electronics

MDPI

Review

A Comprehensive Survey of Distributed Denial of Service Detection and Mitigation Technologies in Software-Defined Network

Yinghao Su [1,*], Dapeng Xiong [2], Kechang Qian [2] and Yu Wang [2]

[1] Institute of Graduate, Space Engineering University, Beijing 101416, China
[2] Institute of Aerospace Information, Space Engineering University, Beijing 101416, China
* Correspondence: hgdyinghao@ldy.edu.rs

Abstract: The widespread adoption of software-defined networking (SDN) technology has brought revolutionary changes to network control and management. Compared to traditional networks, SDN enhances security by separating the control plane from the data plane and replacing the traditional network architecture with a more flexible one. However, due to its inherent architectural flaws, SDN still faces new security threats. This paper expounds on the architecture and security of SDN, analyzes the vulnerabilities of SDN architecture, and introduces common distributed denial of service (DDoS) attacks within the SDN architecture. This article also provides a review of the relevant literature on DDoS attack detection and mitigation in the current SDN environment based on the technologies used, including statistical analysis, machine learning, policy-based, and moving target defense techniques. The advantages and disadvantages of these technologies, in terms of deployment difficulty, accuracy, and other factors, are analyzed. Finally, this study summarizes the SDN experimental environment and DDoS attack traffic generators and datasets of the reviewed literature and the limitations of current defense methods and suggests potential future research directions.

Keywords: software-defined network; distributed denial of service attacks; intrusion detection; network security

Citation: Su, Y.; Xiong, D.; Qian, K.; Wang, Y. A Comprehensive Survey of Distributed Denial of Service Detection and Mitigation Technologies in Software-Defined Network. *Electronics* **2024**, *13*, 807. https://doi.org/10.3390/electronics13040807

Academic Editors: Xiang Su, Liang Xiao, Nan Qi, Rugui Yao and Lin Zhang

Received: 31 December 2023
Revised: 29 January 2024
Accepted: 8 February 2024
Published: 19 February 2024

1. Introduction

With the increasing complexity of computer networks, traditional network architectures are finding it difficult to meet the requirements of current cloud computing, the mobile Internet, and other aspects for diversified and scalable network services. This is due to their fixed form and tight coupling of control and data-forwarding functions [1]. SDN, proposed in this context, is a new type of network architecture that separates the network control function from the data forwarding function, providing greater flexibility and programmability compared to traditional networks. Although SDN architecture achieves centralized network control and on-demand traffic forwarding, it still has significant security vulnerabilities and is more susceptible to security threats. Among them, denial-of-service attacks that disrupt the availability of SDN are a common attack method [2].

A denial of service attack is when an attacker sends malicious traffic to computer network hosts, depleting the network's limited resources and disrupting its availability, rendering it incapable of providing regular services. When attackers control a large number of hosts to launch a DoS attack, it becomes a distributed denial of service attack. DoS attacks exploit vulnerabilities in network protocols and the limited nature of network resources by sending a large number of invalid data packets. This consumes the network's bandwidth, connection, and service resources, ultimately preventing authorized users from accessing the network. Currently, DoS attacks have become a significant method of cyber warfare [3]. In the Russia–Ukraine conflict, Russia launched DDoS attacks against multiple military, government, and financial websites in Ukraine. These attacks caused several

critical infrastructures and important network systems to collapse, significantly impacting Ukraine's social order. Therefore, effectively preventing and mitigating network DDoS attacks has become an urgent problem that needs to be addressed.

While SDN offers benefits such as agility, flexibility, and programmability, it remains susceptible to DDoS attacks. Due to centralized management in the SDN architecture, DDoS attacks can easily overwhelm SDN controllers and switch flow tables, resulting in significant network performance degradation. Currently, numerous research topics focus on DDoS attack detection and mitigation technology in traditional networks. However, many of these solutions are not applicable to SDN controllers. At the same time, it is challenging to effectively detect new DDoS attacks in SDN environments. Hence, it is essential to systematically review the relevant literature on DDoS attack detection and mitigation technology in SDN environments.

Many survey papers on DDoS defense solutions are available in the literature, which is closely related to our work. In previous literature reviews, Mittal et al. [4] and Ali et al. [5] focused solely on examining DDoS attack defense strategies from a single technical standpoint. Karnani et al. [6] conducted a review specifically on mitigation strategies. While Ubale et al. [3] and Kaur et al. [7] provided comprehensive overviews of DDoS attack detection and mitigation technologies; however, these articles fail to include a summary of the existing literature on moving target defense technology utilizing the SDN network architecture. This article presents a comprehensive analysis of the current literature on the detection and mitigation of DDoS attacks in SDN. We categorize and examine the various technical approaches employed in this field, with a particular focus on moving target defense technology mitigation strategies, which have received limited attention in previous reviews. In addition, we also classified the reviewed literature according to the experimental environment used and summarized the existing technical issues and challenges faced in current research. Table 1 shows a comparison of the proposed study with existing survey papers in recent years.

Table 1. Comparison of proposed study with the existing studies.

Covered Topic		Ref. [3]	Ref. [5]	Ref. [6]	Ref. [7]	Ref. [8]	Our Work
Vulnerable points and DDoS attack types in SDN		√	√	√	√	√	√
DDoS attack detection technology	Statistical analysis and information entropy	-	-	√	√	√	√
	Machine learning	√	√	-	√	√	√
	Hybrid detection		√	-	√	√	√
DDoS attack mitigation techniques	Policy-based techniques	√	-	√	√	√	√
	Moving target defense	-	-	√	-	-	√
Experimental environment analysis		√	-	√	√	√	√
Research challenges and gaps		√	√	√	√	√	√

"√": The paper contains this content. "-": The paper does not contain this content.

The main contributions of our paper can be summarized as follows:

- We provide a description of the security vulnerabilities that exist in SDN as well as the prevalent DDoS attacks that target SDN networks.
- We conducted a literature review on popular DDoS attack detection and mitigation technologies in SDN and categorized and evaluated them according to the technologies utilized. DDoS attack detection and mitigation technologies in SDN environments encompass statistical analysis techniques, machine learning techniques, hybrid detection techniques, policy-based techniques, and, particularly, moving target defense techniques, which are less commonly discussed in the literature. Furthermore, we conducted a comparative assessment of the benefits and drawbacks linked to these technologies.

- Finally, we analyze the experimental environment used in the relevant literature and briefly summarize the research challenges and gaps in DDoS attack defense technology in SDN.

The rest of this paper is structured as follows: Section 2 elaborates on the methods used to search the literature during the research process of this article; Section 3 presents the vulnerable points and DDoS attacks in SDN; Section 4 presents the DDoS attack detection technology in SDN and Section 5 presents the DDoS attack mitigation techniques in SDN. In Section 6, we analyze the experimental part of the collected literature. Section 7 summarizes the research challenges and gaps in existing work. Section 8 concludes our work.

2. Research Methodology

Our research primarily focuses on detecting and mitigating DDoS attacks in SDN environments. Through a comprehensive review of the relevant literature, we aim to address the following questions:

- RQ 1: What are the weaknesses of SDN compared to traditional networks, and to what DDoS attacks is it more susceptible?
- RQ 2: What technical methods do researchers typically use to detect and mitigate DDoS attacks in SDN environments?
- RQ 3: What are the benefits and drawbacks of current detection and mitigation technologies? What are the current challenges in research?

To find papers related to the research questions, we followed a three-stage selection procedure: (1) identifying search terms, (2) selecting sources, and (3) applying inclusion/exclusion criteria to the selected papers:

- **Search terms:** This stage primarily determines the keywords to search and search strings. For the research of DDoS attacks on SDN, the identified keywords were as follows: "SDN", "DDoS", "Controller Resource Saturation", and "Flow Table Overloading". Meanwhile, to define the search string, the Boolean operation "OR" was used to select optional words and synonyms, while "AND" was used to select relevant terms, thereby generating the search string. The following keywords were selected as the search string: "(software-defined network" OR "SDN") AND ("DDoS" OR "Controller Resource Saturation" OR "Bandwidth Saturation" OR "Flow Table Overloading")".
- **Search library:** We selected Google scholar, IEEE Xplore, Springer, Science Direct, Wiley, Hindawi, and ACM as the databases to search the literature. At the same time, we also searched the relevant literature on CNKI and selected articles with higher impact factors.
- **Inclusion/exclusion criteria:** We further reviewed the literature retrieved from the database and established exclusion criteria to eliminate studies that were not relevant to the defined research question. The exclusion criteria are defined as follows:
- Multiple research studies.
- Studies that do not provide an equivalent amount of information.
- The literature without adequate experimental support.
- Not strongly correlated with SDN.

3. Vulnerable Points and DDoS Attacks in SDN

Due to its flexible architecture and non-standardized protocols, SDN has more vulnerabilities in terms of security, leading to a series of new security issues. SDN is not only susceptible to DDoS attacks targeting server devices or services in traditional networks but also to new types of DDoS attacks against switches and controllers, which can cause damage to the network. The security issues and DDoS attacks faced by SDN architecture are shown in Figure 1.

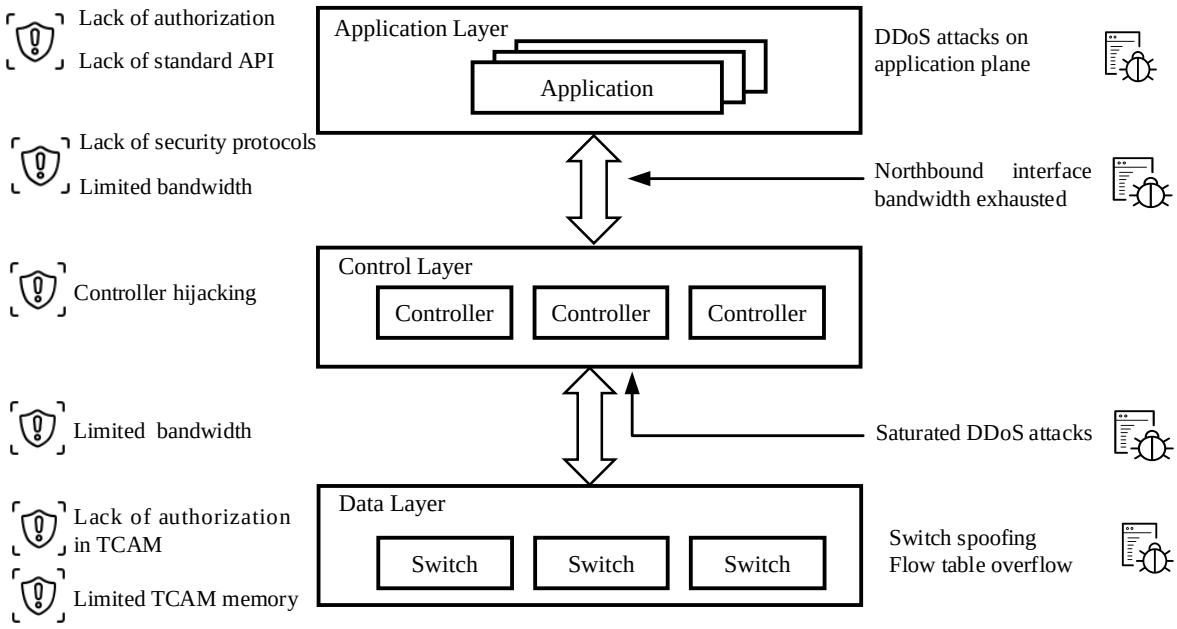

Figure 1. Security problems and DDoS attacks in the SDN architecture.

- The security of the data plane. The limited storage space for flow table entries in data plane switches can lead to overload or buffer overflow when attackers send a large amount of traffic, depleting the computing resources of the control plane. There is a vulnerability in the timeout mechanism of the OpenFlow protocol used for communication between the controller and the switch. Flow table entries are not updated in real-time, so when attackers send false flow table entries to the switch, they continue to be stored in the switch, affecting the normal forwarding of related packets and potentially disrupting the network topology [9].
- The security of the control plane. The control plane controller has network control capabilities. When the controller is hijacked, attackers can use it to carry out network eavesdropping, IP address spoofing, and routing modifications, which can compromise the integrity and confidentiality of the network [8]. A hijacked controller can also send false messages to launch DDoS attacks and deplete network resources.
- The security of the application plane. The application layer defines the functionality of the network controller. However, due to the absence of strict access control mechanisms, attackers can execute malicious programs on the application layer to gain access to network intelligence or deplete resources. Attackers can also target specific applications in SDN systems by sending resource-intensive requests to consume the network bandwidth and disrupt network availability [10].
- The security of communication and protocols. The OpenFlow protocol used in the southbound interface encrypts data using SSL/TLS for secure communication. However, the OpenFlow 1.3.0 specification made TLS optional, which means that communication in the southbound interface may not be secure [11]. Therefore, attackers can intercept or tamper with data packets in southbound communication or exploit the interactive nature of the OpenFlow protocol to launch DDoS attacks and deplete network resources. The absence of standardized protocols in the northbound interface makes data transmission vulnerable to eavesdropping, significantly compromising network confidentiality.

Based on these aforementioned security issues, attackers can exploit vulnerabilities to launch DDoS attacks, which can impact network availability. Since the controller is a core component of the SDN architecture, DDoS attacks targeting SDN controllers have become an important type of DDoS attack [12]. Figure 1 also categorizes DDoS attack types

according to the SDN architecture plane. Table 2 summarizes the characteristics of DDoS attacks in SDN environments.

Table 2. DDoS attack types in the SDN environment.

