Uluslararası İleri Doğa Bilimleri ve Mühendislik Araştırmaları Dergisi
Sayı 7, S. 327-349, 4, 2023

Araştırma Makalesi

IJANSER

https://as-proceeding.com/index.php/ijanser
ISSN: 2980-0811

International Journal of Advanced Natural Sciences and Engineering Researches
Volume 7, pp. 327-349, 4, 2023

Research Article

EXPLORING THE LANDSCAPE OF SDN-BASED DDOS DEFENSE: A HOLISTIC EXAMINATION OF DETECTION AND MITIGATION APPROACHES, RESEARCH GAPS AND PROMISING AVENUES FOR FUTURE EXPLORATION

Tasnim ALASALI[1] and Omar DAKKAK[1]

[1]*Department of Computer Engineering, Karabük Üniversitesi, 78050 Karabük, Türkiye*

2028150011@ogrenci.karabuk.edu.tr

(Received: 17 April 2023, Accepted: 22 May 2023)

(DOI: 10.59287/ijanser.726)

(1st International Conference on Contemporary Academic Research ICCAR 2023, May 17-19, 2023)

ATIF/REFERENCE: Alasali, T. & Dakkak, O. (2023). Exploring The Landscape Of Sdn-Based Ddos Defense: A Holistic Examination Of Detection And Mitigation Approaches, Research Gaps And Promising Avenues For Future Exploration. *International Journal of Advanced Natural Sciences and Engineering Researches*, 7(4), 327-349.

Abstract – Over the course of time, a multitude of security solutions have been proposed in order to safeguard the Internet architecture from an extensive array of malware threats. However, the task of ensuring the security of the Internet and its associated applications remains an ongoing research challenge. Researchers persistently delve into the exploration of innovative network architectures, such as the utilization of HTTP as the narrow waist, the implementation of Named Data Networking (NDN), the development of programmable networks, and the adoption of Software-Defined Networking (SDN), with the aim of designing a more dependable and resilient network infrastructure. Among these alternative approaches, SDN has emerged as a robust and secure solution for countering malicious activities. By separating the control plane from the data plane, SDN provides an array of advantages, including enhanced manageability, improved control, dynamic rule updates, advanced analysis capabilities, and a comprehensive network overview facilitated by a centralized controller. Despite its superiority over conventional IP-based networks, SDN is susceptible to various network intrusions and encounters significant challenges in terms of deployment. The purpose of this paper is to conduct a comprehensive review of approximately 70 prominent mechanisms employed for the detection and mitigation of Distributed Denial of Service (DDoS) attacks in SDN networks. These mechanisms are systematically categorized into four main groups, namely information theory-based methods, machine learning-based methods, approaches based on Artificial Neural Networks (ANN), and other miscellaneous methods. Furthermore, the paper identifies and discusses several unresolved research issues and gaps that exist in the deployment of a secure DDoS defense solution within SDN networks. The objective of this comprehensive

review is to provide valuable insights to the research community, assisting in the development of more robust and reliable DDoS mitigation solutions that are specifically tailored for SDN networks.

Keywords – Software Defined Networks (SDN), DDOS, Mitigation, SDN Security, Control Plane, Detection.

1. INTRODUCTION:

In the past twenty years, there has been a significant expansion in the usage of internet-based services and applications, which has led to a sizeable global user base of around 57 percent [1]. In direct proportion, there has been an increase in the level of worry regarding the safety of the internet. Since its inception, the Internet has been susceptible to a wide variety of security flaws, such as worms, port scans, distributed denial of service attacks, and Trojan horses [1]. The research community has focused a large amount of emphasis on one of these dangers, which is known as a denial of service assault (DoS). Attacks that use the denial of service technique entail obstructing the access of authorized users to certain network resources on purpose. A notable and unforgettable event took place on February 7, 2000, when a hacker known as Mafiaboy, who was just 15 years old at the time, staged a series of distributed denial of service assaults on large online retailers such as Amazon and eBay [2]. Since then, DoS attack tactics have advanced, with attackers employing geographically spread devices to launch DDoS attacks. DDoS stands for "distributed denial of service." Intruders use network vulnerabilities as a means to carry out this type of assault, which is generally known as a Trojan Horse attack [3], since it involves the covert installation of a harmful program into computers whose users are unaware that the program is present. The invader creates a network of compromised systems, often known as a botnet, by spreading this program over several networked devices, which is how the botnet gets its common name [3]. An individual human operator, sometimes known as a bot master, controls the entirety of the botnet through remote control [3–5]. An adversary can start a distributed denial of service attack by issuing commands to all compromised devices. These commands tell the compromised machines to generate and send an overwhelming amount of meaningless network traffic towards the adversary's desired target. The magnitude of the infected devices, which can reach millions, causes the victim's resources to become overwhelmed, which results in the victim's resources becoming inaccessible to legitimate users and culminates in a DDoS attack.

The rapidly accelerating development of IT infrastructure has resulted in a significant increase in the size and complexity of network configurations. As a direct result of this, ensuring essential network features such as integrity, confidentiality, authentication, information availability, and non-repudiation has become a far more difficult task [13]. As a direct result of this, both academic researchers and private industry have redirected their attention to the development of network architectures that are more robust, scalable, and secure [14]. In contrast to the static and decentralized nature of traditional networks, recent developments, most notably SDN, have emerged as a significant step towards the establishment of a dynamic and centralized network environment. This is in contrast to the nature of traditional networks. Despite this, the already deployed networks are still extremely complicated and difficult to administer [15]. When attempting to apply high-level network regulations in typical IP-based networks, administrators of the network run into challenges since they are forced to configure each network device using instructions that are unique to the particular vendor of the equipment [16]. As a consequence of this, IP-based networks present a number of important issues when it comes to the implementation of desired procedures and the reconfiguration of network devices [1]. Furthermore, these networks exhibit a tight coupling between the control plane, which is responsible for network traffic handling, and the data plane, which forwards traffic based on decisions made by the control plane, thereby limiting flexibility and impeding innovation in network infrastructure [1]. The control plane is responsible for handling

network traffic, and the data plane forwards traffic based on decisions made by the control plane.

Scaling network infrastructures to meet the demands of computer networks is necessary in order to solve the problems that have been identified. Nevertheless, the overall complexity of the network will be increased even further as a result of this development [17]. In recent years, both the scientific community and the sector that deals with networks have come up with a number of different ideas in an effort to create better future networks [18]. Included in these solutions are the hypertext transfer protocol (HTTP) as the thin waist [19], named data networking (NDN) [20], programmable networks [21], and SDN [22]. SDN has emerged as a promising technique to tackle the challenges in contemporary networks, making it one of these potential answers to the problem. The software-defined network (SDN) is an ever-evolving network design that holds out hope for a more effective network infrastructure. It is able to accomplish this by divorcing the control plane, which is comprised of the network's control logic, from the data plane, which is comprised of the underlying routers and switches that are responsible for routing network traffic in accordance with the control logic. As a result of this decoupling, the control logic can be installed in logically centralized controllers, while the network switches continue to perform their traditional roles as straightforward packet forwarding devices. A split of this kind increases flexibility, implementation speed, and programmability while also making network management more straightforward [22].

Despite the fact that it has been shown to be able to improve network security through the use of centralized controllers, global network visibility, and the creation of traffic forwarding rules on demand [23], the SDN architecture is still faced with a number of challenges and concerns, including network security, scalability, and supportability. The failure of the centralized controller can cause disruptions throughout the entire network, making security one of the most important concerns among these problems. Both the centralized control and the communication between the controller and the switches are susceptible to sophisticated Distributed Denial-of-Service (DDoS) assaults [24], [25]. These attacks have been demonstrated to have a considerable influence on the performance of SDN networks [24], [25].

The network is separated into the data plane, the control plane, and the application plane by the SDN architecture. Because of this, DDoS assaults in a software-defined network (SDN) can be separated into application-layer, control-layer, and data-layer DDoS attacks, depending on the plane that is being targeted [8]. There is a significant body of research on DDoS protection solutions [6–12, 23–28], which is very closely related to the work that we have been doing. However, only a small number of authors have concentrated their efforts on detecting and mitigating DDoS assaults explicitly in the context of SDN Several writers, including Bawany et al. [6], Joelle et al. [7], Dong et al. [8], Fajar et al. [9], Xu et al. [10], Kalkan et al. [11], and Singh et al. [12], have made attempts to review strategies for detecting and mitigating distributed denial of service assaults (DDoS). Nevertheless, in comparison to these previously published evaluations, the review that we present in this study is more in-depth and contains more specific technical information. In addition, the study that we have conducted is multi-faceted, as can be seen in Table 1, which is where we highlight the research gaps that have not been addressed in these review publications.

The following is a list of the primary contributions that our paper makes:

Review in Depth This article provides a comprehensive review of the many different kinds of DDoS assaults that can be initiated within the context of SDN.

A Detailed Investigation Into: We give a thorough analysis of the current state of the art regarding methods for detecting and mitigating distributed denial of service attacks in SDN networks. In addition to this, we classify these protective measures into four distinct groups and include

comprehensive technical information regarding the aforementioned mechanisms.

Identification of Void Areas in Research: After conducting an exhaustive study of the DDoS-based solutions in SDN, we have compiled a list of the research gaps that exist in the various SDN-based DDoS solutions now in use.

The remaining content of the document is organized as follows throughout the subsequent sections of this study: In the following section, we will present an in-depth analysis of the various DDoS detection and mitigation strategies currently in use. In Section 5, a summary of the most significant research voids in the body of previous work is presented. Our investigation is brought to a close in Section 6, which focuses on potential future directions.

2. DISTRIBUTED DENIAL OF SERVICE TYPES IN SDN:

The SDN architecture represents a pioneering concept aimed at enhancing network manageability and security by decoupling the control plane from the data plane [49]. This architectural approach possesses inherent characteristics such as the separation of the control plane from the data plane, a comprehensive network view, network programmability, software-based traffic analysis, and dynamic updating of network policies, all with the goal of improving network security.

