3168299

Foster, N., Guha, A., Reitblatt, M., Story, A., Freedman, M. J., Katta, N. P., Monsanto, C., Reich, J., Rexford, J., Schlesinger, C., Walker, D., & Harrison, R. (2019). Languages for software-defined networks. *IEEE Communications Magazine*, *51*(2), 128–134. https://doi.org/10.1109/MCOM.2013.6461197

Foster, N., Harrison, R., Freedman, M. J., Monsanto, C., Rexford, J., Story, A., & Walker, D. (2016). Frenetic: a network programming language. *ACM SIGPLAN Notices*, *46*(9), 279–291. https://doi.org/10.1145/2034574.2034812

Freytsis, M., Perelstein, M., & Sana, Y. C. (2024). Anomaly detection in the presence of irrelevant features. *Journal of High Energy Physics*, *2024*(2), 220. https://doi.org/10.1007/JHEP02(2024)220

Gonzalez, C., & Charfadine, S. M. (2023). SDN Controllers and ML-Based Anomaly Detection in Embedded Systems: A Comparative Analysis. *2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, 1–6. https://doi.org/10.1109/wincom59760.2023.10322912

Hamdan, M., Hassan, E., Abdelaziz, A., Elhigazi, A., Mohammed, B., Khan, S., Vasilakos, A. V., & Marsono, M. N. (2021). A comprehensive survey of load balancing techniques in software-defined network. *Journal of Network and Computer Applications*, *174*, 102856. https://doi.org/10.1016/j.jnca.2020.102856

Hannache, O., & Batouche, M. C. (2020). Neural Network-Based Approach for Detection and Mitigation of DDoS Attacks in SDN Environments. *International Journal of Information Security and Privacy*, *14*(3), 50–71. https://doi.org/10.4018/ijisp.2020070104

Hesamifard, E., Takabi, H., Ghasemi, M., & Wright, R. N. (2018). Privacy-preserving Machine Learning as a Service. *Proceedings on Privacy Enhancing Technologies*, *2018*(3), 123–142. https://doi.org/10.1515/popets-2018-0024

Islam, Md. M., Khan, M. T. R., Saad, M. M., & Kim, D. (2021). Software-defined vehicular network (SDVN): A survey on architecture and routing. *Journal of Systems Architecture*, *114*, 101961. https://doi.org/10.1016/j.sysarc.2020.101961

Kokila, R. T., Thamarai Selvi, S., & Govindarajan, K. (2014). DDoS detection and analysis in SDN-based environment using support vector machine classifier. *2014 Sixth International Conference on Advanced Computing (ICoAC)*, 205–210. https://doi.org/10.1109/icoac.2014.7229711

Li, Y., & Louri, A. (2021). ALPHA: A Learning-Enabled High-Performance Network-on-Chip Router Design for Heterogeneous Manycore Architectures. *IEEE Transactions on Sustainable Computing*, *6*(2), 274–288. https://doi.org/10.1109/tsusc.2020.2981340

Liu, Y., Wang, F., Zhao, S., & Tang, Y. (2024). A novel framework for predicting active flow control by combining deep reinforcement learning and masked deep neural network. *Physics of Fluids*, *36*(3). https://doi.org/10.1063/5.0194264

Maheshwari, A., Mehraj, B., Khan, M. S., & Idrisi, M. S. (2022). An optimized weighted voting based ensemble model for DDoS attack detection and mitigation in SDN environment. *Microprocessors and Microsystems*, *89*, 104412. https://doi.org/10.1016/j.micpro.2021.104412

Monsanto, C., Foster, N., Harrison, R., & Walker, D. (2017). A compiler and run-time system for network programming languages. *ACM SIGPLAN Notices*, *47*(1), 217–230. https://doi.org/10.1145/2103621.210368

Musumeci, F., Fidanci, A. C., Paolucci, F., Cugini, F., & Tornatore, M. (2022). Machine-Learning-Enabled DDoS Attacks Detection in P4 Programmable Networks. *Journal of Network and Systems Management*, *30*(1), 21. https://doi.org/10.1007/s10922-021-09633-5