Attack Type	SDN Plane	Security Vulnerabilities Exploited
Flow Table Overflow	Data Plane	The OpenFlow switch possesses a restricted amount of storage capacity for flow tables.
Switch spoofing	Data Plane	The OpenFlow switch lacks authentication for flow tables.
Saturated DDoS attacks	Control Plane	Packet In datagram blocking controller
Malicious program DDoS attacks	Application plane	The application plane lacks robust authentication and access control mechanisms for applications.
Northbound interface bandwidth exhausted	Application plane	The application layer lacks robust authentication and access control mechanisms for applications, and the northbound interface has limited bandwidth resources [7].

- **Data plane DDoS attacks.** Data plane OpenFlow switches use ternary content-addressable memory (TCAM) to store forwarding rules. TCAM has high storage efficiency but is expensive and has limited space. When there is a need to store forwarding rules for a large amount of traffic, table overflow can occur [13]. On the other hand, OpenFlow switches have a vulnerability in their static timeout policy. The flow rules stored in the switch are only deleted if no matching packets are received within a certain period of time. The Low-Rate Flow Table Overflow (LOFT) attack exploits this vulnerability by sending low-rate attack traffic based on the flow table timeout rules, saturating the switch's flow table entries and preventing the normal forwarding of traffic [14]. Another common data plane attack is switch spoofing [15]. Since data plane switches do not have the ability to identify controller flow tables, attackers can send malicious flow table entries to modify the switch's IP address. When the controller tries to connect to the switch using an IP address, the malicious switch impersonates the IP address and communicates with the controller, causing the controller to lose connection with legitimate switches and disrupt network availability.
- **Control plane DDoS attack.** When a switch processes packets that do not match its flow table entries, it sends a Packet In message to the controller in order to retrieve the corresponding flow table information. Attackers inject a large number of invalid packets, causing the switch to send numerous Packet In messages to the controller. This action consumes controller resources and achieves the goal of saturating the controller with a DDoS attack.
- **Application plane DDoS attacks.** Application plane DDoS attacks exploit the weak access control mechanism of SDN [7]. Applications with design flaws can create a large number of threads, which can consume memory resources or deplete the bandwidth resources of northbound interfaces. Malicious applications can simultaneously consume controller resources by generating a large number of resource-intensive requests. Traditional application plane attacks, such as HTTP Flood and DNS Flood attacks, are also major DDoS attack methods in SDN.

4. DDoS Attack Detection Technology in SDN

4.1. Statistical Analysis-Based DDoS Attack Detection Technology

In SDN environments, effectively identifying DDoS attack behaviors is a crucial prerequisite for issuing timely warnings and successfully implementing defense measures. This is essential for maintaining the normal operation and security of the network. Given that

attackers often use technological means to disguise malicious traffic as legitimate traffic to confuse the public, the precise detection of DDoS attacks faces significant challenges. When a system is subjected to such attacks, the network traffic characteristics typically undergo significant changes. By conducting a comprehensive statistical analysis of these abnormal features, potential DDoS attack activities can be effectively identified. Mainstream statistical analysis and detection methods include, but are not limited to, techniques based on the information entropy theory. These methods reveal hidden attack patterns by quantifying their uncertainty in network traffic. And detection techniques utilize statistical prediction models, which are trained using historical data to predict future traffic conditions. These models serve as a benchmark to identify traffic features that significantly deviate from normal conditions, effectively capturing the occurrence of DDoS attacks. Both of these methods are important for detecting DDoS attacks in the current SDN environment.

4.1.1. Information Entropy-Based DDoS Attack Detection Technology

The information entropy theory and information divergence proposed in the information theory can be used to reflect the uncertainty of information in a system. Information entropy is a method used to measure the probability of a random variable occurring at a specific time. Detection methods based on entropy mainly use different header features of network traffic, such as a source IP address, destination IP address, source port, etc., to calculate the randomness of data packets in the network. In a communication system, communication between hosts is unrelated, and the features of network traffic, such as the destination IP, have a high degree of uncertainty. The characteristics of DDoS attack traffic are that a large number of hosts (or spoofed sources) aggregate malicious traffic to one or a few destination hosts. Under the influence of this malicious traffic, the distribution of source IP addresses and destination IP addresses often deviates from the legitimate pattern, and the calculated entropy value also undergoes significant changes in a short period of time. Finally, by combining intrusion detection, machine learning, and other technologies, it is possible to further determine if the system is under a DDoS attack.

Due to the programmability of SDN controllers, it is possible to extract and analyze network traffic, calculate entropy, and detect DDoS attacks in the network. Yadav et al. [16], Ahalawat et al. [17], and Carvalho et al. [18] utilized Shannon entropy to detect DDoS attacks in SDN environments. These methods collect traffic and select features using SDN controllers or OpenFlow switches. They calculate entropy and determine the presence of DDoS attacks based on a threshold. These methods have high real-time capability, and low resource consumption. However, they suffer from low detection accuracy and are prone to false positives.

To address the issue of the low detection accuracy associated with static threshold detection based on information entropy, Zahra et al. [19] proposed a method that utilizes a dynamic threshold setting in information entropy detection. This method collects entropy values in each period, divides them into normal entropy values and attack entropy value sets based on their relationship with the threshold, and updates the entropy threshold based on the mean and standard deviation of the two sets. Although this method improves accuracy to some extent, setting dynamic thresholds is relatively simple and can still result in false alarms. Future research on dynamically designing threshold methods is one of the hot topics in this field.

Raja et al. [20] proposed a method for detecting DDoS attacks based on generalized entropy, which combines Shannon entropy and Rényi entropy. This method utilizes the Snort intrusion detection system to extract traffic features and calculate the generalized entropy (GE) and generalized information distance (GID) of these features. These measurements are used to determine whether the system is experiencing a DDoS attack. By employing high-order calculations, generalized entropy amplifies the fluctuations in entropy, rendering it more responsive to variations in network traffic. Reference [20] reduced the redundancy of traffic features by calculating the information distance. This approach helps to minimize the overhead of identifying attack packets by the controller. Furthermore,

reducing redundant traffic features also helps improve the accuracy of deep learning when detecting traffic in subsequent sections.

Liu et al. [21] utilized relative entropy to detect DDoS attacks in SDN. Relative entropy, also known as Kullback–Leibler divergence, reflects the differences between two distributions. In SDN, abnormal traffic changes and DDoS attacks can be detected by statistically analyzing the distribution of traffic features and calculating relative entropy with normal or previous traffic feature distributions. For known attacks, calculating relative entropy can enhance detection effectiveness. However, the effectiveness of detection depends on the prior statistical distribution of normal traffic. Therefore, it is necessary to determine the optimal feature set of network traffic in order to improve detection accuracy.

The calculation formula for Shannon entropy primarily focuses on utilizing a singular traffic feature to identify DDoS traffic, while disregarding the potential correlation with other packet features. Reference [22] introduced a DDoS attack model based on joint entropy detection. Joint entropy employs multiple traffic packet header information to compute entropy values, thereby mitigating the occurrence of false alarms that may arise from solely calculating entropy values for the destination IP. Simultaneously selecting various features for the computation of joint entropy has the potential to identify distinct categories of DDoS attacks. For instance, through the exploitation of vulnerabilities in the ICMP protocol and the calculation of joint entropy using attributes such as the packet destination IP, protocol type, destination port, and packet size, it is possible to achieve a more precise identification of the attack's source. Although joint entropy exhibits superior performance in the detection of DDoS attacks, it is accompanied by a higher level of computational complexity. Consequently, it cannot ensure real-time performance within the context of SDN.

Ming et al. [23] proposed a method for detecting DDoS attacks based on conditional entropy. One of the characteristics of DDoS attacks is the convergence of multiple sources targeting a single destination. By utilizing conditional entropy, it is possible to calculate the probability of the correlation between source IP addresses and destination IP addresses, thus enabling the detection of DDoS attacks. Conditional entropy reflects the correlation between traffic characteristics and is effective at identifying malicious traffic. However, computational complexity is correspondingly increased.

Li et al. [24] demonstrated the feasibility of using φ-entropy to detect DDoS attack traffic and proposed a DDoS attack detection scheme based on φ-entropy in SDN networks. This work introduces the parameter φ to adjust the sensitivity of the event frequency measurement. Compared to Shannon entropy, φ-entropy can amplify the correlation between random variables and is able to analyze the traffic correlation effectively in network traffic analysis. The proposed scheme involves the controller periodically obtaining the entropy value of the destination IP address of the data flow and comparing it with a threshold. When the entropy value is less than the threshold for five consecutive periods, it is determined that a DDoS attack is occurring. Through experiments, the authors have demonstrated that φ-entropy is more effective than Shannon entropy in detecting high-intensity DDoS attacks. However, it is necessary to adjust the parameter φ used in the detection according to the network situation.

Table 3 shows a comparison of information entropy-based detection methods. The detection method based on information theory has low algorithmic complexity, which does not impose a heavy burden on the controller and has certain real-time capabilities. However, it also has certain limitations. In the case of high-traffic SDN networks, the detection method based on information entropy has the drawback of high false alarm and missed detection probabilities. Additionally, it does not perform well in detecting low-rate DDoS attacks. In DDoS attack detection, the information entropy-based method can be used as an initial detection scheme, combined with machine learning methods, to form a multi-level detection scheme, thereby enhancing the capability of detection.

Table 3. Comparison of DDoS attack detection parameters based on information entropy.

Calculation Parameters	Features	Strengths	Weaknesses	Improvement Methods
Shannon Entropy	Probability of variation in traffic characteristics	Easy to calculate. Less computing resources	Low detection accuracy	Dynamic threshold adjustment. Joint detection of multiple traffic features
Generalized entropy (GE)	Expansion of Shannon entropy and amplification of the variation in Shannon entropy.	The parameter exhibits a higher level of sensitivity towards variations in traffic characteristics.	When the order of magnitude is high, the computational complexity experiences an increase.	Set different orders for different DDoS attacks
Relative entropy (KL divergence)	Measuring the difference between normal traffic and malicious traffic	High recognition rate for known attacks	Dependent on previous traffic data models	Extract traffic characteristics of different attack types and use relative entropy to detect attack types
Conditional entropy	Reflecting the interrelationships among various attributes of traffic flow	High detection accuracy	The computational time and space complexity are significant, posing challenges in meeting real-time requirements.	Selecting an appropriate conditional entropy detection model for different DDoS attacks
Joint entropy	Using multiple traffic packet header features for entropy calculation	Compared to a single entropy value, the accuracy is elevated. Can detect unknown attacks	More resources are required for computation. Static thresholds are prone to false alarms	Threshold adaptive adjustment Selecting accurate detection features to reduce computational complexity
φ-entropy	Introducing parameters φ Sensitivity of adjusting entropy to probability changes in flow characteristics	Amplified the correlation between traffic, with high sensitivity.	The parameters φ need to be pre-set, and different designs are needed according to the changes in network traffic φ Parameters.	φ Parameter adaptive change

4.1.2. Traffic Statistics-Based DDoS Attack Detection Technology

When a DDoS attack occurs in the network, certain network features may deviate from their normal values. Defenders can select network features based on attack characteristics, analyze changes over a certain period of time, and issue DDoS attack alerts when abnormal features are detected. Additionally, they can establish regression models or time series prediction models based on historical statistical data to predict future traffic changes. This allows for timely alerts to be issued for impending high-traffic behavior in the network [11].

Kalkan et al. [25] proposed a statistical packet filtering model. When the traffic on the switch exceeds the bandwidth threshold, a comparator compares the suspicious traffic characteristics with the configuration file, calculates a matching score, and discards the data packet if the score exceeds the threshold. This approach selects multiple different attributes based on attack traffic to generate various configuration files, resulting in the effective detection of known attacks on switches. Fouladi et al. [26] utilized time series analysis to detect DDoS attacks. This approach statistically analyzes historical traffic change patterns and utilizes ARMA and chaos theory models to forecast future network traffic changes. It also generates alerts in cases of traffic overload. Shohani et al. [27] proposed a statistical prediction detection method for detecting blind DDoS attacks that are difficult to identify. This method utilizes information entropy and principal component analysis techniques. The controller statistically tracks the changes in the number of flow table entries that are not hit when the switch receives normal traffic. It uses the Exponentially

Weighted Moving Average (EWMA) method to establish a trapezoidal detection threshold. When a switch is under a DDoS attack, the number of missed flow entries in the switch exceeds the threshold, thereby detecting the DDoS attack. Although this method has a strong defense effect against blind DDoS attacks, it has a weak detection effect for DDoS attacks originating from a single host.

The label-based statistical analysis method involves adding flow labels to various switch traffic data. It then performs statistical analysis on data flow information within the network to detect DDoS malicious traffic. Furthermore, it can trace malicious traffic by utilizing the labels. Wang et al. [28] utilized the encoding of data packet forwarding paths as parameters for detecting attacks and generating alerts when abnormal traffic is identified on a specific path. This method is suitable not only for detecting DDoS attacks but also for detecting whether there are loops in the data packet forwarding process. Sahay et al. [29] proposed adding Packet in messages to flow-ID labels based on the VLAN ID field. The traffic statistics collector collects the source address, destination address, and flow-ID label of the packets. When the threshold is exceeded, a security alert is issued, and suspicious switches can be traced using the flow-ID label.

The statistical analysis method has a higher detection accuracy compared to the information entropy detection method, but it also requires the collection of a large amount of historical data, which consumes the network's computing and storage resources [30]. The dynamic adjustment of feature selection and thresholds is also a consideration for different DDoS attacks and attack rates.

4.2. Machine Learning-Based DDoS Attack Detection Technology

With the recent advancements in artificial intelligence in various fields, machine learning algorithms have been widely used for pattern recognition, object detection, and classification and regression problems. Machine learning algorithms utilize large amounts of data and expert experience to improve algorithmic strategies and parameters, achieving optimal performance standards for computer programs. In DDoS attack detection, defenders can train machine learning-based traffic classification tools based on historical traffic data to achieve the anomaly detection of network traffic. Commonly used machine learning algorithms for detecting DDoS attack traffic include support vector machines (SVM), the Naive Bayes algorithm, supervised learning algorithms, self-organizing maps (SOMs), and an unsupervised algorithm. Table 4 illustrates the commonly used machine learning methods for DDoS threat detection.