However, despite these security enhancements, the SDN architecture remains vulnerable to attacks targeting the data plane, control plane, and the interfaces between planes. These attacks present significant challenges and compromise the integrity of the SDN architecture, rendering it a complex and demanding network architecture [90]. Figure 5 provides an overview of the vulnerability of specific planes to different types of attacks within the SDN architecture. The following attacks are examined in detail:

- **Packet_in Flooding**: The attacker will carry out this attack by delivering a large number of packets to the virtual switch (vswitch) that contain faked IP addresses. This will force the vswitch to send a large number of packet in messages to the centralized controller through the southbound interface. The controller is flooded with manufactured flow requests, which causes it to become unreachable to users who are actually authorized to use it [91].

- **Spoofing of Switch:** During this type of attack, the perpetrator makes a fictitious change to the IP address of a switch and then uses that altered address to create a connection with the controller. The attacker concurrently activates a second malicious switch with a faked IP address, which then establishes a connection with the controller while the first switch, which is legitimate, is in the process of communicating with the controller. As a consequence of this, the controller severing its connection with the honest switch in order to engage in activity with the dishonest switch, which in turn causes the performance of the network to suffer. This attack results in bogus requests being sent to the controller, which makes the implementation of mitigating methods [92] necessary.

- **Flow Table Overflow:** In the design of a SDN, the flow rules that are sent by the controller are saved in the flow table of an OpenFlow switch. Old flow rules are removed from the flow table by the switch when a predetermined amount of time has elapsed, as determined by the timeout value associated with each flow rule [93]. However, there is a problem caused by the restricted capacity of the Ternary Content Addressable Memory (TCAM) that is utilized for storing flow table entries [89]. TCAM is both expensive and power-intensive. An adversary can overwhelm the

switch by flooding it with a large number of newly generated flows if they take advantage of this weakness and exploit it. As a consequence of this, the switch's flow table memory is exhausted very quickly, which results in the deletion of valid entries and a decrease in overall network performance.

- **Congestion of Southbound APIs**: In the SDN architecture, when an OpenFlow switch forwards a packet_in request to the controller, it transmits a portion of the packet while storing the remaining part in its buffer. However, if the buffer becomes saturated, the switch must forward the entire packet to the controller [88]. By dispatching multiple fabricated flows to the switch, an attacker can easily overload the limited bandwidth employed in the southbound APIs, causing congestion that renders it unavailable to legitimate users [93].

- **Controller Saturation**: In a network that uses SDN, the controller is responsible for handling multiple packet in requests, including those that are brought about by bogus flows coming from an attacker. The controller will create queues so that these requests can be processed in order. However, when there are many fake packets present, the controller gets obsessed with processing false requests. This causes a decline in performance and is a substantial challenge for SDN-based networks [94]. Attackers are able to generate a sufficient volume of fabricated flows, which can exhaust the controller's processing capabilities if they are successful. Even if supplementary controllers have been suggested as a potential solution [25], even the secondary controller can still be compromised by these kinds of attacks. As mentioned in [76], utilizing numerous controllers as a defense mechanism against DoS and DDoS attacks is not a workable method since it can result in a failure that cascades across all controllers. This renders the solution infeasible.

- **Buffer Saturation**: When a switch delivers a packet in message to the controller, it sends a portion of the packet to the controller and stores the rest part of the packet in its buffer memory. The controller receives the piece of the packet that the switch sends to it. An attacker can initiate an attack by flooding the switch with many manufactured packets, which quickly depletes the buffer. This functionality can be exploited in order to accomplish this. Once the switch has used up all of the available buffering space on its internal bus, it is required to send the complete packet to the controllers as an event, which results in yet another bottleneck for the SDN architecture. This stage makes it difficult for legitimate users to process their flow requests, which gives the attacker the opportunity to slow down the network.

- **Unauthorized Applications**: Within SDN, multiple applications that access network resources in order to deliver services to the controller and the network are grouped together under the rubric of the application plane. Certain apps have the capability of gaining access to network resources by utilizing instances of other applications. On the other hand, because applications do not have authentication and authorisation procedures, it is possible for malicious applications to acquire illegal access through instances of other applications. These malicious programs have the ability to control the behavior of the network and cause a performance decrease in the network.

Thus, it is evident that despite the innovative security features integrated into the SDN architecture, various attacks targeting different planes within the architecture can undermine its

robustness and effectiveness, thereby posing significant challenges to its implementation and operation.

3. REVIEW OF DDOS DEFENCE SOLUTIONS IN SDN

DDoS attacks have become a significant concern for researchers in recent years. While SDN presents certain features that can address the problem of DDoS attacks, SDN itself has become a target for attackers. Consequently, numerous researchers have proposed mechanisms for detecting and mitigating DDoS attacks in the SDN context. This section provides a systematic analysis of the recent work conducted in DDoS attack detection and mitigation within SDN.

4.1. Review of DDoS detection solutions in SDN

This section provides an in-depth analysis of the various DDoS detection solutions that are currently available in SDN. These remedies are split up into four distinct classes according to the kind of detection measure and detection technique that they make use of. DDoS security solutions based on information theory, machine learning-based DDoS defense solutions, artificial neural network-based defense solutions, and other types are included in the categories.

4.1.1. Information theory-based DDoS defense solutions in SDN

Metrics based on information theory, such as entropy and divergence, have seen widespread application in the field of DDoS attack detection [95]. While entropy quantifies the degree of similarity between two probability distributions, divergence metrics assess the degree to which two probability distributions are alike. Claude Shannon was the first person to publicly discuss the idea of quantifying uncertainty in 1948 [95]. Divergence metrics, which evaluate the information distance between various probability distributions of traffic flows, are used to discover irregularities in network traffic. These abnormalities can be caused by a number of different things. Utilizing entropy measurement allows for the observation of current network behavior that deviates from typical behavior, which enables the identification of DDoS assaults. Several groups of academics have come up with a variety of DDoS defense strategies that make use of entropy metrics.

The programmability of SDN networks enables the collection and analysis of network traffic statistics. SDN refers to a type of networking that is defined by software. This feature was expanded by Giotis et al. [97], who used information extraction to make the controller's job easier, hence minimizing the amount of work it had to do. Their strategy entails performing regular data collecting and analysis by employing the entropy method in order to identify network irregularities. The collector module is responsible for gathering data and transmitting it to the anomaly detection module. The anomaly detection module then examines all flow entries for each time window in order to determine whether flows are malicious. After malicious flows have been identified, the mitigation module's flow rules can be used to prevent such flows. Using the network of the National Technical University of Athens, benign data was gathered, while malicious traffic was created with tools such as Tcpreplay and Scapy to validate the effectiveness of their approach..

Wang et al. [98] presented a method that decreases the overhead of flow collecting in the controller by implementing the scheme on OpenFlow switches. This method reduces the amount of work that has to be done to collect flows. Their approach focuses on flow statistics and makes use of entropy in order to identify anomalies in network behavior. The Mininet network simulator was used for the validation of their methodology. A approach that is both effective and lightweight was presented by Mousavi et al. [99] for the purpose of attack detection in SDN. Atypical behavior is identified by their system through the utilization of the entropy fluctuation of the destination IP address. Monitoring the destination IP of incoming packets and measuring the packet count from the same IP are the

two primary methods by which the system identifies attacks on the controller. When multiple packets with the same destination are recorded, the entropy of the network decreases, which enables the identification of irregularities within the system. The experimental setup made use of the Mininet network emulator, and the Scapy program was used to generate the test traffic.

Botie et al. [100] proposed a novel method that makes use of the processing capabilities that are already there in the switch to detect and stop DDoS attacks. A network anomaly can be identified through the monitoring of traffic characteristics and the use of an algorithm based on entropy. As soon as a violation is found, new flow rules are placed on switches to prevent malicious flows from being transmitted. Within the framework of their defense strategy, Tsai et al. [101] implement an entropy-based DDoS detection approach. The entropy is determined by an application that is running on the controller, and it watches and calculates the incoming traffic. In the event that an attack is identified, the application contributes to the process of putting a mitigation strategy into action by closing the relevant port. In addition, this information is relayed to the Network Intrusion and Security Division (NISD) so that it can undergo additional analysis and perhaps block harmful activity. Their methodology is proven in an experimental setting by utilizing the Ryu controller, and the Scapy tool is used for the production of traffic in the experimental setting.

In their paper [102], Kalkan et al. introduced an innovative security strategy for SDN networks that makes use of joint entropy. Their proposed approach, which they call JESS (Joint Entropy-based Security Scheme), is broken down into three stages: nominal, preparatory, and active mitigation. During the phase in which there is no threat of an attack, baseline information is compiled by developing nominal pair profiles for each attribute pair, and all traffic is routed through the controller. The preliminary stage starts as soon as congestion is identified, and during this stage the switch merely delivers information about packets to the controller. The controller will determine the joint entropies of pair profiles, and it will determine whether or not a DDoS attack has occurred based on whether or not the difference between the entropies exceeds a certain threshold. When an attack is discovered, the controller notifies the switch, and the Attack Mitigation module immediately begins dropping attack packets while safeguarding genuine packets. This keeps the network secure.

Metrics based on Generalized Entropy (GE) and Generalized Information distance (GID) were introduced by Sahoo et al. [103] in order to detect low-rate DDoS attacks on the control layer of a software-defined network (SDN). They used information distance as a metric in order to quantify the difference in the traffic flows throughout the network. Their strategy, which made use of GE and GID, produced superior results to those obtained using Shannon entropy and KL-divergence. The method was tested by running it on the Mininet emulator while using the Linux Ubuntu 14.04 LTS operating system.

Jiang et al. [105] presented EDDM, which stands for Entropy-based DDoS Defence Mechanism. This is a novel approach to detecting and mitigating DDoS attacks on the SDN controller. The three phases that make up EDDM are as follows: the Window Construction Phase, the DDoS Detection Phase, and the DDoS Mitigation Phase. The entropy metric is applied to analyze network traffic in order to differentiate between legitimate and malicious activity. This method can identify bots by using the corresponding In-port of the In-switch to prevent an attack from occurring. In order to verify the effectiveness of their strategy, the floodlight controller was built on Mininet.