Myint Oo, M., Kamolphiwong, S., Kamolphiwong, T., & Vasupongayya, S. (2019). Advanced Support Vector Machine- (ASVM-) Based Detection for Distributed Denial of Service (DDoS) Attack on Software Defined Networking (SDN). *Journal of Computer Networks and Communications*, *2019*, 1–12. https://doi.org/10.1155/2019/8012568

Nadeem, M. W., Goh, H. G., Ponnusamy, V., & Aun, Y. (2022). DDoS Detection in SDN using Machine Learning Techniques. *Computational Materials and Continua*, *71*(1), 771–789. https://doi.org/10.32604/cmc.2022.021669

Nain, A., Sheikh, S., Shahid, M., & Malik, R. (2024). Resource optimization in edge and SDN-based edge computing: a comprehensive study. *Cluster Computing*, *27*(5), 5517–5545. https://doi.org/10.1007/s10586-023-04256-8

Nurwarsito, H., & Nadhif, M. F. (2021). DDoS Attack Early Detection and Mitigation System on SDN using Random Forest Algorithm and Ryu Framework. *2021 8th International Conference on Computer and Communication Engineering (ICCCE)*, 178–183. https://doi.org/10.1109/iccce50029.2021.9467167

Oo, M. M., Kamolphiwong, S., Kamolphiwong, T., & Vasupongayya, S. (2020). Analysis of Features Dataset for DDoS Detection by using ASVM Method on Software Defined Networking. *International Journal of Networked and Distributed Computing*, *8*(2), 86–93. https://doi.org/10.2991/ijndc.k.200325.001

Perez-Diaz, J. A., Valdovinos, I. A., Choo, K.-K. R., & Zhu, D. (2020). A Flexible SDN-Based Architecture for Identifying and Mitigating Low-Rate DDoS Attacks Using Machine Learning. *IEEE Access*, *8*, 155859–155872. https://doi.org/10.1109/access.2020.3019330

Phan, T. V., Bao, N. K., & Park, M. (2016). A Novel Hybrid Flow-Based Handler with DDoS Attacks in Software-Defined Networking. *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, 350–357. https://doi.org/10.1109/uic-atc-scalcom-cbdcom-iop-smartworld.2016.0069

Phan, T. V., Bao, N. K., & Park, M. (2017). Distributed-SOM: A novel performance bottleneck handler for large-sized software-defined networks under flooding attacks. *Journal of Network and Computer Applications*, *91*, 14–25. https://doi.org/10.1016/j.jnca.2017.04.016

Polat, H., Polat, O., & Cetin, A. (2020). Detecting DDoS Attacks in Software-Defined Networks Through Feature Selection Methods and Machine Learning Models. *Sustainability*, *12*(3), 1035. https://doi.org/10.3390/su12031035

Qin, M. (2021). A survey on SDN network programming languages. *IEEE Xplore*.

Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). *Zero Trust Architecture*. https://doi.org/10.6028/NIST.SP.800-207

Sahoo, K. S., Iqbal, A., Maiti, P., & Sahoo, B. (2018). A Machine Learning Approach for Predicting DDoS Traffic in Software Defined Networks. *2018 International Conference on Information Technology (ICIT)*, 199–203. https://doi.org/10.1109/icit.2018.00049

Sahoo, K. S., Tripathy, B. K., Naik, K., Ramasubbareddy, S., Balusamy, B., Khari, M., & Burgos, D. (2020). An Evolutionary SVM Model for DDOS Attack Detection in Software Defined Networks. *IEEE Access*, *8*, 132502–132513. https://doi.org/10.1109/access.2020.3009733

Sangodoyin, A. O., Akinsolu, M. O., Pillai, P., & Grout, V. (2021). Detection and Classification of DDoS Flooding Attacks on Software-Defined Networks: A Case Study for the Application of Machine Learning. *IEEE Access*, *9*, 122495–122508. https://doi.org/10.1109/access.2021.3109490