Table 4. Machine learning algorithm for DDoS attack detection.

Algorithm Classification	Algorithm	References
Traditional machine learning	SVM	[31–35]
	Decision Tree	[36–38]
	KNN	[38–41]
	Naive Bayes	[38,42–44]
	Random Forest	[36–38]
Deep learning	SOM	[41,45,46]
	ANN	[47–49]
	LSTM	[48–50]
	DNN	[51–53]
	RNN	[50,53]

The SVM algorithm is a binary classification model utilized for distinguishing between normal and abnormal data in the context of DDoS attack detection based on traffic characteristics. Based on the traffic characteristics observed in the SDN network environment, the SVM detection algorithm is employed to gather input feature vectors in order to develop

an algorithm for detecting malicious behavior within the network. The accuracy of the SVM algorithm is significantly influenced by the traffic feature vectors and kernel functions that are constructed. Kokila et al. [31], Mehr et al. [32], and Ye et al. [33] employed the SVM algorithm for the purpose of detecting DDoS attacks within the SDN environment. By employing various traffic features and kernel functions, the algorithm was able to enhance its detection accuracy. Myint et al. [35] introduced the advanced support vector machine (ASVM) algorithm as a means to enhance the basic binary classification outcomes of conventional SVM algorithms. The objective was to enable the concurrent identification of UDP Flood and SYN Flood attacks. Reference [34] utilizes the One-Class SVM algorithm for the purpose of detecting DDoS attacks. This study focuses on the training of a One-Class SVM model using 11 feature vectors extracted from DDoS attack traffic. Additionally, an adaptive genetic algorithm was employed to optimize the model's parameters, thereby enhancing the accuracy of the detection process.

The KNN algorithm is a supervised learning algorithm that aims to cluster data by identifying the closest neighbors based on data features. In the context of attack detection, this algorithm categorizes network traffic by quantifying the dissimilarity between various feature values. Dong et al. [39] introduced an enhanced KNN algorithm for the identification of DDoS attacks in SDN. In the context of SDN network traffic, it is essential to consider the following four parameters: traffic length, traffic duration, traffic size, and traffic ratio. These parameters play a crucial role in detecting various types of DDoS attacks. To accomplish this, the KNN model was employed. This model demonstrates a remarkable ability to accurately identify DDoS attacks. However, it is important to note that the simulation experiment topology employed in this study is relatively simplistic, and deploying real-time detection in complex, real-world environments pose significant challenges. Latah et al. [40] employed the KNN algorithm in conjunction with other machine learning algorithms for the purpose of network anomaly traffic detection. The experimental results indicate that the KNN algorithm exhibits superior accuracy and incurs a greater time cost in comparison to alternative algorithms.

Machine learning algorithms such as Naive Bayes, decision trees, and random forests are frequently utilized for the purpose of traffic classification. Currently, numerous studies have synthesized these aforementioned machine learning detection methods and have identified the method that yields the most effective detection results. In order to address the issue of data plane Flow Table Overflow attacks, Santos et al. [37] implemented support vector machines, decision trees, and random forest algorithms within controllers to detect and classify traffic. In the experimental setting of this study, it was observed that decision trees exhibit the shortest processing time, whereas random forest algorithms demonstrate the highest level of accuracy. Khashab et al. [38] implemented a model for detecting DDoS attacks based on data flow for the application plane of the SDN architecture. This study employs a combination of Naive Bayes, logistic regression, decision tree, random forest, SVM, and KNN algorithms in order to detect and classify malicious network traffic. Based on empirical investigations, it has been determined that the random forest algorithm outperforms other algorithms in terms of accuracy and real-time performance. Aslam et al. [54] also implemented these six aforementioned algorithms in the context of SDN for the purpose of detecting DDoS attacks. Unlike previous studies on traffic recognition, this method integrates six distinct algorithms to identify and classify traffic, and subsequently determines the presence of malicious traffic by analyzing the outcomes of these six algorithm classifiers. This approach enhances the overall accuracy of the system. Wu et al. [55] employed a factorization machine (FM) algorithm to identify low-rate DDoS attacks on the data plane. In order to address the concealed and challenging-to-identify attributes of low-speed DDoS attack traffic within the data plane, this approach aims to extract four distinct features from the input flow-table rule. These features were then utilized to train the FM algorithm model, taking into account the correlated characteristics of the attack traffic. Finally, an experiment on a low-speed DDoS attack was conducted using the CAIDA dataset. The experiment compared the performance of the FM algorithm-based DDoS attack detection

method with the CNN model and random forest model. The results demonstrate that the FM algorithm-based method achieves a high recognition rate in this specific environment.

4.3. Deep Learning-Based DDoS Attack Detection Technology

Deep learning algorithms are extensively employed in the field of intrusion detection and malicious traffic recognition, primarily because of their inherent advantages, including self-learning capabilities, self-organization, robustness, good fault tolerance, and parallelism [56]. Deep learning-based DDoS attack detection methods exhibit a superior recognition capability for novel DDoS attacks, as they do not necessitate the filtering of traffic features [57]. The primary techniques employed for DDoS detection in deep learning are neural network models.

Cui et al. [58] conducted a study in which they gathered switch traffic through an SDN controller and employed the BPNN(Back Propagation Neural Network) algorithm for the purpose of classifying the traffic and detecting any malicious activity. Simultaneously, by utilizing the classification outcomes, it was possible to track the origin of malicious IP traffic and eliminate it within the switch, thereby achieving the objective of mitigating attacks. Li et al. [49] employed CNN, RNN, and LSTM algorithms for the purpose of detecting traffic features. At the same time, it is imperative to continuously update the deep learning detection model in real-time, taking into consideration the probability of traffic characteristics. The aforementioned methods result in an increased workload on the switch; however, they exhibit low real-time performance. Nam et al. [41] employed the SOM and KNN algorithms to assess the dissimilarity between traffic and malicious traffic feature vectors. Their objective was to identify if the traffic corresponds to a DDoS attack, enhance the real-time detection capability, and minimize the impact on accuracy. Deepa et al. [59] employed a two-level neural network detection model. Initially, they utilized a deep belief network (DBN) and autoencoder (AE) algorithms to extract attack traffic features. Subsequently, the multiple kernel learning (MKL) algorithm was employed for traffic classification with the aim of identifying DDoS traffic while maintaining a balance between accuracy and efficiency.

The advantage of a machine learning detection mechanism lies in its strong ability to abstract and generalize detection data with high feature dimensions, allowing for the efficient processing of multi-dimensional data. Machine learning models [60–62] can also yield favorable results in recognizing attack types, reducing the dimensionality of traffic data, and tracing attackers. The drawback is that the characteristics of supervised learning algorithms are manually designed and annotated, and the quality of input features significantly affects the detection accuracy of the model. Unsupervised learning algorithms require additional time and resources to train the model, leading to subpar real-time detection.

4.4. Hybrid Detection Technology

While information entropy or the machine learning anomaly detection algorithm can identify DDoS attacks in SDN networks, accurately characterizing extensive data in SDN networks using only information entropy algorithms is challenging. Relying solely on machine learning algorithms also consumes excessive time and resources, posing difficulties in ensuring real-time detection.

Currently, a mature approach is to combine information entropy and machine learning in a hybrid detection model. This involves using information entropy methods for initial detection to identify early attack behaviors or locate attacks, followed by machine learning methods for further detection. Hu et al. [63] proposed a hybrid detection method that combines information entropy and machine learning. This method detects changes in the information entropy of SDN controllers, extracts information entropy as a feature, and uses the SVM algorithm for traffic classification, effectively identifying DDoS traffic. Sun et al. [64] tackled the problem of low accuracy in information entropy by employing the computation of traffic source IP and destination IP φ-entropy. φ-entropy is utilized for the

initial detection of DDoS attacks, and the detection module differentiates normal traffic from malicious traffic using the KNN algorithm to identify malicious DDoS attacks. Novaes et al. [65] proposed a multi-level detection method that divides attack detection into three stages. Firstly, network packet attributes are collected, and information entropy along with other features are calculated. LSTM models are then utilized to predict traffic changes and detect early attack behaviors. Finally, fuzzy logic is utilized to further identify and pinpoint attacks. Dehkordi et al. [66] proposed a hybrid detection method that identifies and locates abnormal traffic using information entropy and statistical analysis methods. In the detection module, a variety of machine learning algorithms, including Bayes Net, J48, logistic regression, and Random Tree algorithms, were utilized for classification to address the high false alarm rate associated with the dynamic threshold. Zhang et al. [67] proposed a multi-level mixed detection method. It initially utilizes information entropy to rapidly detect changes in network traffic, followed by the use of the autoencoder (SSAE)-SVM architecture multi-level detection model to identify abnormal traffic. This approach enhances detection timeliness while reducing the probability of false alarms.

Hybrid detection technology ensures that the model has high detection accuracy while also reducing the processing time for classification detection. However, due to the complexity of the model design, multiple functional modules need to be expanded on the controller, and the deployment and maintenance costs require further research and optimization. Furthermore, the multi-level detection mechanism may increase time and computational resource costs.

5. DDoS Attack Mitigation Techniques in SDN

5.1. Policy-Based DDoS Attack Mitigation Techniques

Implementing forwarding policies in switches and controllers, as well as controlling traffic forwarding, filtering, dropping, rate limiting, and redirecting packets, are widely employed techniques for mitigating DDoS attacks. Figure 2 illustrates commonly used policy-based DDoS attack mitigation methods.

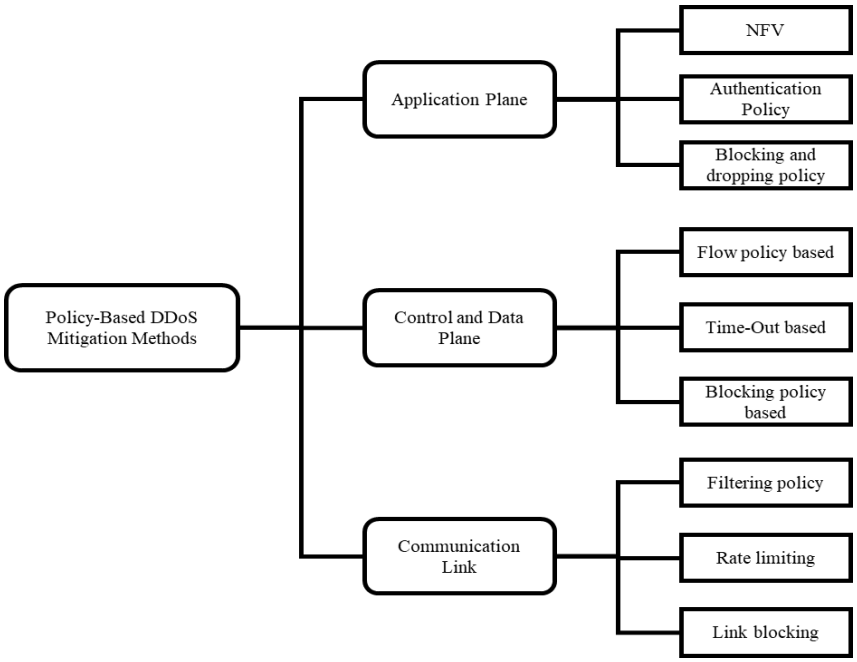

Figure 2. The commonly used policy-based DDoS attack mitigation methods.

The SDN application plane is primarily responsible for implementing various forms of network business logic and strategies. Defenders can mitigate DDoS attacks by implementing authentication policies, conducting traffic monitoring analysis, utilizing NFV, and employing other methods. Singh et al. [68] proposed a scheme called ARDefense for

detecting and mitigating DDoS attacks based on NFV and SDN. This approach leverages NFV technology, server migration policies, and IP spoofing techniques to mitigate application layer DDoS attacks. Ali et al. [69] proposed an intrusion prevention system based on three-layer authentication, which includes user authentication, packet authentication, and flow authentication. Packets that cannot be authenticated are refused for forwarding. This approach enhances the defense against DDoS attacks by employing authentication techniques. However, multi-level authentication also has an impact on network performance. Sarwar et al. [70] proposed a traffic forwarding method based on user trust. This method establishes user trust and directs traffic in the queue according to the level of user trust, while discarding unauthorized user traffic.

The SDN controller establishes traffic regulations to prevent the storage and forwarding of malicious traffic within the network. Deng [71] presents a scheme that suggests the implementation of a DoS attack defense method in the controller, utilizing address matching. This method involves the extraction of MAC, IP, and port information from the Packet-In-Packet received by the controller. Subsequently, this information is compared with the network device information. If there is any discrepancy in the information, the packet is discarded. Ravi et al. [72] implemented a traffic control scheme in controllers that relied on a blacklist approach. This method is designed to counter SYN Flood attackers who employ fraudulent IP addresses to initiate their attacks. By performing IP packet analysis, this method detects instances of IP-MAC address spoofing within the network. It then proceeds to blacklist any identified illegal addresses and subsequently discards associated data packets. Cao et al. [73] conducted a study that specifically examined the characteristics of DDoS traffic forwarding. In their research, they employed an RNN model to effectively identify links within the network associated with attack flows. The controller made decisions regarding traffic forwarding by considering factors such as the IP address, hop count, and the router it passed through, while also discarding any malicious traffic.