A technique that utilizes a dynamic threshold was proposed by Hong et al. [106], and it is designed to balance the load in both attack and non-attack scenarios. The entropy of the network's features is computed using this approach, and the threshold is determined in a dynamic manner. The load balancer will find other routes to disperse the network traffic and prevent congestion at a given switch when it learns about a higher rate of network traffic. This

happens when the load balancer obtains knowledge about the higher rate of network traffic.

An adaptive framework for detecting and mitigating DDoS attacks was introduced by Bawany et al. [107] under the name SEAL (SEcure and AgiLe). This framework is comprised of three modules: d-defence, a-defence, and c-defence. The adaptability of SEAL is achieved by using a modified form of EWAA filters (estimated weighted moving average). Other applications call for the utilization of various filters, including proactive, active, and passive varieties. The Mininet network simulator was utilized in order to do the validation of their methodology.

Using the entropy metric, Ahalawat et al. [108] presented yet another way for identifying and mitigating DDoS assaults. After the mitigation module has identified an attack, it will restrict the input of switches to lessen the damage caused by the attack. Additionally, they tested their strategy by utilizing Mininet in conjunction with the Ryu controller.

Conditional entropy was used for detection by Xuanyua et al. [109], and wildcard policy was used to get rid of unwanted packets and prevent attacks. On the other hand, it was discovered that the wildcard policy was not adequate for separating malicious traffic from regular traffic in a reasonable manner.

Cognitively-inspired computing combined with dual address entropy was the basis for Cui et al[110] .'s solution to the problem of detecting distributed denial of service attacks. The statistic collection module will, on a regular basis, gather data and do calculations to determine the frequency of each source and destination IP. The Feature Computing module is responsible for calculating the entropy of both the source address and the destination. When a DDoS attack is identified, it is because the entropy of the source is higher than the regular traffic threshold, and the entropy of the destination is lower than the threshold. To prevent further attack from being done by the attack, the DDoS defender throws away all of the table entries. A floodlight controller that was connected to Mininet was utilized in the process of validating their technique.

These experiments, taken as a whole, show that entropy-based metrics are beneficial for detecting and mitigating DDoS attacks in SDN networks. The many different strategies that have been suggested make use of the programmability and adaptability offered by SDN architectures in order to strengthen network security and guarantee reliable operation.

4.1.2. Machine learning-based DDoS defense solutions in SDN

It has been established that machine learning (ML) algorithms are effective in tackling difficult problems in a variety of contexts [112], thanks to the widespread and successful implementation of these algorithms. At the area of network security, machine learning algorithms have been shown to be extremely effective in identifying DDoS assaults, to the point where they outperform traditional signature-based detection techniques [113]. These methods make it possible to accurately identify aberrant network traffic behavior by using classifiers that have been trained on that data. The Support Vector Machine (SVM), the Hidden Markov Model (HMM), the Decision Tree (J48), the Advanced Support Vector Machine (SVM), Naive Bayes, Logistic regression, Random Trees, the Binary Bat algorithm, Random Forest, and K-nearest neighbor (KNN) are examples of popular ML-based classifiers [113].

As can be seen from the papers that are compiled in Table 6, researchers have been able to improve DDoS attack detection by making use of strong machine learning techniques. A multi-vector DDoS detection system that runs as a network application on an SDN controller was proposed by Niyaz et al. [114]. They gathered normal data from a Home Wireless Network that was connected to the Internet and then created attack traffic using Hping within a VMware ESXi host environment. The attack traffic included a variety of DDoS attacks.

Hidden Markov Models were applied in the Hurley et al. [115] intrusion detection system for the SDN environment (HMM). They trained Hidden Markov

Models utilizing the Baum-Welch technique after establishing an experimental setup with the Mininet emulator and the Floodlight OpenFlow controller.

Alshamrani et al. [116] proposed a solution to deal with attacks known as Misbehavior and NewFlow. Their system collected information about the network at regular intervals and used a technique called machine learning to identify network flows as either normal or attack flows. Emulation of a network was used to verify the viability of the strategy.

Entropy and a support vector machine classifier were the foundation of the flooding attack detection and mitigation system that Hu et al. [117] demonstrated. They did this by collecting data on network traffic using the SDN controller and sFlow agents and then implementing a mitigation agent to prevent attack traffic while still allowing legitimate users to have normal access to network resources. The Mininet emulator was utilized in order to carry out the validation of their methodology.

A hybrid DDoS attack detection system that incorporated statistical and machine learning techniques was presented by Dehkordi et al. [118]. They used an analytical approach for the feature extraction process, and for the classification step, they used a machine learning approach. The validity of the suggested method was checked by using the UNB-ISCX dataset, as well as the CTU-13 and ISOT data sets.

In their study [119], Li et al. presented a two-stage IDS (Intrusion Detection System) that could detect network anomalies intelligently by recording network flows from a global perspective. They used a voting mechanism and implemented the Bat method with Swarm Division and Binary Differential Mutation for feature extraction. Additionally, they utilized Random Forest as a classifier with customizable weights for sample data. Adaptability was used to validate the approach by changing the significance of samples through a vote system.

Another hybrid method was presented by Guozi et al. [120], which made use of the KNN machine learning technology in addition to the entropy method. KNN was used for the classification of network traffic, whereas -entropy was utilized for the process of picking particular features from within a feature set.

Deepa et al. [121] proposed an ensemble technique for detecting aberrant behavior in network traffic within the SDN controller. This method makes use of multiple different types of data. They used KNN, Naive Bayes, support vector machines, and self-organizing maps to gain increased productivity by combining these methods. The method was validated by utilizing the Mininet emulator, and the authors discovered that SVM-SOM produced a greater detection rate and accuracy compared to other combinations. This was in comparison to the other combinations.

Phan et al. [122] presented a unique DDoS attack defender that improved the detection rate and speed for traffic classification by employing a hybrid machine learning technique and an enhanced History-based IP Filtering (eHIPF) scheme, in place of HIPF. This was done to replace HIPF. When an attack is detected by eHIPF, the mitigation agent will send a flow mod message to the cloud's border that includes a drop action. This will cause every packet to be discarded. An experimental setting was rigged up in a research facility's network in order to verify the viability of the proposed methodology.

Myint et al. [123] developed a solution that uses a sophisticated support vector machine to detect DDoS attacks with a small amount of additional processing load. In order to cut down on the amount of time spent training and testing, volumetric and asymmetric features were utilized. This strategy was successful in identifying two different types of flooding assaults, namely ICMP flood and UDP flood. In order to verify the viability of the proposed technique, the OpenDaylight controller was built and deployed on Mininet.

These studies provide evidence of the varied applications of machine learning algorithms in

efficiently identifying and mitigating distributed denial of service attacks (DDoS), highlighting the potential of these methods to improve network security in an SDN context.

.4.1.3. Artificial Neural Network-Based DDoS Defense Solutions In SDN

Researchers have shown a significant amount of interest in the application of Artificial Neural Networks (ANN) due to the inherent benefits offered by these networks, which include better fault tolerance and robustness, self-organization, parallelism, and self-learning capabilities. Because of these characteristics, ANN is a potential solution for DDoS attack detection [124], since it has the ability to recognize both known and unexpected attack patterns. The intelligence and flexibility of intrusion detection systems (IDS) can be improved through the use of artificial neural network (ANN) approaches. Researchers have made use of a number of powerful algorithms, such as Self-Organizing Maps (SOM), exact-STORM, Back Propagation Neural Networks (BPNN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM), in order to detect DDoS attacks in software-defined networking-based networks. These efficient algorithms are summarized in Table 7, and further explanation is provided below.

Braga et al. [125] developed a lightweight technique to the detection of DDoS attacks that minimizes the attack on performance overhead. Their technique took into consideration many aspects of traffic flow and made use of flow-based information. The system kept an eye on the NOX switches and gathered data from the flow entries of all switches to determine things like the average duration of each flow, the percentage of flows that were pair flows, the growth of single flows, and the growth of different ports. The classification of traffic as either normal or an attack was accomplished by the classifier module by using the precise position of the neuron that emerged victorious in the topological map. For the flooding attacks, a synthetic dataset that was built with the help of the Stacheldraht tool was substituted for legitimate traffic that was collected from real datasets.

The SD-Anti DDoS defense mechanism was developed by Cui et al. [126] for use in SDN as a means of protecting against DDoS attacks. The strategy that was suggested included the following four modules: an Attack Detection Trigger, an Attack Detection, an Attack Traceback, and an Attack Mitigation. The Attack Detection Trigger module was developed to give a quicker response time against DDoS attacks, hence minimizing the amount of work that needs to be done by the Controller and the switches. When the mitigation module identified an attack, it immediately began the process of blocking attack traffic and cleared the switch flow tables of any and all flows that were associated with phony traffic.

Xu et al. [68] presented an original method for detecting Distributed Denial of Service attacks that consisted of two phases: victim detection and attack detection. After determining who the victim was, their system used something called self-organizing maps (SOM) in conjunction with neural network technology to determine whether network traffic was typical or an attempt at an attack. The authors verified their methodology by utilizing the topology of the Internet network.

A temporally-based approach to DDoS attack detection was proposed by Cui et al. [127], in which a back-propagation neural network was trained to extract attack patterns for identification purposes. When the attack detection module realized that it was under attack, it switched over to the attack defense module, which then closed the port where the malicious packets had first entered the system. However, because this operation could potentially block ports for legitimate users as well, the port recovery module dynamically restored the ports after they were blocked.

A DDoS attack detection mechanism based on deep learning in SDN was proposed by Li et al. [128]. This mechanism included a feature processing unit that processed raw data samples and provided a dataset for deep learning training. SDN stands for

software-defined networking. Based on the statistics that were provided by the Information Statistics module, the Flow Table Generator module determined the flow entries and priorities for various attack packages that could be dropped and sent to the OpenFlow switch. These flow entries and priorities were determined by the module. The authors trained on the ISCX dataset and then subjected their DDoS defensive architecture to real-time DDoS attacks to validate it.