Santos, R., Souza, D., Santo, W., Ribeiro, A., & Moreno, E. (2020). Machine learning algorithms to detect DDoS attacks in SDN. *Concurrency and Computation: Practice and Experience*, *32*(16), e5402. https://doi.org/10.1002/cpe.5402

Satheesh, N., Rathnamma, M. V., Rajeshkumar, G., Sagar, P. V., Dadheech, P., Dogiwal, S. R., Velayutham, P., & Sengan, S. (2020). Flow-based anomaly intrusion detection using machine-learning model with software defined networking for OpenFlow network. *Microprocessors and Microsystems*, *79*, 103285. https://doi.org/10.1016/j.micpro.2020.103285

Scaranti, G. F., Carvalho, L. F., Barbon, S., & Proenca, M. L. (2020). Artificial Immune Systems and Fuzzy Logic to Detect Flooding Attacks in Software-Defined Networks. *IEEE Access*, *8*, 100172–100184. https://doi.org/10.1109/access.2020.2997939

Shaghaghi, A., Kaafar, M. A., Buyya, R., & Jha, S. (2020). Software-Defined Network (SDN) Data Plane Security: Issues, Solutions and Future Directions. *Handbook of Computer Networks and Cyber Security: Principles and Paradigms*, 341–387. https://doi.org/10.1007/978-3-030-22277-2_14

Swami, R., Dave, M., & Ranga, V. (2021). Detection and Analysis of TCP-SYN DDoS Attack in Software-Defined Networking. *Wireless Personal Communications*, *118*(4), 2295–2317. https://doi.org/10.1007/s11277-021-08127-6

Tan, L., Pan, Y., Wu, J., Zhou, J., Jiang, H., & Deng, Y. (2020). *A New Framework for DDoS Attack Detection and Defense in SDN Environment*. IEEE Access. https://doi.org/10.1109/access.2020.3021435

Tonkal, Ö., Polat, H., Başaran, E., Cömert, Z., & Kocaoğlu, R. (2021). Machine Learning Approach Equipped with Neighbourhood Component Analysis for DDoS Attack Detection in Software-Defined Networking. *Electronics*, *10*(11), 1227. https://doi.org/10.3390/electronics10111227

Voellmy, A., & Wang, J. (2012). Scalable Software Defined Network Controllers. *ACM SIGCOMM Computer Communication Review*, *3*, 289–290.

Voellmy, A., Kim, H., & Feamster, N. (2012). Procera: a language for high-level reactive network control. *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, 43–48. https://doi.org/10.1145/2342441.2342451

von Rechenberg, M., Rettore, P. H. L., Lopes, R. R. F., & Sevenich, P. (2021). Software-Defined Networking Applied in Tactical Networks: Problems, Solutions and Open Issues. *2021 International Conference on Military Communication and Information Systems (ICMCIS)*, 1–8. https://doi.org/10.1109/ICMCIS52405.2021.9486399

Wang, X., Wang, Y., Javaheri, Z., Almutairi, L., Moghadamnejad, N., & Younes, O. S. (2023). Federated deep learning for anomaly detection in the internet of things. *Computers and Electrical Engineering*, *108*, 108651. https://doi.org/10.1016/j.compeleceng.2023.108651

Sellami, A. (2022). Energy-aware task scheduling and offloading using deep reinforcement learning in SDN enabled IoT network. Elsevier

Xu, Y., Sun, H., Xiang, F., & Sun, Z. (2019). Efficient DDoS Detection Based on K-FKNN in Software Defined Networks. *IEEE Access*, *7*, 160536–160545. https://doi.org/10.1109/access.2019.2950945

Yang, L. L., Dantu, R. anderson, T. A., & Gopal, R. (2004). *Forwarding and Control Element Separation (ForCES) Framework*.

Zhang, C. (2020). Design and application of fog computing and Internet of Things service platform for smart city. *Future Generation Computer Systems*, *112*, 630–640. https://doi.org/10.1016/j.future.2020.06.016