SDN data plane Flow Table Overflow DDoS attacks inject traffic slowly into OpenFlow switches, mimicking the characteristics of legitimate users. This results in false positives when using information entropy and machine learning methods to detect such attacks [74]. Bawany et al. [75] implemented an adaptive filtering strategy based on flow rules, which defines three filtering strategies according to network traffic. According to the size and rate of the attack traffic, strategies such as dropping packets, blocking ports, and redirecting data flow were selected to achieve adaptive DDoS attack mitigation. Yuan et al. [76] introduced a peer-to-peer support strategy. When a switch requests a new policy from the controller, the status monitor module facilitates the transfer of the flow to other peer switches. This transfer is based on various parameters, including switch TCAM usage, distance from other switches, and switch busyness. The purpose of this transfer is to alleviate switch storage pressure. Bhushan et al. [77] introduced a flow table space model that is grounded in the queuing theory. When the available space in the flow table of a switch is inadequate, the queuing model is employed to transfer the corresponding flow policies to a switch with sufficient space. This process involves deleting low utilization policies to prevent overflow in the flow table. Katta et al. [78] conducted an optimization study on the storage strategy of switch flow tables. Dang et al. [79] mitigated DDoS attacks by implementing timeout policies. When the network controller detects a high volume of TCP semi-connected packets, it adjusts the timeout rule according to changes in network traffic, promptly discards semi-connected packets in the flow table, and implements a blacklist mechanism to reject malicious traffic packets. They achieved the objective of expanding the storage space by reducing unnecessary storage content [80], taking into consideration the dependency relationship among flow table items [81].

Attackers typically launch DDoS attacks on links in the SDN control plane and data plane, disrupting normal traffic forwarding. Zakaria et al. [82] proposed a rate-limiting strategy for mitigating reflective DDoS attacks. This strategy employs statistical analysis and machine learning methods to identify malicious traffic characteristics and establish rate-limiting policies to reduce the forwarding of packets with malicious traffic characteristics.

Hong et al. [83] proposed a dynamic routing defense strategy that utilizes information entropy and a dynamic threshold to detect and locate abnormal traffic hosts in the network. The unusual host traffic is redistributed based on the network channel's capacity to reduce the link congestion caused by individual target DDoS attacks. Kalkan et al. [25] proposed a traffic filtering strategy called SDNScore for mitigating link DDoS attacks. This strategy utilizes statistical methods to analyze the characteristics of traffic packets, such as IP, port, and TTL and assesses the similarity of traffic. The controller filters the attack traffic based on its traffic score. Alamri et al. [84] proposed a bandwidth limitation algorithm. In this algorithm, the SDN controller detects when the traffic on a specific link exceeds a threshold, dynamically adjusts the traffic limit of the link using a bandwidth adjustment factor, and activates a traffic detection module based on the XGBoost algorithm to identify malicious traffic. Wang et al. [85] developed a global search algorithm based on an SDN controller to detect and identify congested links to link Flooding attacks. To manage congested links, methods such as blocking and discarding abnormal data packets are employed to protect against DDoS attacks.

Policy-based methods can be easily implemented on the network and have a negligible effect on resources. However, this approach is susceptible to inducing regular traffic loss and necessitates the accurate identification of malicious traffic through detection techniques. Defenders are required to establish forwarding policies that are tailored to the specific characteristics of network attacks, with the aim of rejecting any malicious traffic. Due to the implementation of SDN network policies, it is observed that there are still instances of malicious flow table entries being generated by attack flows within the switch. In addition to restricting traffic forwarding, it is imperative to cleanse the entries in the switch flow [27].

5.2. Moving Target Defense Technology

With the continuous evolution of attacker–attack methods, traditional network defense methods such as blocking and killing are becoming increasingly ineffective in achieving real-time defense. Additionally, these methods have clear shortcomings when it comes to dealing with DDoS attacks. Moving target defense (MTD), as an emerging network security defense strategy, has changed the passive nature of defense in traditional network attacks and often achieves positive outcomes when responding to such attacks [86]. Due to network programmability and the centralized control of logic, SDN can easily deploy moving target defense strategies to cope with DDoS attacks. The end information jump strategy enables the dynamic change in host information in the SDN data plane and provides protection against DDoS attacks in the data plane.

5.2.1. Port Address Hopping-Based Defense Technology

The concept of Port Address Hopping (PAH) defense technology originated from the APOD project of the US military in 2003 [87]. Shi et al. [88] introduced the effectiveness of end information hop in defending against DoS attacks and established a mixed hop communication network using the Java language. This network includes features such as the port, address, service time slot, and encryption algorithm. The availability of the network was verified by simulating SYN Flood attacks. By comparing this with non-jump systems, the effectiveness of end information jump in resisting DoS attacks is demonstrated.

Port hopping is a relatively simple and effective method for deploying end-to-end information hopping. Badishi et al. [89] proposed a port-hopping protocol that filters packets based on packet addresses and port numbers to mitigate DoS attacks. Zhang et al. [90] proposed a port hopping scheme called PH-DM, which was implemented in SDN controllers to enable random port hopping in communication. The synchronization between the sender and receiver was achieved through a timestamp feedback-based synchronization method, enabling smooth communication. The MASON framework, proposed by Chowdhary et al. [91], first performed threat scoring on the system to identify high-risk services and hosts. It then deployed port-hopping strategies. This plan was highly targeted and had a strong defense against vulnerable SDN protection devices. Compared to blind jumps, it

has a smaller impact on the network. However, there is insufficient consideration given to the security of jump synchronization, and further design is needed for the evaluation method of network threats. Zhao et al. [92] proposed an encryption strategy that enhanced the Diffie Hellman algorithm by incorporating port hopping. This approach ensures the randomness of the hopping port and the confidentiality of the synchronization process, effectively guaranteeing the security and confidentiality of the SDN network.

Port hopping does not require modifying existing protocols, but strict synchronization rules need to be established between the sender and the receiver. Establishing synchronization rules poses a challenge for implementing hopping methods. Based on the characteristics of centralized control in SDN, deploying port hopping rules in SDN controllers can enhance network security. When attackers launch DDoS attacks against specific ports, it can be challenging to provide effective protection. At the same time, this strategy also has a defensive effect on port scanning by attackers. However, if the security of the jump strategy design is insufficient, attackers can obtain jump rules based on multiple information collections, rendering the jump strategy ineffective. When deploying port hop rules in SDN, it is also necessary to consider how port hop traffic can smoothly pass through firewalls and address other related issues.

Port hopping, to a certain degree, mitigates DDoS attacks targeting specific ports. However, attackers can still execute DDoS attacks by crafting packets originating from the IP address of the target. In order to bolster the system's defense capabilities against intricate threats posed by attackers, the concept of address hopping is put forth. Under the address jump rule, both communication parties modify their IP address information in accordance with predetermined rules. Only data containing accurate IP address information can be transmitted to the designated destination address, thereby providing an effective defense against external DDoS attacks. Reference [93] introduced the concept of hybrid network address hopping in computer networks as a means to augment the security of data transmission. By conducting computer simulations of the NAH system, it was confirmed that this system exhibits superior anti-interference capabilities and ensures enhanced confidentiality during the transmission of network data. Taking advantage of the independent control plane offered by SDN, the address jump strategy can be implemented within SDN controllers. Zheng et al. [94] proposed a scheme for address hopping in SDN, where the flow table entries of the hop IP are allocated to the OpenFlow switch via an SDN controller. The switch verifies the accuracy of the system's message transmission by confirming the source IP and destination IP of the message and subsequently executes matching forwarding. When the packet traffic of a specific address surpasses a predetermined threshold, it initiates the next address jump, prompting the controller to reallocate the flow table information in order to effectively evade potential attacks. Tu et al. [95] proposed a novel address jump scheme for chaotic sequences. This scheme utilizes chaotic sequences as the foundation for generating address jump patterns, effectively addressing the issue of vulnerability to static jump rule cracking. Reference [96] implemented the address jump rule on SDN switches and terminal nodes, resulting in a reduction in the controller load and network overhead. At present, the integration of address hopping with deep learning algorithms enables the attainment of adaptive hopping. Reference [97] suggests the utilization of CNN detectors for the purpose of detecting attacker behavior and promptly initiating address jumps. Compared to conventional jump rules, this scheme demonstrates a higher level of specificity and the ability to incorporate attacker behavior in order to dynamically adapt the jumping process. Consequently, this approach effectively mitigates the system overhead resulting from address jumps. Reference [98] presents an SDN address hopping algorithm that utilizes flow counting synchronization to achieve adaptive address hopping according to network traffic patterns. At the same time, the security of the transmitted information is guaranteed by employing RSA verification, thereby enhancing the system's resilience against DDoS attacks.

Deploying address-hopping strategies in SDN can provide a certain level of defense against DDoS attacks. However, it remains challenging to accurately differentiate between

legitimate and malicious traffic, as well as guarantee the normal forwarding of traffic. In the context of address jump rules, the magnitude of the address space also plays a role in determining the efficacy of defense mechanisms. For certain targeted network attacks, the implementation of address-hopping strategies can be employed in conjunction with honeypots or intrusion detection devices to not only achieve defensive effects but also facilitate the deception and tracing of attackers [99].

Hybrid hopping integrates various information mutation techniques to enhance defensive efficacy. Shi et al. [100] introduced a novel active network defense technology that utilizes mixed-end information hopping. This approach enables simultaneous port and address hopping during communication while ensuring information synchronization through an end information-expanding synchronization strategy. As a result, high-speed Port Address Hopping is achieved. This method guarantees the availability of the system, even when faced with multiple DDoS attacks. Hu et al. [101] introduced a novel moving target defense scheme that utilizes the OpenFlow protocol. The proposed scheme involves the dynamic alteration of IP addresses at each hop of the OpenFlow switch, the implementation of port hopping in inter-domain networks, and the synchronization of information through a dedicated synchronization server. This methodology can be implemented not only in SDN environments but also in conventional networks that are equipped with OpenFlow switches. This method demonstrates a robust defense mechanism against DDoS attacks targeting specific nodes while also offering a relatively straightforward deployment process. The AEH-MTD technology proposed in reference [102] employs the entropy method to identify various types of DDoS attacks. It determines the information jump period by assessing whether the attacker is engaged in blind attacks. This approach ensures optimal defense effectiveness while minimizing the impact of jump rules on system availability. Additionally, it aims to limit the attacker's access to useful information, thereby providing a robust defense against diverse attack types.

The comparative analysis for the mitigation approaches based on hopping technology is shown in Table 5. Given the varied manifestations and high level of obfuscation associated with DDoS attacks, it is insufficient to rely solely on end information hopping as a means of achieving optimal defense effectiveness. Simultaneously, the implementation of port address information hop technology necessitates the development of robust synchronization techniques by defenders in order to ensure uninterrupted network availability [103]. In order to enhance the protection against security threats posed by attackers, it is imperative to integrate the end-to-end information hop strategy with other dynamic defense strategies. This can be achieved by leveraging cutting-edge technology to devise and implement a robust architecture and information space. By continuously altering the attack surface, the attacker's assault maneuvers can be rendered ineffective and exposed, thereby accomplishing the objective of countering DDoS attacks and upholding system security.

Table 5. Comparative analysis of network hopping technology.

Tactics	Strengths	Weaknesses	References
Port hopping	No protocol modification required Simple deployment	Poor DDoS attack defense capability easy to discover hopping rule	[89–92]
IP address hopping	No protocol modification required Good DDoS attack defense capability	Implementation is relatively complex Small hopping address space	[94–98]
Hybrid hopping	Strong security difficult to discover hopping rule	Difficulty in deployment High deployment cost Terminal time synchronization issue	[100–102]
Routing hopping	Can defend against link layer DDoS attacks	Protocol modification required implementation complexity impact on network availability	[104,105]

5.2.2. Other Moving Target Defense Technology

A crossfire DDoS attack is an emerging form of cyber-attack that specifically targets crucial network links. Traditional defense mechanisms, such as host information hop strategies, have proven to be ineffective in countering this type of attack. The routing reconstruction strategy mitigates DDoS attacks against links by adjusting the link structure. Xie et al. [104] introduced a dynamic routing jump defense strategy as a response to crossfire attacks. This scheme was implemented in SDN controllers, and it facilitated the reconfiguration of routing for a specific link in order to mitigate link attacks when abnormal traffic was detected. Liu et al. [105] introduced a routing hop strategy that utilized the OpenFlow protocol. This scheme aims to establish a matrix of traffic characteristic entropy by collecting network traffic data in order to detect and identify network anomalies. This triggers route mutation based on the results of anomaly detection. It utilizes an enhanced ant colony algorithm for the purpose of generating novel routing paths, thereby achieving the capability to withstand DDoS attacks. This scheme employs a jump strategy that relies on detecting abnormal traffic conditions. However, it is important to note that a short reconstruction cycle can also have an impact on network availability. Route reconstruction serves as a preventive measure against the expansion of the attackers' attack range by leveraging known routes. Additionally, they possess a defensive capability against link-layer DDoS attacks. Traffic redirection technology protects the target host by diverting the attack traffic. Hyder et al. [106] proposed a moving target defense technology scheme based on traffic redirection for crossfire DDoS attacks. This scheme utilizes NFV technology to redirect the traffic of the attacked link to the shadow host, achieving the goal of mitigating the attack.