In the study that Nam et al. [129] conducted, statistical tools and techniques involving neural networks were used in order to discover anomalous activity in the network. They utilised the entropy measure as a means of selecting particular features from among a set of attributes and utilized SOM as a means of categorizing network behavior. Emulation with the POX controller was used to verify that the authors' approach was successful.

A hybrid strategy was developed by Novaes et al. [130] as a means of identifying and mitigating assaults such as DDoS and port scans that occur in SDN networks. The operation of this plan consisted of three stages: characterization, the detection of anomalies, and mitigation. In order to quantify network properties, the authors made use of an entropy metric, and they used LSTM (Long Short-Term Memory) to simulate the signature of each attribute as it appears in regular traffic. After that, they made use of fuzzy logic in order to identify any irregularities in the network. The authors implemented the floodlight controller as a means of validating their methodology through the use of Mininet simulation.

4.1.4. Other Methods

Researchers have investigated other strategies as a means of detecting and mitigating DDoS assaults in SDN. These strategies are in addition to those already listed. These techniques include a wide variety of methodologies, including the SYN cookie algorithm, TRW-CB, rate limitation, graph theory, queuing theory, Bloom filters, and cumulative sums, among others. Table 8 provides an overview of the various strategies that are derived from these methods:

An intrusion detection system for software-defined networks that makes use of the TRW-CM and rate-limiting algorithms was presented by Dotcenko et al. [131]. Their strategy was put to the test on Mininet and passed with flying colors thanks to the Beacon controller.

Chin et al. [132] presented a strategy for the prevention of attack and the discovery of malicious activity that was developed specifically for TCP SYN flood assaults. The strategy makes use of two different components, namely a monitor and a correlator. The monitor maintains a constant listening posture, keeping an ear out for certain kinds of attacks, and communicating any relevant information to the correlator. The correlator is responsible for communicating with the monitor and Open vSwitch (OVS) in order to detect and mitigate attacks. This is accomplished by verifying the attacks' presence, running additional queries based on the most recent flow table, and making use of the OpenFlow API to drop malicious traffic.

A DDoS attack mitigation architecture called DaMask was introduced by Wang et al. [85], and it was built on software-defined networking. The DaMask-D module is the one responsible for generating attack alarms, and these alerts are then forwarded to the DaMask-M module.

Bloom filters were presented as part of an attack detection architecture for use in SDN by Xiao and colleagues [133]. Bloom filters are utilized by the framework's modules in order to gather and identify anomalous flow patterns. A testbed built with Mininet was utilized in order to verify the validity of the technique.

An strategy that makes use of graph theory was suggested by Aleroud et al. [134] as a method for identifying assaults in SDN networks. They began by obtaining traditional network datasets and then extracted an attack signature database from those datasets. Next, they examined various samples of OpenFlow to establish whether or not previously developed signatures could be utilized in flow

analysis. The authors claimed that the findings obtained using their methodology were superior to those obtained using [125], [143], and [149].

In the context of SDN, Conti et al. [135] proposed a lightweight DDoS attack detection technique. To detect DDoS assaults, they utilized a non-parametric method known as Cumulative Sum. Both the CAIDA Internet traces and the DARPA intrusion detection evaluation dataset were utilized in the process of validating the technique.

SDNScore is a statistical and packet-based security mechanism developed by Kalkan et al. [136] for use in SDN networks as a protection against DDoS attacks. This hybrid system gives switches the ability to collect statistical data and collaborate with the controller to come to decisions regarding current DDoS attacks. Instead of discarding all of the packets that make up a flow, a packet-based analysis is used to identify and remove attack packets.

Bhushan et al. [28] suggested an innovative method for sharing flow tables as a means of shielding SDN networks against DDoS attacks that use flow table overloading. In order to make the network more resistant to overflow DDoS attacks, this strategy makes use of alternative switch flow table space while keeping communication overhead to a minimum. A mathematical model that is based on queuing theory can provide an approximation of the utilised and unused flow table space. During an attack, the method examines the flow table status of every other switch in order to locate a switch that is appropriate for the situation. Through the use of the network simulator Mininet, the authors were able to demonstrate the viability of their methodology.

These varied approaches offer multiple ways to deal with DDoS attacks in SDN, reflecting the ongoing attempts to improve detection and mitigation strategies in this area of the domain.

4.2. Review of DDoS Mitigation Techniques in SDN

In order to protect network resources from such ubiquitous dangers, it is of the utmost necessity to take measures to mitigate distributed denial of service, or DDoS, attacks. Researchers have successfully mitigated DDoS assaults using a variety of strategies while operating within networks that were built using the SDN architecture. These methods include connection migration, packet migration, limiting the bandwidth of inflows, adjusting timeouts, and implementing controller-to-controller communication protocols. Table 9 provides a complete description of the various strategies for mitigating the impact, which is followed by the following discussion in greater detail:

Shin et al. [75] presented Avant-Guard, a novel method to protect against saturation attacks by expanding the capabilities of the preexisting OpenFlow data plane. The actuation trigger module and the connection migration module are both incorporated into Avant-Guard. The connection migration module makes it possible to switch failed Transmission Control Protocol (TCP) sessions on the data plane before notifying the control plane of the failure. While this is happening, the actuating trigger module is collecting information about the network state as well as the payload of the packets, which allows it to activate particular flow rules based on predetermined criteria. The authors constructed Avant-Guard by making use of the NetFPGA architecture so that they could demonstrate the efficacy of their methodology.

FloodGuard is a framework that was presented by Wang et al. [138]. It is supposed to be simple, effective, and independent of any specific protocol in order to improve the safety of SDN networks. FloodGuard is made up of two essential components, namely the proactive module and the packet migration module. By keeping an eye on the SDN controller's runtime logic, the proactive module is able to generate proactive flow rules in real time, which, in turn, helps to ensure that network policy is consistently adhered to. On the other hand, the packet migration module would temporarily cache data packets before submitting them to the controller through the utilization of rate restriction and Round Robin scheduling

mechanisms. This will effectively prevent the controller from becoming overwhelmed.

Piedrahita et al. [139] created FlowFense, a method that is both quick and lightweight for defending against DDoS attacks. In order to identify conditions of congestion, this method calls for assessing the level of use of the interfaces present on routers and the SDN controller. When the router detects congestion at particular interfaces, it will send an alert to the controller. This will allow the router to restrict the bandwidth that is available on those interfaces.

Wang et al. [140] developed a safe approach for access control that would authenticate entities before allowing access to the system. The structure is made up of three different modules: authentication, registration, and policy management; access control and communication policy; and traceback and audit policy. In order to communicate with one another, entities need to first register with an authentication and registration module, which gives them a password for subsequent communication. These modules were constructed by the authors on the application layer of the SDN architecture, and a POX controller was utilized in the validation of their strategy.

Using the available unused RAM all across the SDN system, Yuan et al. [141] designed a peer's support method for the purpose of mitigating flow table overflow DDoS attacks. Their plan takes into account every switch on a peer level, and in it, switches that are under attack are aided by other switches by utilizing the flow table space that is unused on those other switches. Utilizing queuing theory allows for the accurate calculation of vacant spaces on switches that are not currently under attack.

SDN Guard is a unique approach that was proposed by Dridi et al. [142] for safeguarding SDN networks against DDoS attacks. This is accomplished by dynamically modifying the route that harmful traffic takes and adjusting flow timeouts. The authors confirmed their method by putting it into practice using Mininet, which demonstrated a reduction of up to 32 percent in the amount of influence it had on controller performance.

In order to defend against flooding attacks, Phan et al. [143] developed an improved technique that they named Idle-time Adjustment (IA). In order to process flow information, this system uses something called a Support Vector Machine (SVM). Flows are either forwarded to the policy enforcement module or analyzed by the IA algorithm, the latter of which evaluates whether a flow is normal or whether it requires the adoption of a new policy based on the output of the SVM.

ArOMA is a strategy that was presented by Sahay and colleagues [144] to combat DDoS attacks. This strategy makes use of the centralized manageability and programmability capabilities of SDN. In this method, a controller at the endpoint of the Internet service provider (ISP) receives alerts and then generates policies for switches to deal with the threats. In order to validate their method, the authors' implementation included the use of a Ryu controller.

Hameed et al. [145] presented a collaborative strategy for reducing the effects of DDoS attacks in the context of SDN. They devised a protocol known as Controller-to-Controller (C-to-C) that allows for secure communication and the exchange of attack information between SDN controllers. It was decided to implement the POX controller on Mininet in order to evaluate their methodology.

A defense against DDoS assaults in SDN that include route spoofing and resource fatigue was suggested by Conti et al. [146]. Their Selective Blocking module will gather IP and MAC address data, and then inform the controller so that further action can be taken. The Periodic Monitoring module performs a calculation to determine the entropy of the destination IP and port in order to identify aberrant activity. The authors put their method into action in a hypothetical target environment by using Mininet.

Karmakar et al. [147] made use of a northbound application in order to reduce the effects of DDoS attacks in SDN. Their system can handle DDoS

attacks thanks to the security policy design and storage that it uses. In order to verify the efficacy of their strategy, the ONOS controller was built and deployed.

Wang et al. [148] came up with the SafeGuard Scheme (SGS) as a means of defending the control plane from DDoS assaults. The Backpropagation Neural Network (BPNN) approach is employed by the Anomaly Detection module in order to locate irregularities in the flow of network traffic. By implementing access control rules, the Controller Remapping procedure prevents hosts from delivering bogus traffic and redirects flows to other controllers.

These diverse approaches contribute to the effective mitigation of DDoS attacks in SDN networks, highlighting the ongoing efforts to enhance the security and resilience of such networks against these threats.

4. RESEARCH GAPS

Several research gaps have been identified as a result of a thorough review of the existing literature on DDoS defense solutions in the context of SDN .These research gaps indicate areas that need additional exploration. If these holes are patched, it will help in the creation and implementation of better secure solutions for SDN networks.