In light of DDoS attacks in SDN, a moving target defense approach rooted in game theory is employed to determine the most effective defense strategy for the defending party, taking into consideration the prevailing network conditions, with the aim of achieving equilibrium. Chowdhary et al. [107] proposed a model that conceptualizes DDoS attacks as a dynamic game process involving both attackers and defenders. The researchers devised defense rules, as well as reward and punishment mechanisms, with the aim of identifying the optimal strategy for minimizing network bandwidth consumption and mitigating the impact of DDoS attacks. This scheme is dependent on the utilization of Snort for the purpose of intruder detection. However, it was observed that attackers have the ability to circumvent the intrusion detection system by adhering to specific rules, thereby rendering defense strategies ineffective. Zhou et al. [108] aimed to mitigate the issue of high cost associated with MTD defense methods by proposing the utilization of multi-objective Markov decision processes for the development of MTD strategies. This scheme not only takes into account network attackers and defenders but also integrates legitimate users into the game process, achieving an optimal balance between the cost and benefit of the shuffle-based MTD strategy. This particular game model necessitates early training in order to attain convergence, heavily relies on pre-existing knowledge, and is unable to achieve optimal defense against novel DDoS attacks. Du et al. [109] applied the game theory to enhance honeypot-based DDoS attack defense technology. This article first proposes a two-fold honeypot strategy for SDN based on the game theory from the perspective of attackers. The defender sets up a pseudo honeypot game to lure attackers, constantly adapts the pseudo honeypot to protect against FTP flow and SYN Flood attacks in the network, and strikes a balance between resource usage and defense effectiveness. Priyadarsini et al. [110] designed a trust value controller attack detection (TCAD) model based on the signal game theory. This model constructs trust values based on changes in switch traffic, distinguishes normal users and attackers based on trust values, and achieves the goal of detecting and mitigating DDoS attacks.

The game theory model does not propose novel defense measures against DDoS attacks; rather, it emphasizes the importance of striking a balance between the costs and benefits associated with existing DDoS defense strategies. The game process is dependent on the modeling of past attackers' behavior, and the effectiveness of defense against new

DDoS attacks is suboptimal. Currently, the predominant moving target defense strategies in game-based scenarios are primarily static in nature. These strategies fail to fully account for the dynamic and multi-stage nature of the attack and defense confrontation between attackers and defenders. Deploying real-world networks does not allow for the attainment of optimal decisions in the context of multi-stage network attacks.

The relevant comparisons of moving target defense technologies are presented in Table 6. Moving target defense technology through constantly changing protection strategies or system configurations can make it difficult for attackers to identify effective target points and focus their firepower on saturation attacks against a single target. However, the requirement for the real-time monitoring of network conditions and the need to respond swiftly to changes in moving target defense technology result in increased computational resource consumption and a decrease in overall network performance. Currently, there is no standardized and mature solution for moving target defense technology, and the synchronization issue of network devices is a significant challenge that impacts the application of this technology.

Table 6. Comparative analysis of the moving target defense technology.

Tactics	Strengths	Weaknesses	References
Port address hopping	No protocol modification required Simple deployment	Poor DDoS attack defense capability Easy to discover hopping rule	[89–102]
Routing reconstruction	Can defend against link layer DDoS attacks	Protocol modification required implementation complexity impact on network availability	[104,105]
Shadow host/Honeypot	Can identify attack types Traceable attacker	Possible identification by attackers	[106,109]
Game theory	Game strategy can deceive attackers and diminish attack effectiveness.	An attack model needs to be developed. It needs to be used in conjunction with other defense strategies.	[107–110]

6. Experiment Environment Analysis of the Literature

In the literature reviewed above, significant progress has been made in technical methods and theory. However, there are notable variations in the construction and standardization of experimental environments. For example, some studies are based on single-controller architecture SDN simulation experiments, while others extend the experimental environment to multi-controller SDN. There are also significant differences in the selection of traffic generation tools among different studies, ranging from Scapy and Hping3 to customized Botnet simulators, etc. Additionally, there is diversity in modeling attack scenarios, setting network size, and selecting performance indicators. This section analyzes the common experimental environments and existing problems from the perspectives of simulators and controllers, DDoS traffic generators, and the datasets used in these literature experiments.

6.1. SDN Simulator and Controller

In terms of selecting a simulator and experimental environment, over 90% of the literature opts for Mininet as the experimental platform. Mininet is a lightweight network virtualization tool that leverages the namespaces provided by the Linux kernel, virtual Ethernet devices, and Open vSwitch technology to create a comprehensive SDN environment. It is suitable for rapidly constructing and testing SDN as well as deploying associated applications. The literature [69] uses OMNeT++ as a simulation tool. Compared to Mininet, OMNeT++ has the ability to redefine network layers and protocols, making it suitable for complex network models and capable of handling large-scale and detailed network

simulations. Meanwhile, some works in the literature [42,47] integrate SDN with IoT to assess the effectiveness of its detection and mitigation methods in the IoT environment.

In the selection of experimental controllers shown in Table 7, most literature uses Pox and Ryu controllers to implement their defense schemes. Both controllers are written in the Python language, which is highly flexible and easy to develop and expand. Compared to other large-scale controller projects, Ryu and Pox have lower resource consumption and are more suitable for research and small-scale deployment scenarios. However, these two controllers require additional custom development in the experiment. Meanwhile, there may be faults when handling large-scale, high-concurrency traffic.

Table 7. Classification of some reviewed articles based on the utilized controllers.

Experiment Controller	Detection Techniques			Mitigation Techniques		Literature Proportion
	Statistical Analysis	Machine Learning and Deep Learning	Hybrid Detection	Policy-Based	Moving Target Defense	
Ryu	[17,20,22,27,29]	[33,36,37,41,43,46,52,55,58]	-	[72,80,84]	[97]	30%
Pox	[16,18,19,23,26]	[40,46,49,51,57,59]	[60]	[70,74,77]	[85]	26.7%
Floodlight	[24,28,71,79,81]	[45–48,51,52]	[66–68]	[82]	-	23.3%
OpenDaylight	[21]	[46]	-	[83]	[91,96,108,110]	13.3%
ONOS	-	[34,46]	-	[75]	[75]	5%
NOX	-	[46]	-	-	-	1.7%

The Floodlight controller is an open-source OpenFlow SDN controller licensed under Apache that supports the Java language. It boasts strong cross-platform capabilities, efficient memory management, and concurrent processing abilities, making it suitable for large-scale network environments and high-concurrency traffic scenarios. At the same time, it supports networks consisting of OpenFlow switches and non-OpenFlow switches.

OpenDaylight is a substantial open-source project overseen by the Linux Foundation. It offers comprehensive southbound and northbound interfaces designed for intricate, large-scale SDN network environments. As indicated in the table, the literature on moving target defense technology predominantly utilizes OpenDaylight as the experimental controller.

The ONOS controller is a highly modular SDN controller written in Java, with high availability and large-scale deployment capabilities. It performs well in managing large volumes of data and handling concurrent requests in large-scale SDN networks. At the same time, the controller encounters challenges in deployment, high resource consumption, and low flexibility, which restrict its scalability in experimental scenarios.

NOX is a well-known early SDN controller written in the C++ language, known for its high operational efficiency. However, it has since been replaced by other next-generation controllers, which may have limitations in terms of functionality and usability.

In terms of controller architecture, most experimental environments that utilize machine learning methods adopt a single-controller SDN environment. These methods primarily rely on the controller to detect the traffic of the corresponding switch without taking into account the impact of a single point of switch failure on DDoS attack detection. Due to the necessity of perceiving the network environment and adapting to dynamic changes in network deployment, defense strategies and moving target defense methods against link attacks are frequently evaluated in a multi-controller architecture environment.

6.2. DDoS Traffic Generation Tools and Datasets

In order to accurately evaluate and optimize the DDoS defense mechanism in SDN architecture, network traffic generation tools are widely used to simulate DDoS attack scenarios, verify the effectiveness of detection algorithms, and test the performance of defense systems. Table 8 presents the primary traffic generation tools identified in the relevant literature research.

Table 8. Classification of some reviewed articles based on the traffic generation tools.

Traffic Simulator	Description	Research Works
Scapy	Scapy is an interactive packet processing program that allows users to build, send, receive, and parse network protocol packets at the underlying level.	[18–20,24,74,82,83]
Hping3	Hping3 is a command line TCP/IP packet assembly/testing tool that provides richer functionality than traditional ping.	[21,33,37,47,64,67]
D-ITG	D-ITG is a high-performance network traffic generation tool that can generate complex network traffic with multiple streams and protocols and can simulate traffic loads in high-concurrency scenarios.	[52,55]
BotNet simulator	As a zombie network simulator, it can simulate the attack behavior of a large number of controlled nodes and simulate real distributed attack scenarios	[29,60]
TFN2K	The early distributed denial of service attack tools were used to analyze the behavior patterns and attack mechanisms of attackers.	[60]

Scapy is an interactive packet processing library based on Python. It offers a high level of flexibility for constructing, sending, receiving, and parsing packets of various network protocols. In DDoS attack simulation scenarios, Scapy can be used to meticulously design and execute complex attack traffic models to assess the effectiveness of target systems or defense mechanisms.

Hping3 is a robust command line network tool that allows for the extensive manipulation of various aspects of the TCP/IP protocol stack. It is used to generate and send customized network traffic, mimicking common techniques in DDoS attacks, such as TCP SYN Flooding, and can be utilized for security testing and auditing.

As an advanced network performance testing tool, D-ITG functions to generate a large amount of multi-protocol and multi-mode real network traffic. In a compliant security experimental environment, the main purpose of this tool is to measure network performance and service quality. It achieves this by configuring high-load traffic with DDoS characteristics, which helps users evaluate the resilience of network devices and protection systems.

The BotNet simulator is primarily used to simulate the behavior of zombie networks in a legal and controllable manner. It can simulate a large number of network requests initiated by concurrent nodes, reproduce large-scale DDoS attack scenarios, and provide researchers with an important platform to understand zombie network attack mechanisms, propagation strategies, and test defense measures.

TFN2K was an illegal DDoS attack tool in the early days, showcasing the technical features of early distributed denial of service attacks. In today's research environment, it is feasible to apply its principles to develop a credible simulator. This tool-assisted academic researchers and network security experts in analyzing historical attack methods and refining modern defense technologies accordingly.

Traffic generation tools simulate DDoS attack behavior by creating a large number of packets of the same type. Meanwhile, low-speed DDoS attacks can be simulated by adjusting the rate. However, the features of traffic generation tools are predetermined and lack variability, which makes it challenging to accurately represent the complex traffic characteristics of the network using traffic generation tools and simulators. A recommended method involves gathering actual network traffic and blending it in proportion with malicious traffic generated by traffic generation tools to assess the efficacy of DDoS tool detection and mitigation techniques.

In machine learning-based DDoS attack detection schemes, high-quality datasets are the cornerstone for constructing and validating detection models. Table 9 lists the DDoS attack datasets commonly used in relevant literature research.

Table 9. Classification of some reviewed articles based on the utilized dataset.

Dataset	Description	Research Works
CIC-DDoS 2019	Data containing normal traffic and multiple types of DDoS attacks provides a simulation of DDoS attack scenarios in modern data center environments.	[32,58,65,72,84]
CAIDA	The dataset includes anonymized packet-level records, stream-level data, and real-time or historical BGP routing information for network measurement, topology analysis, and security research.	[52,57,73,79]
NSL-KDD	A preprocessed classic dataset containing four types of network attacks and normal traffic is used to evaluate the performance of intrusion detection systems.	[39,48,52]
CIC-IDS-2017	Contains a large amount of data that simulates different types of attacks and normal traffic in real network environments, suitable for the development and testing of machine learning-based intrusion detection systems.	[31,36,54]
ISCX	A dataset of various types of network attack traffic, including mixed attacks and normal traffic, supporting research on new attack techniques.	[56,66,75]
DARPA	Datasets from the Early Large Intrusion Detection Project "Intrusion Detection System Evaluation" of the US Defense Advanced Research Projects Agency	[42]
UNSW-NB15	Contains data for 9 types of attacks and normal traffic, characterized by rich features and diverse types of attacks	[59]
CTU-13	Provided PCAP format network traffic data for a range of malicious software activities, especially Botnet	[66]
MAWI Working Group Traffic Archive	Public large-scale network traffic data archiving is mainly used for research in network engineering, transmission protocol analysis, and traffic modeling.	[26]
Kaggle	DDoS and other network attack datasets contributed by the cybersecurity community	[35]
LLS 2.0 DDoS dataset	This dataset is specifically designed for DDoS attack scenarios and contains DDoS attack traffic samples of different scales and complexities.	[23]

The CIC-DDoS 2019 dataset was released by the University of New Brunswick in Canada for research on DDoS attacks. The dataset contains a substantial number of labeled data points that differentiate between normal network traffic and various DDoS attack traffic. This provides researchers with an experimental environment featuring the latest attack patterns and defense challenges. This dataset highlights the significance of accurately identifying DDoS attacks in intricate network environments, and its feature set may include detailed packet inspection (DPI) level information.

CAIDA is a significant resource center for Internet traffic research. The platform offers a wide range of public Internet traffic datasets, such as anonymous packet-level data, route table snapshots, and instances of large-scale DDoS attacks. These data provide valuable information for the academic and industrial communities to enhance network traffic models, conduct research on DDoS defense strategies, and analyze the security of network infrastructure.

NSL-KDD is a preprocessed version of the KDD Cup 1999 dataset, primarily utilized for research on intrusion detection systems. Although it mainly focuses on general types of intrusion behavior rather than specifically targeting DDoS attacks, it does contain a small number of DDoS-related samples that can be used to train and test network intrusion detection algorithms.

The CIC-IDS-2017 dataset, released by the University of Carleton in Canada, is an intrusion detection dataset based on real network traffic. It includes various types of network attacks, such as DDoS attacks. This updated dataset aims to represent the current threat landscape in modern network environments and serve as an experimental platform for the latest security research.

ISCX series is a collection of intrusion detection system datasets created by the Information Security and Cryptography Laboratory at the University of Ottawa, Canada. The

ISCX IDS 2012 dataset contains a substantial volume of both normal and abnormal traffic records, making it suitable for DDoS attack detection and other network attack research.

The intrusion detection and evaluation project by DARPA has generated a series of significant datasets to facilitate research competitions in the field of network security. These datasets contain various types of network attacks, including early instances of DDoS attacks, which are highly significant for understanding the development of DDoS attacks.

The UNSW-NB15 dataset was created by the University of New South Wales in Australia. It includes detailed feature descriptions and covers a wide range of attack categories, including various types of DDoS attacks. It is currently widely used as a benchmark dataset in the fields of network intrusion detection and DDoS research.

The CTU-13 dataset from the Prague University of Technology in the Czech Republic focuses on botnet activities and covers various DDoS attack scenarios. This dataset offers samples of malicious traffic generated in real network environments, making it particularly valuable for in-depth research on the sources and propagation methods of DDoS attacks.

The MAWI Working Group Traffic Archive collects real-time network traffic data on the Japanese Internet backbone. These data are valuable for researchers studying large-scale network behaviors, such as pattern recognition and traffic characteristics analysis of DDoS attacks.

As the world's largest data science competition platform, Kaggle frequently releases datasets related to cybersecurity and machine learning in collaboration with industry partners. It contains real datasets focused on DDoS attacks.

The LLS 2.0 DDoS dataset is provided by MIT Lincoln Laboratory and is specifically designed for detecting DDoS attacks. It provides simulated or real DDoS attack traffic data for training and testing the effectiveness of DDoS defense systems. This type of dataset helps researchers better simulate real-world attack scenarios when developing effective defense mechanisms.

Although the aforementioned datasets have yielded favorable results in training DDoS attack detection models, the continuous evolution of DDoS attacks means that early datasets may not capture the latest network attack technologies and trends. Simulation-based datasets may deviate from real-world scenarios. Furthermore, various datasets offer varying feature dimensions and depths, thereby complicating the process of feature selection and processing in machine learning models. Many datasets are not designed for SDN and do not accurately reflect the traffic characteristics in real SDN environments. Further experiments are still needed in real-world network scenarios.

7. Research Challenges and Gap

In this paper, we aim to examine the existing research literature on detection and mitigation technologies against DDoS attacks in SDN environments. It categorizes and reviews these methods based on the technologies employed in the literature. The comparison of the application scope, advantages, and disadvantages of the detection and mitigation technologies mentioned in this article can be found in Table 10.

Although the DDoS detection and mitigation techniques mentioned above can mitigate DDoS attacks in some experimental settings within the SDN environment, the diversified and covert nature of DDoS attack methods continues to present ongoing challenges. Therefore, there are still issues and challenges that need to be overcome in DDoS attack defense mechanisms.

- **Application plane security.** At present, most DDoS attack detection methods are deployed on the SDN control plane and data plane, neglecting security detection on the application plane. In fact, the security of the northbound interface of the SDN control plane also plays a crucial role in the normal operation of the SDN. Due to the openness and flexibility of SDN, there is a lack of strict access control, identity authentication, and abnormal detection mechanisms in the application layer. Attackers can launch a high volume of API calls within a short timeframe using malicious applications, resulting in controller crashes and the complete paralysis of the entire

network. Therefore, strengthening the security of the SDN application layer is also an important measure to defend against DDoS attacks.

- **Real network scenarios and load balance.** In real-world scenarios, SDN architecture inevitably faces synchronization and load-balancing issues caused by multi-controller systems. Currently, most research is based on simulation experiments of single-controller SDN systems. In real SDN deployments, a single controller system is unreliable. In a multi-controller system, the traffic of switches is distributed among various switches, which poses difficulties for DDoS attack detection. On one hand, DDoS attacks are more covert due to dispersed traffic, requiring more targeted detection thresholds. On the other hand, SDN with multiple controllers also needs to consider load balancing, distributing traffic evenly among different controllers to prevent being mistaken for an attack due to heavy load on a single controller. Wang et al. [111] deployed a DDoS attack defense scheme in a multi-controller system but did not consider the synchronization strategy of multiple controllers. The problem of effectively allocating resources, achieving load balancing, and synchronizing flow table information from multiple controllers is a challenge that SDN security policy deployment needs to address.
- **Network information synchronization.** Network information synchronization is the core issue of DDoS dynamic defense methods. If the synchronization of the sender and receiver information cannot be guaranteed during the information hopping process, it impacts network availability. The commonly used synchronization methods at present are time-based synchronization methods and protocol-based synchronization methods [112]. Time-based synchronization methods are affected by network latency and time accuracy, making it difficult to achieve accurate information synchronization. The protocol-based synchronization method requires prior communication negotiation and confirmation between the parties involved in the communication. However, this method is susceptible to replay attacks and tampering, which can disrupt the synchronization of network information jumps. Security research on information synchronization methods for dynamic defense is also a research direction.
- **Distinguishing between DDoS attacks and flash events.** In a real network, there are often multiple legitimate users accessing the network simultaneously, which can lead to flash events. During these events, the website server is unable to provide normal services [113]. Unlike DDoS attacks, this event is caused by a surge in network traffic from legitimate users and cannot be prevented solely through DDoS attack defense strategies. Luo et al. [114] introduced methods to distinguish and detect flash events and DDoS attacks, along with a dataset for detection. Sun et al. [64] proposed a method for detecting flow feature-based DDoS attacks and discriminating flash events in SDN. At present, it is also an urgent problem to distinguish between DDoS attacks and flash events in SDN and adopt different mitigation strategies to avoid affecting the legitimate use of the network by normal users.
- **Adaptive DDoS attack defense.** Attackers often adapt their attack methods based on the intelligence gathered in the early stages to evade network defenses and detection methods. Studying adaptive attack detection mechanisms for DDoS attacks in SDN has become an important topic. Based on statistical information for detecting DDoS attack methods, dynamic detection thresholds are set according to the actual network traffic size and attack methods in order to reduce false alarm rates. They minimize the impact on network availability while ensuring accurate detection. In machine learning detection methods, selecting traffic features based on attack types helps train the model for detection. This approach reduces model complexity while improving accuracy [115]. The currently commonly used method is to combine lightweight identification methods with heavyweight detection algorithms to efficiently and accurately detect and identify DDoS attacks. In dynamic defense methods, the selection of the information jump space and period also requires an adaptive adjustment in order to achieve an adaptive information jump. At present, research is focused on achieving

network adaptive DDoS attack detection and minimizing the impact on network availability. This involves developing DDoS defense measures that target various attack methods and scales.

- **Protocol security.** At present, there is no clear industry standard for security in SDN network architecture. Although organizations such as the Open Network Foundation [116] and the European Telecommunications Standards Association [117] have established certain security standards, there are still no fully recognized security standards domestically and internationally. This lack of recognized standards also impacts the security of SDN. Kloti et al. [118] conducted a security analysis on the OpenFlow protocol and experimentally verified that attackers can easily perform sniffing and DoS attacks on devices that deploy OpenFlow. In response to vulnerabilities in the SDN communication protocol, attackers can also compromise SDN security through methods such as man-in-the-middle attacks and spoofing attacks. Therefore, establishing security protocol standards is also an important measure to defend against DDoS attacks and ensure the security of SDN.

Table 10. Summary of DDoS attack defense techniques in SDN.

Technology	Scope	Plane	Key Points	Strengths	Weaknesses
Statistical analysis	Detection	Data/control	Utilizing statistical parameters of traffic characteristics or information entropy for the detection of DDoS attacks.	Low resource consumption and high real-time performance	High false alarm rate (FAR)
Machine learning	Detection/ Mitigation	Control	The deployment of machine learning algorithms in control planes to identify DDoS attack traffic in networks.	High accuracy	Model training is complex and has low real-time performance
Hybrid detection	Detection/ Mitigation	Data/Control	Statistical analysis and machine learning multi-level detection methods for DDoS attack detection.	Balancing real-time detection and accuracy	Difficulty in deployment Parameter settings affect detection effectiveness
MTD	Mitigation	Control	Dynamic changes in network information to mitigate DDoS attacks	Improve the security of SDN	High requirements for network systems and communication synchronization issues
Policy-based mitigation	Mitigation	Data/Control/ Application	Set traffic forwarding policies to effectively discard malicious traffic and ensure the transmission of clean traffic.	Easy to implement and minimal resource usage	May affect normal traffic

8. Conclusions

With the application of SDN architecture in various real-world scenarios, the security issues of SDN remain a significant challenge. On the one hand, many traditional network security problems still exist in SDN. On the other hand, the openness of SDN brings new security issues. This article focuses on the common DDoS attack problems in SDN and introduces several mainstream methods for detecting and mitigating DDoS attacks in SDN environments. The advantages and limitations of these methods are analyzed in terms of attack detection accuracy, real-time performance, network resource consumption, and types of DDoS attacks. Finally, this article raises questions and challenges regarding existing methods. Of course, there are also research areas that have not been covered in this article, such as the SDN communication protocol against DDoS attacks [119] and application-level defense methods against DDoS attacks [120]. Absolute network security does not exist, and

security attacks on networks will never end. Detecting and defending against new types of DDoS attacks in SDN environments will continue to be an area of exploration in the future. In future work, the author plans to utilize threat intelligence to model attackers, develop dynamic defense strategies using SDN programmability, achieve the early detection of DDoS attacks and localization of attackers, minimize controller overhead, and establish the traceability of attackers.

Funding: This research was funded by the Science and Technology on Complex Electronic System Simulation Laboratory, grant number 614201002012204.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chen, J.; Zheng, X.; Rong, C. Survey on software-defined networking. In Proceedings of the Second International Conference on Cloud Computing and Big Data in Asia, Huangshan, China, 17–19 June 2015; Springer: Cham, Switzerland, 2015; pp. 115–124.
2. Scott-Hayward, S.; Natarajan, S.; Sezer, S. A Survey of Security in Software Defined Networks. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 623–654. [CrossRef]
3. Ubale, T.; Jain, A.K. Survey on DDoS attack techniques and solutions in software-defined network. In *Handbook of Computer Networks and Cyber Security*; Springer: Cham, Switzerland, 2020; pp. 389–419.
4. Mittal, M.; Kumar, K.; Behal, S. Deep learning approaches for detecting DDoS attacks: A systematic review. *Soft Comput.* **2023**, *27*, 13039–13075. [CrossRef]
5. Ali, T.E.; Chong, Y.W.; Manickam, S. Machine Learning Techniques to Detect a DDoS Attack in SDN: A Systematic Review. *Appl. Sci.* **2023**, *13*, 3183. [CrossRef]
6. Karnani, S.; Shakya, H.K. Mitigation strategies for distributed denial of service (DDoS) in SDN: A survey and taxonomy. *Inf. Secur. J. Glob. Perspect.* **2023**, *32*, 444–468. [CrossRef]
7. Kaur, S.; Kumar, K.; Aggarwal, N.; Singh, G. A comprehensive survey of DDoS defense solutions in SDN: Taxonomy, research challenges, and future directions. *Comput. Secur.* **2021**, *110*, 102423. [CrossRef]
8. Behal, S.; Singh, J. Detection and Mitigation of DDoS attacks in SDN: A Comprehensive Review, Research Challenges and Future Directions. *Comput. Sci. Rev.* **2020**, *37*, 100279.
9. Maleh, Y.; Qasmaoui, Y.; El Gholami, K.; Sadqi, Y.; Mounir, S. A comprehensive survey on SDN security: Threats, mitigations, and future directions. *J. Reliab. Intell. Environ.* **2023**, *9*, 201–239. [CrossRef]
10. Ahmad, S.; Mir, A.H. SDN Interfaces: Protocols, Taxonomy and Challenges. *Int. J. Wirel. Microwave Technol.* **2022**, *12*, 11–32. [CrossRef]
11. Alhijawi, B.; Almajali, S.; Elgala, H.; Salameh, H.B.; Ayyash, M. A survey on DoS/DDoS mitigation techniques in SDNs: Classification, comparison, solutions, testing tools and datasets. *Comput. Electr. Eng.* **2022**, *99*, 107706. [CrossRef]
12. Patwardhan, A.; Jayarama, D.; Limaye, N.; Vidhale, S.; Parekh, Z.; Harfoush, K. SDN Security: Information disclosure and flow table overflow attacks. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019; IEEE: New York, NY, USA, 2019; pp. 1–6.
13. Cao, J.; Xu, M.; Li, Q.; Sun, K.; Yang, Y.; Zheng, J. Disrupting SDN via the data plane: A low-rate flow table overflow attack. In Proceedings of the International Conference on Security and Privacy in Communication Systems, Niagara Falls, ON, Canada, 22–25 October 2017; Springer: Cham, Switzerland, 2017; pp. 356–376.
14. Dover, J.M. *A Denial of Service Attack against the Open Floodlight SDN Controller*; Dover Networks LCC.: Edgewater, MD, USA, 2013.
15. Rauf, B.; Abbas, H.; Usman, M.; Zia, T.A.; Iqbal, W.; Abbas, Y.; Afzal, H. Application Threats to Exploit Northbound Interface Vulnerabilities in Software Defined Networks. *ACM Comput. Surv.* **2021**, *54*, 1–36. [CrossRef]
16. Yadav, S.K.; Suguna, P.; Velusamy, R.L. Entropy based mitigation of Distributed-Denial-of-Service (DDoS) attack on Control Plane in Software-Defined-Network (SDN). In Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 6–8 July 2019; IEEE: New York, NY, USA, 2019; pp. 1–7.
17. Ahalawat, A.; Dash, S.S.; Panda, A.; Babu, K.S. Entropy based DDoS detection and mitigation in OpenFlow enabled SDN. In Proceedings of the 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), Vellore, India, 30–31 March 2019; IEEE: New York, NY, USA, 2019; pp. 1–5.
18. Carvalho, R.N.; Bordim, J.L.; Alchieri EA, P. Entropy-based DoS attack identification in SDN. In Proceedings of the 2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Rio de Janeiro, Brazil, 20–24 May 2019; IEEE: New York, NY, USA, 2019; pp. 627–634.
19. Hemmati, Z.; Mirjalily, G.; Mohtajollah, Z. Entropy-based DDoS Attack Detection in SDN using Dynamic Threshold. In Proceedings of the 2021 7th International Conference on Signal Processing and Intelligent Systems (ICSPIS), Tehran, Iran, 29–30 December 2021; IEEE: New York, NY, USA, 2021; pp. 1–5.