- The fact that simulation settings typically only have a single controller is a significant research hole that needs to be filled. In previous research (for example, [106–109, 111, 121, 122]), single controllers were utilized the majority of the time. These controllers include POX, NOX, Floodlight, and OpenDayLight. However, the availability of safe and robust controllers is limited, and depending entirely on a single controller presents the possibility of a failure point for the entire network [6]. The implementation of multiple controllers and distributed defense solutions should be investigated for use in future defense strategies. These solutions can help to distribute the overhead across a number of different machines and enable load balancing based on the requirements of individual scenarios. It is essential to minimize synchronization and communication overheads while taking into consideration the deployment of topologies with many controllers and the implementation of distributed DDoS defense measures.

- Another area of study that needs improvement is the integration of more sophisticated security modules into SDN switches. A number of researchers (for example [28, 75, 85, 98, 107, 128, 136, 139, 146]) have suggested adding these modules to SDN switches in order to improve the functionality of the switches. This method does lower the computational overhead of the controller and the communication overhead that exists between the data plane and the control plane; however, it does so at the expense of increased complexity and additional expenditures. As a result, there is a requirement to deploy security modules in switches in an effective manner while simultaneously reducing the amount of communication complexity between devices.

- Virtualizing network devices and the connections between them, which is a standard practice in validation and is accomplished with the help of tools like the Mininet emulator, has a substantial impact on the results. It is difficult to create an accurate model of Internet behavior via simulation tools since, to this day, there is no formula that can accurately describe Internet behavior [159]. In addition, the fact that my fellow researchers all utilize the same system in their experimental setups makes it difficult to validate security strategies. As a result, it is essential to incorporate a number of different physical machines in order to evaluate the proposed security strategies and take into consideration the complexity of real-world circumstances.

- A great number of solutions that are based on information theory metrics (for example, [97,

99, 100, 102–106]) use predefined threshold values for anomaly detection that are determined by the standard behavior of the network. However, due to the limited public deployment of SDN-based networks, it might be difficult to determine the suitable behavior to use as a baseline for these systems. In addition, the absence of benchmarked datasets reflecting regular traffic in addition to attack traffic is a factor that reduces the accuracy of these systems. It is common for traffic generator programs to be unable to correctly mimic the features of today's high-speed networks; as a result, the proportion of attack traffic, background traffic, and regular traffic that should be generated is thrown off [158]. Therefore, forecasting the appropriate behavior of a baseline network and getting datasets that have been benchmarked remain key areas of study need.

- According to the findings of earlier authors [114–120], the availability of normal traffic datasets that are specific to SDN networks is an essential component in the process of training machine learning models for DDoS detection. However, these methods are hindered by the absence of such datasets, which are necessary for their use. The usage of artificially generated datasets by other academics results in the introduction of bias and a failure to adequately portray the complexity of situations that occur in the actual world. As a consequence of this, there is an urgent need to solve the research gap of training machines with the appropriate normal behavior.

- Low-rate DDoS assaults continue to be difficult to detect, despite the fact that many academics have concentrated their efforts on developing methods to identify high-rate DDoS attacks by evaluating aberrations in network activity in comparison to regular traffic flow. Because sophisticated low-rate assaults are able to circumvent existing protection systems, it is imperative that effective detection tools that specifically target such attacks be developed.

- Multiple parameters are used by DDoS detection and mitigation systems that are based on artificial neural networks (ANN) in order to evaluate the present status of the network. In spite of the fact that these solutions work within the control plane of the SDN architecture, the excessive computational overhead that results from the usage of many parameters might have an adverse effect on the performance of the centralized controller that is responsible for policymaking. As a result, in order to keep controller performance stable, it is absolutely necessary to cut down on the total number of parameters used in ANN-based approaches for DDoS detection.

- Validation methods used by certain publications, such as Jiang et al. [105], rely on small network topologies that are not typical of practical settings. These topologies were used in the validation process. Validation efforts need to take into account plausible network topologies in order to guarantee that the offered solutions are actually implementable.

- Although using several controllers in a master-slave design might increase network performance, as shown by Wang et al. [148], it is important not to disregard the synchronization overhead that comes along with using many controllers. The lack of study on efficient synchronization techniques is a research gap that needs to be filled if one wants to ensure that such systems are effective.

- When it comes to DDoS attack detection, the vast majority of security solutions rely on network traffic characteristics derived from SDN switches (e.g., [68, 125, 126, 131, 139]). When utilizing native OpenFlow statistics gathering approaches, however, a significant amount of processing overhead is introduced onto the centralized control plane. This is

especially the case in large-scale networks. Although some writers have used the sFlow technique to circumvent this issue, the fact that it only records a subset of the information available means that the accuracy of the defense solution is compromised. As a result, there is a vacuum in research on the discovery of ways to collect network statistics while keeping the accuracy level high despite having a small amount of overhead.

- Some writers have attempted to detect DDoS attacks in software-defined networks (SDNs) by utilizing generalized information entropy metrics, such as Renyi's Entropy and -Entropy (e.g., [103, 104, 111, 120]). In comparison to the more conventional Shannon Entropy measure, these metrics produce significantly better results. Nevertheless, choosing the best value for the entropic index parameter that should be used presents a substantial challenge.

- The usage of a single SDN controller in the validation efforts of several authors (for example, [75, 139, 140, 142, 145–147]) raises concerns over the vulnerability of the central controller itself when DDoS attacks are carried out. In order to solve this problem, it is required to utilize a number of different SDN controllers in order to replicate realistic distributed network topologies that are representative of real-world settings.

In conclusion, these identified research gaps emphasize the need for further investigation and innovation in the field of DDoS defense solutions in SDN. By addressing these gaps, researchers can contribute to the development of more secure and resilient SDN networks in practical settings.

5. CONCLUSION AND FUTURE DIRECTIONS

The increase in the number of Internet-based services and apps has resulted in an increase in the number of threats that pose dangers to the stability of those services and applications. Over the course of time, a great number of workable solutions have been developed to protect the infrastructure of networks. SDN , one of these solutions, has evolved as a reliable and resilient strategy to tackling network-wide difficulties related to flexibility, management, and adaptation. This makes SDN a standout among these solutions. The capacity of SDN to partition network functions into many layers improves management of the network and brings novel approaches to network security. The adoption of SDN, on the other hand, is not without its hurdles, as it continues to be vulnerable to DDoS assaults, which presents a substantial obstacle in its deployment.

This academic study provides a comprehensive analysis of innovative forms of DDoS attacks that especially target SDN settings. These attacks deviate from standard DDoS attack patterns in several ways. The evaluation takes into account about seventy significant research articles in the aforementioned subject. According to the findings of our research, approximately 47 percent of researchers have relied on techniques that are based on information theory, approximately 42 percent have relied on strategies that are based on machine learning, and approximately 20 percent have relied on techniques that are based on artificial neural networks to identify DDoS attacks in SDN. In addition, the study dives into the technical features of a variety of security measures in an effort to improve fellow researchers' understanding of cutting-edge procedures. It is essential to keep in mind that the controller, which continues to be a key target for attackers despite playing a pivotal part in SDN-enabled networks, will continue to be a focus of their attention. It was determined that there were gaps in the study that were detected in the current literature, and these gaps were properly scrutinized and carefully debated.

In terms of the work that will be done in the future, the primary focus of our efforts will be on the creation of a distributed DDoS detection and mitigation system that makes use of generalized information theory metrics. Within SDN networks,

the overarching goal is to reduce the amount of work that must be performed by a single controller. In addition, we are aware of the necessity to solve the difficult challenge of distinguishing flash events from high-rate DDoS attacks, which share commonalities with one another.

REFERENCES

[1] Internet growth usage statistics, 2019, https://www.clickz.com/internetgrowth-usage-stats-2019-time-online-devices-users/235102/.

[2] DoS attack report, 2020, https://www.britannica.com/technology/denialof-service-attack.

[3] M. Feily, A. Shahrestani, S. Ramadass, A survey of botnet and botnet detection, in: 2009 Third International Conference on Emerging Security Information, Systems and Technologies, IEEE, 2009, pp. 268–273.

[4] M. Abu Rajab, J. Zarfoss, F. Monrose, A. Terzis, A multifaceted approach to understanding the botnet phenomenon, in: Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement, 2006, pp. 41–52.

[5] B. Saha, A. Gairola, Botnet: an overview, CERT-In White Paper, CIWP-2005-05, Vol. 240, 2005.

[6] N.Z. Bawany, J.A. Shamsi, K. Salah, DDoS attack detection and mitigation using SDN: methods, practices, and solutions, Arab. J. Sci. Eng. 42 (2) (2017) 425–441.

[7] M.M. Joëlle, Y.-H. Park, Strategies for detecting and mitigating DDoS attacks in SDN: A survey, J. Intell. Fuzzy Systems 35 (6) (2018) 5913–5925.

[8] S. Dong, K. Abbas, R. Jain, A survey on distributed denial of service (DDoS) attacks in SDN and cloud computing environments, IEEE Access 7 (2019) 80813–80828.

[9] A.P. Fajar, T.W. Purboyo, A survey paper of distributed denial-of-service attack in software defined networking (sdn), Int. J. Appl. Eng. Res. 13 (1) (2018) 476–482.

[10] X. Xu, H. Yu, K. Yang, DDoS attack in software defined networks: a survey, ZTE Commun. 15 (3) (2017).

[11] K. Kalkan, G. Gur, F. Alagoz, Defense mechanisms against DDoS attacks in SDN environment, IEEE Commun. Mag. 55 (9) (2017) 175–179.

[12] M.P. Singh, A. Bhandari, New-flow based DDoS attacks in SDN: Taxonomy, rationales, and research challenges, Comput. Commun. (2020). [13] C. Douligeris, D.N. Serpanos, Network Security: Current Status and Future Directions, John Wiley & Sons, 2007.

[14] B. Mukherjee, L.T. Heberlein, K.N. Levitt, Network intrusion detection, IEEE Netw. 8 (3) (1994) 26–41.