20. Ujjan RM, A.; Pervez, Z.; Dahal, K.; Khan, W.A.; Khattak, A.M.; Hayat, B. Entropy based features distribution for anti-DDoS model in SDN. *Sustainability* **2021**, *13*, 1522. [CrossRef]
21. Tao, L.; Sheng, Y. DDoS attack detection and recognition based on cross entropy in SDN environment. *Comput. Appl. Softw.* **2018**, *38*, 328–333.
22. Kalkan, K.; Altay, L.; Gür, G.; Alagöz, F. JESS: Joint entropy-based DDoS defense scheme in SDN. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 2358–2372. [CrossRef]
23. Xuanyuan, M.; Ramsurrun, V.; Seeam, A. Detection and mitigation of DDoS attacks using conditional entropy in software-defined networking. In Proceedings of the 2019 11th International Conference on Advanced Computing (ICoAC), Chennai, India, 18–20 December 2019; IEEE: New York, NY, USA, 2019; pp. 66–71.
24. Li, R.; Wu, B. Early detection of DDoS based on φ-entropy in SDN networks. In Proceedings of the 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chongqing, China, 12–14 June 2020; IEEE: New York, NY, USA, 2020; Volume 1, pp. 731–735.
25. Kalkan, K.; Gür, G.; Alagöz, F. SDNScore: A statistical defense mechanism against DDoS attacks in SDN environment. In Proceedings of the 2017 IEEE Symposium on Computers and Communications (ISCC), Heraklion, Greece, 3–6 July 2017; IEEE: New York, NY, USA, 2017; pp. 669–675.
26. Fouladi, R.F.; Ermiş, O.; Anarim, E. A DDoS attack detection and defense scheme using time-series analysis for SDN. *J. Inf. Secur. Appl.* **2020**, *54*, 102587. [CrossRef]
27. Shohani, R.B.; Mostafavi, S.; Hakami, V. A statistical model for early detection of DDoS attacks on random targets in SDN. *Wirel. Pers. Commun.* **2021**, *120*, 379–400. [CrossRef]
28. Wang, M.H.; Wu, S.Y.; Yen, L.H.; Yen, L.H.; Tseng, C.C. PathMon: Path-specific traffic monitoring in OpenFlow-enabled networks. In Proceedings of the 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN), Vienna, Austria, 5–8 July 2016; IEEE: New York, NY, USA, 2016; pp. 775–780.
29. Sahay, R.; Blanc, G.; Zhang, Z.; Debar, H. ArOMA: An SDN based autonomic DDoS mitigation framework. *Comput. Secur.* **2017**, *70*, 482–499. [CrossRef]
30. Yuhua, X.; Zhixin, S. Research progress in abnormal traffic detection in software-defined networks. *J. Softw.* **2020**, *31*, 183–207. Available online: http://www.jos.org.cn/1000%E2%80%939825/5879.htm (accessed on 6 November 2019).
31. Kokila, R.T.; Selvi, S.T.; Govindarajan, K. DDoS detection and analysis in SDN-based environment using support vector machine classifier. In Proceedings of the 2014 Sixth International Conference on Advanced Computing (ICoAC), Chennai, India, 17–19 December 2014; IEEE: New York, NY, USA, 2014; pp. 205–210.
32. Mehr, S.Y.; Ramamurthy, B. An SVM based DDoS attack detection method for Ryu SDN controller. In Proceedings of the 15th International Conference on Emerging Networking Experiments and Technologies, Orlando, FL, USA, 9–12 December 2019; pp. 72–73.
33. Ye, J.; Cheng, X.; Zhu, J.; Feng, L.; Song, L. A DDoS attack detection method based on SVM in software defined network. *Secur. Commun. Netw.* **2018**, *2018*, 9804061. [CrossRef]
34. Zhao, J.; Zeng, P.; Shang, W.; Tong, G. DDoS attack detection based on one-class SVM in SDN. In Proceedings of the International Conference on Artificial Intelligence and Security, Hohhot, China, 17–20 July 2020; Springer: Singapore, 2020; pp. 189–200.
35. Myint Oo, M.; Kamolphiwong, S.; Kamolphiwong, T.; Vasupongayya, S. 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*, 8012568. [CrossRef]
36. Abdullahi Wabi, A.; Idris, I.; Mikail Olaniyi, O.; Joseph, A.; Surajudeen Adebayo, O. Modeling DDOS attacks in sdn and detection using random forest classifier. *J. Cyber Secur. Technol.* **2023**, 1–14. [CrossRef]
37. Santos, R.; Souza, D.; Santo, W.; Ribeiro, A.; Moreno, E. Machine learning algorithms to detect DDoS attacks in SDN. *Concurr. Comput. Pract. Exp.* **2020**, *32*, e5402. [CrossRef]
38. Khashab, F.; Moubarak, J.; Feghali, A.; Bassil, C. DDoS attack detection and mitigation in SDN using machine learning. In Proceedings of the 2021 IEEE 7th International Conference on Network Softwarization (NetSoft), Tokyo, Japan, 28 June–2 July 2021; IEEE: New York, NY, USA, 2021; pp. 395–401.
39. Dong, S.; Sarem, M. DDoS attack detection method based on improved KNN with the degree of DDoS attack in software-defined networks. *IEEE Access* **2019**, *8*, 5039–5048. [CrossRef]
40. Latah, M.; Toker, L. Towards an efficient anomaly-based intrusion detection for software-defined networks. *IET Netw.* **2018**, *7*, 453–459. [CrossRef]
41. Nam, T.M.; Phong, P.H.; Khoa, T.D.; Huong, T.T.; Nam, P.N.; Thanh, N.H.; Thang, L.X.; Tuan, P.A.; Dung, L.Q.; Loi, V.D. Self-organizing map-based approaches in DDoS flooding detection using SDN. In Proceedings of the 2018 International Conference on Information Networking (ICOIN), Chiang Mai, Thailand, 10–12 January 2018; IEEE: New York, NY, USA, 2018; pp. 249–254.
42. Hnamte, V.; Balram, G. Implementation of Naive Bayes Classifier for Reducing DDoS Attacks in IoT Networks. *J. Algebr. Stat.* **2022**, *13*, 2749–2757.
43. Nadeem, M.W.; Goh, H.G.; Ponnusamy, V.; Aun, Y. DDoS Detection in SDN using Machine Learning Techniques. *Comput. Mater. Contin.* **2022**, *71*, 1. [CrossRef]
44. Alubaidan, H.; Alzaher, R.; AlQhatani, M.; Mohammed, R. DDoS Detection in Software-Defined Network (SDN) Using Machine Learning. *Int. J. Cybern. Inform.* **2023**, *12*, 93–104. [CrossRef]

45. Wang, J.; Wang, L. SDN-Defend: A Lightweight Online Attack Detection and Mitigation System for DDoS Attacks in SDN. *Sensors* **2022**, *22*, 8287. [CrossRef] [PubMed]
46. Wang, J.; Wang, L.; Wang, R. A Method of DDoS Attack Detection and Mitigation for the Comprehensive Coordinated Protection of SDN Controllers. *Entropy* **2023**, *25*, 1210. [CrossRef]
47. Jmal, R.; Ghabri, W.; Guesmi, R.; Alshammari, B.M.; Alshammari, A.S.; Alsaif, H. Distributed Blockchain-SDN Secure IoT System Based on ANN to Mitigate DDoS Attacks. *Appl. Sci.* **2023**, *13*, 4953. [CrossRef]
48. Priyadarshini, I.; Mohanty, P.; Alkhayyat, A.; Sharma, R.; Kumar, S. SDN and application layer DDoS attacks detection in IoT devices by attention-based Bi-LSTM-CNN. *Trans. Emerg. Telecommun. Technol.* **2023**, *34*, e4758. [CrossRef]
49. Li, C.; Wu, Y.; Yuan, X.; Sun, Z.; Wang, W.; Li, X.; Gong, L. Detection and defense of DDoS attack–based on deep learning in OpenFlow-based SDN. *Int. J. Commun. Syst.* **2018**, *31*, e3497. [CrossRef]
50. Bastola, S.B.; Shakya, S.; Sharma, S. Distributed Denial of Service Attack Detection on Software Defined Networking Using Deep Learning. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 13–16 September 2017.
51. Makuvaza, A.; Jat, D.S.; Gamundani, A.M. Deep neural network (DNN) solution for real-time detection of distributed denial of service (DDoS) attacks in software defined networks (SDNs). *SN Comput. Sci.* **2021**, *2*, 1–10. [CrossRef]
52. Zhao, J.; Xu, M.; Chen, Y.; Xu, G. A DNN Architecture Generation Method for DDoS Detection via Genetic Alogrithm. *Future Internet* **2023**, *15*, 122. [CrossRef]
53. Al-Dunainawi, Y.; Al-Kaseem, B.R.; Al-Raweshidy, H.S. Optimized Artificial Intelligence Model for DDoS Detection in SDN Environment. *IEEE Access* **2023**, *11*, 106733–106748. [CrossRef]
54. Aslam, M.; Ye, D.; Tariq, A.; Asad, M.; Hanif, M.; Ndzi, D.; Chelloug, S.A.; Elaziz, M.A.; Al-Qaness, M.A.A.; Jilani, S.F. Adaptive Machine Learning Based Distributed Denial-of-Services Attacks Detection and Mitigation System for SDN-Enabled IoT. *Sensors* **2022**, *22*, 2697. [CrossRef]
55. Zhijun, W.; Qing, X.; Jingjie, W.; Meng, Y.; Liang, L. Low-rate DDoS attack detection based on factorization machine in software defined network. *IEEE Access* **2020**, *8*, 17404–17418. [CrossRef]
56. Li, J.; Liu, Y.; Gu, L. DDoS attack detection based on neural network. In Proceedings of the 2010 2nd International Symposium on Aware Computing, Tainan, Taiwan, 1–4 November 2010; IEEE: New York, NY, USA, 2010; pp. 196–199.
57. Malik, J.; Akhunzada, A.; Bibi, I.; Imran, M.; Musaddiq, A.; Kim, S.W. Hybrid deep learning: An efficient reconnaissance and surveillance detection mechanism in SDN. *IEEE Access* **2020**, *8*, 134695–134706. [CrossRef]
58. Cui, Y.; Yan, L.; Li, S.; Xing, H.; Pan, W.; Zhu, J.; Zheng, X. SD-Anti-DDoS: Fast and efficient DDoS defense in software-defined networks. *J. Netw. Comput. Appl.* **2016**, *68*, 65–79. [CrossRef]
59. Deepa, V.; Sivakumar, B. Detection of DDoS Attack using Multiple Kernel Level (MKL) Algorithm. In Proceedings of the 2022 International Conference on Innovative Trends in Information Technology (ICITIIT), Kottayam, India, 12–13 February 2022; IEEE: New York, NY, USA, 2022; pp. 1–5.
60. Qi, N.; Wang, W.; Xiao, M.; Jia, L.; Tsiftsis, T. A Learning-Based Spectrum Access Stackelberg Game: Friendly Jammer-Assisted Communication Confrontation. *IEEE Trans. Veh. Technol.* **2021**, *70*, 700–713. [CrossRef]
61. Jia, L.; Xu, Y.; Sun, Y.; Feng, S.; Anpalagan, A. Stackelberg Game Approaches for Anti-Jamming Defence in Wireless Networks. *IEEE Wirel. Commun.* **2018**, *25*, 120–128. [CrossRef]
62. Yao, R.; Zhang, Y.; Wang, S.; Qi, N.; Miridakis, N.I.; Tsiftsis, T.A. Deep Neural Network Assisted Approach for Antenna Selection in Untrusted Relay Networks. *IEEE Wirel. Commun. Lett.* **2019**, *8*, 1644–1647. [CrossRef]
63. Hu, D.; Hong, P.; Chen, Y. FADM: DDoS flooding attack detection and mitigation system in software-defined networking. In Proceedings of the GLOBECOM 2017—2017 IEEE Global Communications Conference, Singapore, 4–8 December 2017; IEEE: New York, NY, USA, 2017; pp. 1–7.
64. Guozi Sun Jiang, W.; Yu, G.U.; Danni, R.E.N.; Huakang, L.I. DDoS attacks and flash event detection based on flow characteristics in SDN. In Proceedings of the 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, 27–30 November 2018; IEEE: New York, NY, USA, 2018; pp. 1–6.
65. Novaes, M.P.; Carvalho, L.F.; Lloret, J.; Proenca, M.L. Long short-term memory and fuzzy logic for anomaly detection and mitigation in software-defined network environment. *IEEE Access* **2020**, *8*, 83765–83781. [CrossRef]
66. Banitalebi Dehkordi, A.; Soltanaghaei, M.R.; Boroujeni, F.Z. The DDoS attacks detection through machine learning and statistical methods in SDN. *J. Supercomput.* **2021**, *77*, 2383–2415. [CrossRef]
67. Long, Z.; Jinsong, W. A hybrid method of entropy and SSAE-SVM based DDoS detection and mitigation mechanism in SDN. *Comput. Secur.* **2022**, *115*, 102604. [CrossRef]
68. Singh, A.K.; Jaiswal, R.K.; Abdukodir, K.; Muthanna, A. Ardefense: DDos detection and prevention using nfv and sdn. In Proceedings of the 2020 12th International Congress on Ultra Mod Ern Telecommunications and Control Systems and Workshops (ICUMT), Brno, Czech Republic, 5–7 October 2020; IEEE: New York, NY, USA, 2020; pp. 236–241. [CrossRef]
69. Ali, A.; Yousaf, M.M. Novel three-tier intrusion detection and prevention system in software defined network. *IEEE Access* **2020**, *8*, 109662–109676. [CrossRef]
70. Sarwar, M.A.; Hussain, M.; Anwar, M.U.; Ahmad, M. FlowJustifier: An optimized trust-based request prioritization approach for mitigation of SDN controller DDoS attacks in the IoT paradigm. In Proceedings of the 3rd International Conference on Future Networks and Distributed Systems, Paris, France, 1–2 July 2019; pp. 1–9.

71. Deng, S.; Gao, X.; Lu, Z.; Li, Z.; Gao, X. DoS vulnerabilities and mitigation strategies in software-defined networks. *J. Netw. Comput. Appl.* **2019**, *125*, 209–219. [CrossRef]
72. Ravi, N.; Shalinie, S.M.; Lal, C.; Conti, M. AEGIS: Detection and mitigation of TCP SYN flood on SDN controller. *IEEE Trans. Netw. Serv. Manag.* **2020**, *18*, 745–759. [CrossRef]
73. Cao, Y.; Jiang, H.; Deng, Y.; Wu, J.; Zhou, P.; Luo, W. Detecting and mitigating ddos attacks in SDN using spatial-temporal graph convolutional network. *IEEE Trans. Dependable Secur. Comput.* **2021**, *19*, 3855–3872. [CrossRef]
74. Wang, M.; Zhou, H.; Chen, J.; Tong, B. An approach for protecting the openflow switch from the saturation attack. In Proceedings of the 2015 4th National Conference on Electrical, Electronics and Computer Engineering, Xi'an, China, 12–13 December 2015; Atlantis Press: Dordrecht, The Netherlands, 2015.
75. Bawany, N.Z.; Shamsi, J.A. Seal: Sdn based secure and agile framework for protecting smart city applications from ddos attacks. *J. Netw. Comput. Appl.* **2019**, *145*, 102381. [CrossRef]
76. Yuan, B.; Zou, D.; Yu, S.; Jin, H.; Qiang, W.; Shen, J. Defending against flow table overloading attack in software-defined networks. *IEEE Trans. Serv. Comput.* **2016**, *12*, 231–246. [CrossRef]
77. Bhushan, K.; Gupta, B.B. Distributed denial of service (DDoS) attack mitigation in software defined network (SDN)-based cloud computing environment. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 1985–1997. [CrossRef]
78. Katta, N.; Alipourfard, O.; Rexford, J.; Walker, D. Infinite CacheFlow in software-defined networks. In Proceedings of the Third Workshop on Hot Topics in Software Defined Networking (HotSDN'14), Chicago, IL, USA, 22 August 2014; Association for Computing Machinery: New York, NY, USA, 2014; pp. 175–180.
79. Dang, V.T.; Huong, T.T.; Thanh, N.H.; Nam, P.N.; Thanh, N.N.; Marshall, A. Sdn-based synproxy—A solution to enhance performance of attack mitigation under tcp syn flood. *Comput. J.* **2019**, *62*, 518–534. [CrossRef]
80. Pascoal, T.A.; Dantas, Y.G.; Fonseca, I.E.; Nigam, V. Slow TCAM exhaustion DDoS attack. In Proceedings of the IFIP International Conference on ICT Systems Security and Privacy Protection, Rome, Italy, 29–31 May 2017; Springer: Cham, Switzerland, 2017; pp. 17–31.
81. Ma, D.; Xu, Z.; Lin, D. Defending blind DDoS attack on SDN based on moving target defense. In Proceedings of the International Conference on Security and Privacy in Communication Networks, Beijing, China, 24–26 September 2014; Springer: Cham, Switzerland, 2014; pp. 463–480.
82. Abou El Houda, Z.; Khoukhi, L.; Hafid, A.S. Bringing intelligence to software defined networks: Mitigating ddos attacks. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 2523–2535. [CrossRef]
83. Hong, G.C.; Lee, C.N.; Lee, M.F. Dynamic threshold for DDoS mitigation in SDN environment. In Proceedings of the 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Lanzhou, China, 18–21 November 2019; IEEE: New York, NY, USA, 2019; pp. 1–7.
84. Alamri, H.A.; Thayananthan, V. Bandwidth control mechanism and extreme gradient boosting algorithm for protecting software-defined networks against ddos attacks. *IEEE Access* **2020**, *8*, 194269–194288. [CrossRef]
85. Wang, L.; Li, Q.; Jiang, Y.; Jia, X.; Wu, J. Woodpecker: Detecting and mitigating link-flooding attacks via sdn. *Comput. Netw.* **2018**, *147*, 1–13. [CrossRef]
86. Weizhen, L.; Hailong, L.; Kaiyu, H. End jump technology research review. *Comput. Appl. Res.* **2021**, *38*, 2251–2257. [CrossRef]
87. Atighetchi, M.; Pal, P.; Webber, F.; Jones, C. Adaptive use of network-centric mechanisms in cyber-defense. In Proceedings of the Sixth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing, Hokkaido, Japan, 16 May 2003; IEEE: New York, NY, USA, 2003; pp. 183–192.
88. Leyi, S.; Chunfu, J.; Shuwang, L. Research on Active Network Protection Based on Terminal Information Jump. *J. Commun.* **2008**, *2*, 106–110.
89. Badishi, G.; Herzberg, A.; Keidar, I. Keeping denial-of-service attackers in the dark. *IEEE Trans. Dependable Secur. Comput.* **2007**, *4*, 191–204. [CrossRef]
90. Zhang, L.; Guo, Y.; Yuwen, H.; Wang, Y. A port hopping based dos mitigation scheme in SDN network. In Proceedings of the 2016 12th International Conference on Computational Intelligence and Security (CIS), Wuxi, China, 16–19 December 2016; IEEE: New York, NY, USA, 2016; pp. 314–317.
91. Chowdhary, A.; Alshamrani, A.; Huang, D.; Liang, H. MTD analysis and evaluation framework in software defined network (MASON). In Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization, Tempe, AZ, USA, 21 March 2018; pp. 43–48.
92. Ziyu, Z.; Erdian, G.; Wei, L. Research on encryption-based port jump technology in software-defined network. *Comput. Appl. Softw.* **2017**, *34*, 322–328.
93. Sifalakis, M.; Schmid, S.; Hutchison, D. Network address hopping: A mechanism to enhance data protection for packet communications. In Proceedings of the IEEE International Conference on Communications, ICC 2005, Seoul, Republic of Korea, 16–20 May 2005; IEEE: New York, NY, USA, 2005; Volume 3, pp. 1518–1523.
94. Zheng, K.; Zhao, X.; Li, X.; Zhou, Y. A SDN-based IP Address Hopping Method Design. In Proceedings of the 2016 5th International Conference on Measurement, Instrumentation and Automation (ICMIA 2016), Shenzhen, China, 17–18 September 2016; Atlantis Press: New York, NY, USA, 2016.
95. De, T.; Wei, L. SDN address hopping scheme based on chaotic sequence. *Comput. Digit. Eng.* **2018**, *46*, 2315–2318.

96. Chang, S.Y.; Park, Y.; Babu, B.B.A. Fast IP hopping randomization to secure hop-by-hop access in SDN. *IEEE Trans. Netw. Serv. Manag.* **2018**, *16*, 308–320. [CrossRef]
97. Xu, X.; Hu, H.; Liu, Y.; Zhang, H.; Chang, D. An Adaptive IP Hopping Approach for Moving Target Defense Using a Light-Weight CNN Detector. *Secur. Commun. Netw.* **2021**, *2021*, 8848473. [CrossRef]
98. Lou, W.; Li, H.; Hu, K.; Liu, M.; Dong, Q. Flow count synchronous SDN address hopping technology based on DH-RSA negotiation. In Proceedings of the 2021 International Conference on Neural Networks, Information and Communication Engineering, Qingdao, China, 27–28 August 2021; SPIE: Bellingham, WA, USA, 2021; Volume 11933, pp. 251–259.
99. Jinglei, T.; Hongqi, Z.; Cheng, L.; Zhang, Y.; Chang, D.; Liu, X.; Zhang, H. Research progress on moving target defense technology for SDN. *J. Netw. Inf. Secur.* **2018**, *4*, 12.
100. Shi, L.; Jia, C.; Lü, S.; Liu, Z. Port and address hopping for active cyber-defense. In Proceedings of the Pacific-Asia Workshop on Intelligence and Security Informatics, Chengdu, China, 11–12 April 2007; Springer: Berlin/Heidelberg, Germany, 2007; pp. 295–300.
101. Yixun, H.; Kangfeng, Z.; Yixian, Y.; Xinxin, N. Network Layer Moving Target Defense Scheme based on OpenFlow. *J. Commun.* **2017**, *38*, 102–112.
102. Liu, Z.; He, Y.; Wang, W.; Wang, S.; Li, X.; Zhang, B. AEH-MTD: Adaptive moving target defense scheme for SDN. In Proceedings of the 2019 IEEE International Conference on Smart Internet of Things (SmartIoT), Tianjin, China, 9–11 August 2019; IEEE: New York, NY, USA, 2019; pp. 142–147.
103. Yuyang, Z.; Guang, C.; Chunsheng, G.; Mian, D. Moving targets defense attack surface dynamic transfer technology research review. *J. Softw.* **2018**, *29*, 2799–2820.
104. Lixia, X.; Ying, D. Link SDN flooding attack moving targets defense mechanism. *J. Tsinghua Univ.* **2019**, *59*, 36–43. [CrossRef]
105. Liu, J.; Zhang, H.; Guo, Z. A defense mechanism of random routing mutation in SDN. *IEICE Trans. Inf. Syst.* **2017**, *100*, 1046–1054. [CrossRef]
106. Hyder, M.F.; Fatima, T.; Khan, S.M.; Arshad, S. Countering crossfire DDoS attacks through moving target defense in SDN networks using OpenFlow traffic modification. *Trans. Emerg. Telecommun. Technol.* **2023**, *34*, e4853. [CrossRef]
107. Chowdhary, A.; Pisharody, S.; Alshamrani, A.; Huang, D. Dynamic game based security framework in SDN-enabled cloud networking environments. In Proceedings of the ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization, Scottsdale, AZ, USA, 24 March 2017; pp. 53–58.
108. Zhou, Y.; Cheng, G.; Jiang, S.; Chen, Z. Cost-effective moving target defense against DDoS attacks using trilateral game and multi-objective Markov decision processes. *Comput. Secur.* **2020**, *97*, 101976. [CrossRef]
109. Du, M.; Wang, K. An SDN-Enabled Pseudo-Honeypot Strategy for Distributed Denial of Service Attacks in Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2020**, *16*, 648–657. [CrossRef]
110. Priyadarsini, M.; Bera, P.; Das, S.K.; Rahman, M.A. A security enforcement framework for SDN controller using game theoretic approach. *IEEE Trans. Dependable Secur. Comput.* **2022**, *20*, 1500–1515. [CrossRef]
111. Wang, Y.; Hu, T.; Tang, G.; Xie, J.; Lu, J. SGS: Safe-Guard Scheme for Protecting Control Plane Against DDoS Attacks in Software-Defined Networking. *IEEE Access* **2019**, *7*, 34699–34710. [CrossRef]
112. Weizhen, H.; Fucai, C.; Jie, N.; Jinglei, T.; Shumin, H.; Guozhen, C. Research progress of Dynamic Jump Technology for Network Layer. *J. Netw. Inf. Secur.* **2021**, *7*, 44–55.
113. Bhatia, S.; Mohay, G.; Tickle, A.; Ahmed, E. Parametric differences between a real-world distributed denial-of-service attack and a flash event. In Proceedings of the 2011 Sixth International Conference on Availability, Reliability and Security, Vienna, Austria, 22–26 August 2011; IEEE: New York, NY, USA, 2011; pp. 210–217.
114. Kai, L.; Junyong, L.; Meijuan, Y.; Yan, L.; Lizheng, G. A review on the Identification of DDoS attacks with Flash Crowd. *Comput. Sci.* **2015**, *42*, 313–316+322.
115. Jia, L.; Qi, N.; Chu, F.; Fang, S.; Wang, X.; Ma, S.; Feng, S. Game-theoretic learning anti-jamming approaches in wireless networks. *IEEE Commun. Mag.* **2022**, *60*, 60–66. [CrossRef]
116. ONF. Software-Defined Networking (SDN) Definition. Available online: https://opennetworking.org/sdn-resources/sdn-definition (accessed on 30 June 2022).
117. European Telecommunications Standards Institute. Available online: http://www.etsi.org/ (accessed on 30 June 2022).
118. Kloti, R.; Kotronis, V.; Smith, P. OpenFlow: A security analysis. In Proceedings of the Twenty first IEEE International Conference on Network Protocols (ICNP), Göttingen, Germany, 7–10 October 2013; pp. 1–6.
119. Sjoholmsierchio, M.; Hale, B.; Lukaszewski, D.; Xie, G.G. Strengthening SDN security: Protocol dialecting and downgrade attacks. In Proceedings of the 2021 IEEE 7th International Conference on Network Softwarization (NetSoft), Tokyo, Japan, 28 June–2 July 2021; IEEE: New York, NY, USA, 2021; pp. 321–329.
120. Yang, W.; Guang-ming, T.; Shuo, W.; Jiang, C. DDoS Attack Defense mechanism at SDN Application Layer based on API Call management. *J. Netw. Inf. Secur.* **2022**, *8*, 73–87.