[15] D. Kreutz, F.M. Ramos, P.E. Verissimo, C.E. Rothenberg, S. Azodolmolky, S. Uhlig, Software-defined networking: A comprehensive survey, Proc. IEEE 103 (1) (2014) 14–76.

[16] T. Benson, A. Akella, D.A. Maltz, Unraveling the complexity of network management, in: NSDI, 2009, pp. 335–348.

[17] W. Xia, Y. Wen, C.H. Foh, D. Niyato, H. Xie, A survey on software-defined networking, IEEE Commun. Surv. Tutor. 17 (1) (2014) 27–51.

[18] J. Pan, S. Paul, R. Jain, A survey of the research on future internet architectures, IEEE Commun. Mag. 49 (7) (2011) 26–36.

[19] L. Popa, A. Ghodsi, I. Stoica, HTTP as the narrow waist of the future Internet, in: Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks, 2010, pp. 1–6.

[20] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J.D. Thornton, D.K. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos, et al., Named data networking (ndn) project, in: Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC, Vol. 157, Citeseer, 2010, p. 158.

[21] A.T. Campbell, H.G. De Meer, M.E. Kounavis, K. Miki, J.B. Vicente, D. Villela, A survey of programmable networks, ACM SIGCOMM Comput. Commun. Rev. 29 (2) (1999) 7–23. [22] O.N. Fundation, Software-defined networking: The new norm for networks, ONF White Paper, Vol. 2, pp. 2–6.

[23] I. Ahmad, S. Namal, M. Ylianttila, A. Gurtov, Security in software defined networks: A survey, IEEE Commun. Surv. Tutor. 17 (4) (2015) 2317–2346. J. Singh and S. Behal / Computer Science Review 37 (2020) 100279 23 [24] S. Shin, G. Gu, Attacking software-defined networks: A first feasibility study, in: Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, 2013, pp. 165–166.

[25] P. Fonseca, R. Bennesby, E. Mota, A. Passito, A replication component for resilient OpenFlow-based networking, in: 2012 IEEE Network Operations and Management Symposium, IEEE, 2012, pp. 933–939.

[26] S. Scott-Hayward, S. Natarajan, S. Sezer, A survey of security in software defined networks, IEEE Commun. Surv. Tutor. 18 (1) (2015) 623–654.

[27] S.T. Ali, V. Sivaraman, A. Radford, S. Jha, A survey of securing networks using software defined networking, IEEE Trans. Reliab. 64 (3) (2015) 1086–1097.

[28] K. Bhushan, B.B. Gupta, Distributed denial of service (DDoS) attack mitigation in software defined network (SDN)-based cloud computing environment, J. Ambient Intell. Humaniz. Comput. 10 (5) (2019) 1985–1997.

[29] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama, et al., Onix: A distributed control platform for large-scale production networks, in: OSDI, Vol. 10, 2010, pp. 1–6.

[30] OpenFlow switch, 2020, https://www.opennetworking.org/wp-content/ uploads/2014/10/openflow-switch-v1.5.1.pdf. (Accessed on 11 March 2020).

[31] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, OpenFlow: enabling innovation in campus networks, ACM SIGCOMM Comput. Commun. Rev. 38 (2) (2008) 69–74.

[32] Open networking foundation, 2020, https://www.opennetworking.org. [33] A. Lara, A. Kolasani, B. Ramamurthy, Network innovation using openflow: A survey, IEEE Commun. Surv. Tutor. 16 (1) (2013) 493–512. [34] B.A.A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, T. Turletti, A survey of software-defined networking: Past, present, and future of programmable networks, IEEE Commun. Surv. Tutor. 16 (3) (2014) 1617–1634.

[35] Y. Jarraya, T. Madi, M. Debbabi, A survey and a layered taxonomy of software-defined networking, IEEE Commun. Surv. Tutor. 16 (4) (2014) 1955–1980.

[36] R. Presuhn, J. Case, K. McCloghrie, M. Rose, S. Waldbusser, Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP), Tech. Rep., 2002, STD 62, RFC 3416, December.

[37] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, J. Wilcox, Intelligent design enables architectural evolution, in: Proceedings of the 10th ACM Workshop on Hot Topics in Networks, 2011, pp. 1–6.

[38] B. Raghavan, M. Casado, T. Koponen, S. Ratnasamy, A. Ghodsi, S. Shenker, Software-defined internet architecture: decoupling architecture from infrastructure, in: Proceedings of the 11th ACM Workshop on Hot Topics in Networks, 2012, pp. 43–48.

[39] H. Kim, N. Feamster, Improving network management with software defined networking, IEEE Commun. Mag. 51 (2) (2013) 114–119.

[40] J. Sherry, S. Ratnasamy, J.S. At, A Survey of Enterprise Middlebox Deployments, Technical Report No. UCB/EECS-2012-24, Citeseer, 2012.

[41] Technical Report on SDN, 2019, http://www2.technologyreview.com/ article/412194/tr10-software-defined-networking/.

[42] H. Jamjoom, D. Williams, U. Sharma, Don't call them middleboxes, call them middlepipes, in: Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, 2014, pp. 19–24.

[43] S. Shenker, M. Casado, T. Koponen, N. McKeown, et al., The future of networking, and the past of protocols, Open Networking Summit, Vol. 20, 2011, pp. 1–30.

[44] H. Alkhatib, P. Faraboschi, E. Frachtenberg, H. Kasahara, D. Lange, P. Laplante, A. Merchant, D. Milojicic, K. Schwan, IEEE CS 2022 Report (Draft), Tech. Rep., IEEE Computer Society, 2014.

[45] S. Scott-Hayward, G. O'Callaghan, S. Sezer, SDN security: A survey, in: 2013 IEEE SDN for Future Networks and Services, SDN4FNS, IEEE, 2013, pp. 1–7.

[46] A. Doria, J.H. Salim, R. Haas, H.M. Khosravi, W. Wang, L. Dong, R. Gopal, J.M. Halpern, Forwarding and control element separation (ForCES) protocol specification, RFC 5810 (2010) 1–124.

[47] A. Tewari, B. Gupta, Security, privacy and trust of different layers in Internet-of-Things (IoTs) framework, Future Gener. Comput. Syst. (2018).

[48] H. Song, Protocol-oblivious forwarding: Unleash the power of SDN through a future-proof forwarding plane, in: Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, 2013, pp. 127–132. [49] T. Ubale, A.K. Jain, Survey on DDoS attack techniques and solutions in software-defined network, in: Handbook of Computer Networks and Cyber Security, Springer, 2020, pp. 389–419. [50] Nox controller, 2020, https://github.com/noxrepo/nox. (Accessed on 11 March 2020). [51] Pox controller, 2020, https://github.com/noxrepo/pox. (Accessed on 11 March 2020). [52] Project floodlight, 2020, http://www.projectfloodlight.org/floodlight/. (Accessed on 11 March 2020). [53] Ryu, 2020, https://osrg.github.io/ryu/. (Accessed on 11 March 2020). [54] S. Khan, A. Gani, A.W.A. Wahab, A. Abdelaziz, M.A. Bagiwa, FML: A novel forensics management layer for software defined networks, in: 2016 6th International Conference-Cloud System and Big Data Engineering, Confluence, IEEE, 2016, pp. 619–623. [55] A. Voellmy, H. Kim, N. Feamster, Procera: a language for high-level reactive network control, in: Proceedings of the First Workshop on Hot Topics in Software Defined Networks, 2012, pp. 43–48.

[56] C.J. Anderson, N. Foster, A. Guha, J.-B. Jeannin, D. Kozen, C. Schlesinger, D. Walker, NetKAT: Semantic foundations for networks, ACM SIGPLAN Not. 49 (1) (2014) 113–126.

[57] N. Foster, R. Harrison, M.J. Freedman, C. Monsanto, J. Rexford, A. Story, D. Walker, Frenetic: A network programming language, ACM SIGPLAN Not. 46 (9) (2011) 279–291.

[58] A. Tootoonchian, Y. Ganjali, Hyperflow: A distributed control plane for openflow, in: Proceedings of the 2010 Internet Network Management Conference on Research on Enterprise Networking, Vol. 3, 2010.

[59] Opendaylight user guide, 2020, https://docs.opendaylight.org/en/stablefluorine/user-guide/alto-user-guide.html. (Accessed on 11 March 2020).

[60] H. Uppal, D. Brandon, OpenFlow Based Load Balancing, CSE561: Networking Project Report, University of Washington, Citeseer, 2010.

[61] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, S. Shenker, NOX: towards an operating system for networks, ACM SIGCOMM Comput. Commun. Rev. 38 (3) (2008) 105–110. [62] K. Dhamecha, B. Trivedi, Sdn issues-a survey, Int. J. Comput. Appl. 73 (18) (2013).

[63] A. Voellmy, P. Hudak, Nettle: Taking the sting out of programming network routers, in: International Symposium on Practical Aspects of Declarative Languages, Springer, 2011, pp. 235–249.

[64] W. Stallings, Software-defined networks and openflow, Internet Protocol J. 16 (1) (2013) 2–14. [65] F. Hu, Q. Hao, K. Bao, A survey on software-defined network and openflow: From concept to implementation, IEEE Commun. Surv. Tutor. 16 (4) (2014) 2181–2206.

[66] P. Manso, J. Moura, C. Serrão, SDN-based intrusion detection system for early detection and mitigation of DDoS attacks, Information 10 (3) (2019) 106.

[67] J. Zheng, Q. Li, G. Gu, J. Cao, D.K. Yau, J. Wu, Realtime DDoS defense using COTS SDN switches via adaptive correlation analysis, IEEE Trans. Inf. Forensics Secur. 13 (7) (2018) 1838–1853.

[68] Y. Xu, Y. Liu, DDoS attack detection under SDN context, in: IEEE INFOCOM 2016-the 35th Annual IEEE International Conference on Computer Communications, IEEE, 2016, pp. 1–9.

[69] Z. Liu, R.H. Campbell, M. Mickunas, Active security support for active networks, IEEE Trans. Syst. Man Cybern. C Appl. Rev. 33 (4) (2003) 432–445.

[70] S.W. Shin, P. Porras, V. Yegneswaran, G. Gu, A framework for integrating security services into software-defined networks, in: Open Networking Summit, Open Networking Summit, 2013.

[71] X. Wen, Y. Chen, C. Hu, C. Shi, Y. Wang, Towards a secure controller platform for openflow applications, in: Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, 2013, pp. 171–172.

[72] S. Hartman, M. Wasserman, D. Zhang, Security requirements in the software defined networking model, 2013, Internet Engineering Task Force, Internet-Draft draft-hartman-sdnsec-requirements-01.

[73] H. Xie, T. Tsou, D. Lopez, H. Yin, V. Gurbani, Use cases for ALTO with software defined networks, 2012, Working Draft, IETF Secretariat, Internet-Draft draft-xie-alto-sdn-extension-use-cases-01. txt.

[74] J. Naous, D. Erickson, G.A. Covington, G. Appenzeller, N. McKeown, Implementing an OpenFlow switch on the NetFPGA platform, in: Proceedings of the 4th ACM/IEEE Symposium on Architectures for Networking and Communications Systems, 2008, pp. 1–9.

[75] S. Shin, V. Yegneswaran, P. Porras, G. Gu, Avant-guard: Scalable and vigilant switch flow management in software-defined networks, in: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, 2013, pp. 413–424.

[76] G. Yao, J. Bi, L. Guo, On the cascading failures of multi-controllers in software defined networks, in: 2013 21st IEEE International Conference on Network Protocols, ICNP, IEEE, 2013, pp. 1–2.

[77] Crippling cyber-attacks, 1998, https://www.bbc.com/news/technology35376327. 25/3/2019, (Accessed on 13 February 2020).

[78] N.Z. Bawany, J.A. Shamsi, Application layer DDoS attack defense framework for smart city using SDN, in: The Third International Conference on Computer Science, Computer Engineering, and Social Media, CSCESM2016, 2016, p. 1.

[79] S. Jajodia, K. Kant, P. Samarati, A. Singhal, V. Swarup, C. Wang, Secure Cloud Computing, Springer, 2014. 24 J. Singh and S. Behal / Computer Science Review 37 (2020) 100279

[80] S. Bu, F.R. Yu, X.P. Liu, H. Tang, Structural results for combined continuous user authentication and intrusion detection in high security mobile ad-hoc networks, IEEE Trans. Wireless Commun. 10 (9) (2011) 3064–3073.

[81] S. Sezer, S. Scott-Hayward, P.K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, N. Rao, Are we ready for SDN? Implementation challenges for software-defined networks, IEEE Commun. Mag. 51 (7) (2013) 36–43

. [82] A. Wang, Y. Guo, F. Hao, T. Lakshman, S. Chen, Scotch: Elastically scaling up sdn control-plane using vswitch based overlay, in: Proceedings of the 10th ACM International

on Conference on Emerging Networking Experiments and Technologies, 2014, pp. 403–414.

[83] Q. Yan, F. Yu, Q. Gong, J. Li, Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges, IEEE Commun. Surv. Tutor. 18 (1) (2015) 602–622.

[84] T. Ubale, A.K. Jain, Taxonomy of DDoS attacks in software-defined networking environment, in: International Conference on Futuristic Trends in Network and Communication Technologies, Springer, 2018, pp. 278–291.

[85] B. Wang, Y. Zheng, W. Lou, Y. Hou, DDoS attack protection in the era of cloud computing and software-defined networking, Comput. Netw. 81 (2015) 308–319.

[86] D. Kreutz, F.M. Ramos, P. Verissimo, Towards secure and dependable software-defined networks, in: Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, 2013, pp. 55–60.

[87] L. Schehlmann, S. Abt, H. Baier, Blessing or curse? Revisiting security aspects of software-defined networking, in: 10th International Conference on Network and Service Management (CNSM) and Workshop, IEEE, 2014, pp. 382–387.

[88] Open Networking Specifications 1.5.1, Vol. 3, Open Networking Foundation, 2015.

[89] E. Spitznagel, D. Taylor, J. Turner, Packet classification using extended TCAMs, in: 11th IEEE International Conference on Network Protocols, 2003. Proceedings, IEEE, 2003, pp. 120–131.

[90] M. Parashar, A. Poonia, K. Satish, A survey of attacks and their mitigations in software defined networks, in: 2019 10th International Conference on Computing, Communication and Networking Technologies, ICCCNT, IEEE, 2019, pp. 1–8.

[91] A. Akhunzada, E. Ahmed, A. Gani, M.K. Khan, M. Imran, S. Guizani, Securing software defined networks: taxonomy, requirements, and open issues, IEEE Commun. Mag. 53 (4) (2015) 36–44. [92] J.M. Dover, A Denial of Service Attack Against the Open Floodlight SDN Controller, Tech. Rep., Dover Networks, 2013.

[93] R. Kandoi, M. Antikainen, Denial-of-service attacks in OpenFlow SDN networks, in: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), IEEE, 2015, pp. 1322–1326.

[94] P. Zhang, H. Wang, C. Hu, C. Lin, On denial of service attacks in software defined networks, IEEE Netw. 30 (6) (2016) 28–33.

[95] C.E. Shannon, A mathematical theory of communication, Bell Syst. Tech. J. 27 (1948) 379–423.

[96] C.H. Bennett, P. Gács, M. Li, P.M. Vitányi, W.H. Zurek, Information distance, IEEE Trans. Inform. Theory 44 (4) (1998) 1407–1423.

[97] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, V. Maglaris, Combining openFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments, Comput. Netw. 62 (2014) 122–136.

[98] R. Wang, Z. Jia, L. Ju, An entropy-based distributed DDoS detection mechanism in software-defined networking, in: 2015 IEEE Trustcom/BigDataSE/ISPA, Vol. 1, IEEE, 2015, pp. 310–317.

[99] S.M. Mousavi, M. St-Hilaire, Early detection of DDoS attacks against SDN controllers, in: 2015 International Conference on Computing, Networking and Communications, ICNC, IEEE, 2015, pp. 77–81.

[100] J. Boite, P.-A. Nardin, F. Rebecchi, M. Bouet, V. Conan, Statesec: Stateful monitoring for DDoS protection in software defined networks, in: 2017 IEEE Conference on Network Softwarization, NetSoft, IEEE, 2017, pp. 1–9

. [101] S.-C. Tsai, I.-H. Liu, C.-T. Lu, C.-H. Chang, J.-S. Li, Defending cloud computing environment against the challenge of DDoS attacks based on software defined network, in: Advances in Intelligent Information Hiding and Multimedia Signal Processing, Springer, 2017, pp. 285–292.

[102] K. Kalkan, L. Altay, G. Gür, F. Alagöz, JESS: Joint entropy-based DDoS defense scheme in SDN, IEEE J. Sel. Areas Commun. 36 (10) (2018) 2358–2372.

[103] K.S. Sahoo, D. Puthal, M. Tiwary, J.J. Rodrigues, B. Sahoo, R. Dash, An early detection of low rate DDoS attack to SDN based data center networks using information distance metrics, Future Gener. Comput. Syst. 89 (2018) 685–697.

[104] K.S. Sahoo, M. Tiwary, B. Sahoo, Detection of high rate DDoS attack from flash events using information metrics in software defined networks, in: 2018 10th International Conference on Communication Systems & Networks, COMSNETS, IEEE, 2018, pp. 421–424.

[105] Y. Jiang, X. Zhang, Q. Zhou, Z. Cheng, An entropy-based DDoS defense mechanism in software defined networks, in: International Conference on Communicatins and Networking in China, Springer, 2016, pp. 169–178.

[106] G.-C. Hong, C.-N. Lee, M.-F. Lee, Dynamic threshold for DDoS mitigation in SDN environment, in: 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC, IEEE, 2019, pp. 1–7.

[107] N.Z. Bawany, J.A. Shamsi, SEAL: SDN based secure and agile framework for protecting smart city applications from DDoS attacks, J. Netw. Comput. Appl. 145 (2019) 102381.

[108] A. Ahalawat, S.S. Dash, A. Panda, K.S. Babu, Entropy based DDoS detection and mitigation in openflow enabled SDN, in: 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking, ViTECoN, IEEE, 2019, pp. 1–5.

[109] M. Xuanyuan, V. Ramsurrun, A. Seeam, Detection and mitigation of DDoS attacks using conditional entropy in software-defined networking.

[110] J. Cui, M. Wang, Y. Luo, H. Zhong, DDoS detection and defense mechanism based on cognitive-inspired computing in SDN, Future Gener. Comput. Syst. 97 (2019) 275–283.

[111] R. Li, B. Wu, Early detection of DDoS based on phi-entropy in SDN networks, in: 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference, Vol. 1, ITNEC, IEEE, 2020, pp. 731–735.

[112] N. Pitropakis, E. Panaousis, T. Giannetsos, E. Anastasiadis, G. Loukas, A taxonomy and survey of attacks against machine learning, Comp. Sci. Rev. 34 (2019) 100199.

[113] N. Bindra, M. Sood, Detecting DDoS attacks using machine learning techniques and contemporary intrusion detection dataset, Autom. Control Comput. Sci. 53 (5) (2019) 419–428.

[114] Q. Niyaz, W. Sun, A.Y. Javaid, A deep learning based DDoS detection system in software-defined networking (SDN), 2016, arXiv preprint arXiv: 1611.07400.

[115] T. Hurley, J.E. Perdomo, A. Perez-Pons, HMM-based intrusion detection system for software defined networking, in: 2016 15th IEEE International Conference on Machine Learning and Applications, ICMLA, IEEE, 2016, pp. 617–621.

[116] A. Alshamrani, A. Chowdhary, S. Pisharody, D. Lu, D. Huang, A defense system for defeating DDoS attacks in SDN based networks, in: Proceedings of the 15th ACM International Symposium on Mobility Management and Wireless Access, 2017, pp. 83–92.

[117] D. Hu, P. Hong, Y. Chen, FADM: DDoS flooding attack detection and mitigation system in software-defined networking, in: GLOBECOM 2017-2017 IEEE Global Communications Conference, IEEE, 2017, pp. 1–7.

[118] A.B. Dehkordi, M. Soltanaghaie, F.Z. Boroujeni, A New DDoS Detection Method in Software Defined Network. [119] J. Li, Z. Zhao, R. Li, H. Zhang, Ai-based two-stage intrusion detection for software defined iot networks, IEEE Internet Things J. 6 (2) (2018) 2093–2102.

[120] S. Guozi, W. JIANG, G. Yu, R. Danni, L. Huakang, DDoS attacks and flash event detection based on flow characteristics in SDN, in: 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS, IEEE, 2018, pp. 1–6.

[121] V. Deepa, K. Sudar, P. Deepalakshmi, Design of ensemble learning methods for DDoS detection in SDN environment, in: 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking, ViTECoN, IEEE, 2019, pp. 1–6.

[122] T.V. Phan, M. Park, Efficient distributed denial-of-service attack defense in SDN-based cloud, IEEE Access 7 (2019) 18701–18714.

[123] M. Myint Oo, S. Kamolphiwong, T. Kamolphiwong, S. Vasupongayya, Advanced support vector machine-(ASVM-) based detection for distributed denial of service (DDoS) attack on software defined networking (SDN), J. Comput. Netw. Commun. 2019 (2019).

[124] J. Li, Y. Liu, L. Gu, DDoS attack detection based on neural network, in: 2010 2nd International Symposium on Aware Computing, IEEE, 2010, pp. 196–199.

[125] R. Braga, E. Mota, A. Passito, Lightweight DDoS flooding attack detection using NOX/OpenFlow, in: IEEE Local Computer Network Conference, IEEE, 2010, pp. 408–415.

[126] Y. Cui, L. Yan, S. Li, H. Xing, W. Pan, J. Zhu, X. Zheng, SD-Anti-DDoS: Fast and efficient DDoS defense in software-defined networks, J. Netw. Comput. Appl. 68 (2016) 65–79.

[127] J. Cui, J. He, Y. Xu, H. Zhong, TDDAD: Time-based detection and defense scheme against DDoS attack on SDN controller, in: Australasian Conference on Information Security and Privacy, Springer, 2018, pp. 649–665.

[128] C. Li, Y. Wu, X. Yuan, Z. Sun, W. Wang, X. Li, L. Gong, Detection and defense of DDoS attack–based on deep learning in OpenFlow-based SDN, Int. J. Commun. Syst. 31 (5) (2018) e3497. J. Singh and S. Behal / Computer Science Review 37 (2020) 100279 25

[129] T.M. Nam, P.H. Phong, T.D. Khoa, T.T. Huong, P.N. Nam, N.H. Thanh, L.X. Thang, P.A. Tuan, V.D. Loi, et al., Self-organizing map-based approaches in DDoS flooding detection using SDN, in: 2018 International Conference on Information Networking, ICOIN, IEEE, 2018, pp. 249–254.

[130] M.P. Novaes, L.F. Carvalho, J. Lloret, M.L. Proença, Long shortterm memory and fuzzy logic for anomaly detection and mitigation in software-defined network environment, IEEE Access 8 (2020) 83765–83781.

[131] S. Dotcenko, A. Vladyko, I. Letenko, A fuzzy logic-based information security management for software-defined

networks, in: 16th International Conference on Advanced Communication Technology, IEEE, 2014, pp. 167–171.

[132] T. Chin, X. Mountrouidou, X. Li, K. Xiong, Selective packet inspection to detect DoS flooding using software defined networking (SDN), in: 2015 IEEE 35th International Conference on Distributed Computing Systems Workshops, IEEE, 2015, pp. 95–99.

[133] P. Xiao, Z. Li, H. Qi, W. Qu, H. Yu, An efficient DDoS detection with bloom filter in SDN, in: 2016 IEEE Trustcom/BigDataSE/ISPA, IEEE, 2016, pp. 1–6.

[134] A. AlEroud, I. Alsmadi, Identifying cyber-attacks on software defined networks: An inference-based intrusion detection approach, J. Netw. Comput. Appl. 80 (2017) 152–164.

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

[136] K. Kalkan, G. Gür, F. Alagöz, Sdnscore: A statistical defense mechanism against DDoS attacks in sdn environment, in: 2017 IEEE Symposium on Computers and Communications, ISCC, IEEE, 2017, pp. 669–675.

[137] J. Wang, R. Wen, J. Li, F. Yan, B. Zhao, F. Yu, Detecting and mitigating target link-flooding attacks using sdn, IEEE Trans. Dependable Secure Comput. 16 (6) (2018) 944–956.

[138] H. Wang, L. Xu, G. Gu, Floodguard: A dos attack prevention extension in software-defined networks, in: 2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, IEEE, 2015, pp. 239–250.

[139] A.F.M. Piedrahita, S. Rueda, D.M. Mattos, O.C.M. Duarte, Flowfence: a denial of service defense system for software defined networking, in: 2015 Global Information Infrastructure and Networking Symposium, GIIS, IEEE, 2015, pp. 1–6.

[140] X. Wang, M. Chen, C. Xing, SDSNM: a software-defined security networking mechanism to defend against DDoS attacks, in: 2015 Ninth International Conference on Frontier of Computer Science and Technology, IEEE, 2015, pp. 115–121.

[141] B. Yuan, D. Zou, S. Yu, H. Jin, W. Qiang, J. Shen, Defending against flow table overloading attack in software-defined networks, IEEE Trans. Serv. Comput. 12 (2) (2016) 231–246.

[142] L. Dridi, M.F. Zhani, SDN-guard: DoS attacks mitigation in SDN networks, in: 2016 5th IEEE International Conference on Cloud Networking, Cloudnet, IEEE, 2016, pp. 212–217.

[143] T.V. Phan, T. Van Toan, D. Van Tuyen, T.T. Huong, N.H. Thanh, OpenFlowSIA: An optimized protection scheme for software-defined networks from flooding attacks, in: 2016 IEEE Sixth International Conference on Communications and Electronics, ICCE, IEEE, 2016, pp. 13–18.

[144] R. Sahay, G. Blanc, Z. Zhang, H. Debar, ArOMA: An SDN based autonomic DDoS mitigation framework, Comput. Secur. 70 (2017) 482–499. [145] S. Hameed, H. Ahmed Khan, SDN based collaborative scheme for mitigation of DDoS attacks, Future Internet 10 (3) (2018) 23.

[146] M. Conti, C. Lal, R. Mohammadi, U. Rawat, Lightweight solutions to counter DDoS attacks in software defined networking, Wirel. Netw. 25 (5) (2019) 2751–2768. [147] K.K. Karmakar, V. Varadharajan, U. Tupakula, Mitigating attacks in software defined networks, Cluster Comput. 22 (4) (2019) 1143–1157.

[148] Y. Wang, T. Hu, G. Tang, J. Xie, J. Lu, SGS: Safe-guard scheme for protecting control plane against DDoS attacks in software-defined networking, IEEE Access 7 (2019) 34699–34710.

[149] A.S. Da Silva, C.C. Machado, R.V. Bisol, L.Z. Granville, A. SchaefferFilho, Identification and selection of flow features for accurate traffic classification in SDN, in: 2015 IEEE 14th International Symposium on Network Computing and Applications, IEEE, 2015, pp. 134–141.

[150] S. Agarwal, M. Kodialam, T. Lakshman, Traffic engineering in software defined networks, in: 2013 Proceedings IEEE INFOCOM, IEEE, 2013, pp. 2211–2219.

[151] C.E. Rothenberg, M.R. Nascimento, M.R. Salvador, C.N.A. Corrêa, S. Cunha de Lucena, R. Raszuk, Revisiting routing control platforms with the eyes and muscles of software-defined networking, in: Proceedings of the First Workshop on Hot Topics in Software Defined Networks, 2012, pp. 13–18.

[152] J. Xie, F.R. Yu, T. Huang, R. Xie, J. Liu, C. Wang, Y. Liu, A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges, IEEE Commun. Surv. Tutor. 21 (1) (2018) 393–430.

[153] S. Vissicchio, L. Vanbever, O. Bonaventure, Opportunities and research challenges of hybrid software defined networks, ACM SIGCOMM Comput. Commun. Rev. 44 (2) (2014) 70–75.

[154] J. McCauley, A. Panda, M. Casado, T. Koponen, S. Shenker, Extending SDN to large-scale networks, Open Networking Summit, 2013, pp. 1–2.

[155] S. Hassas Yeganeh, Y. Ganjali, Kandoo: a framework for efficient and scalable offloading of control applications, in: Proceedings of the First Workshop on Hot Topics in Software Defined Networks, 2012, pp. 19–24.

[156] W. Li, W. Meng, L.F. Kwok, A survey on openFlow-based software defined networks: Security challenges and countermeasures, J. Netw. Comput. Appl. 68 (2016) 126–139.

[157] S. Bhatia, G. Mohay, A. Tickle, E. Ahmed, Parametric differences between a real-world distributed denial-of-service attack and a flash event, in: 2011 Sixth International Conference on Availability, Reliability and Security, IEEE, 2011, pp. 210–217.

[158] S. Behal, K. Kumar, M. Sachdeva, Characterizing DDoS attacks and flash events: Review, research gaps and future directions, Comp. Sci. Rev. 25 (2017) 101–114.

[159] S. Floyd, V. Paxson, Difficulties in simulating the Internet, IEEE/ACM Trans. Netw. 9 (4) (2001) 392–403.

[160] L. Yao, P. Hong, W. Zhou, Evaluating the controller capacity in software defined networking, in: 2014 23rd International Conference on Computer Communication and Networks, ICCCN, IEEE, 2014, pp. 1–6.

[161] P. Wang, K.-M. Chao, H.-C. Lin, W.-H. Lin, C.-C. Lo, An efficient flow control approach for SDN-based network threat detection and migration using support vector machine, in: 2016 IEEE 13th International Conference on E-Business Engineering, ICEBE, IEEE, 2016, pp. 56–63