Journal of Network and Computer Applications 190 (2021) 103156

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Review

Towards DDoS detection mechanisms in Software-Defined Networking

Yunhe Cui [a,*], Qing Qian [b], Chun Guo [a], Guowei Shen [a], Youliang Tian [a], Huanlai Xing [c], Lianshan Yan [c]

[a] State Key Laboratory of Public Big Data, School of Computer Science and Technology, Guizhou University, Guiyang, China
[b] School of Information, GuiZhou University of Finance and Economics, Guiyang, China
[c] School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China

ARTICLE INFO

Keywords:
Distributed Denial of Service
Attack detection
Software-Defined Networking
OpenFlow

ABSTRACT

Software-Defined Networking (SDN) is widely considered as one of the next generation network architecture. However, SDN faces with a series of issues which restraint its development and application, where the security is one of the serious issues. The Distributed Denial of Service (DDoS) is such a devastating security problem. In this work, a comprehensive review of the DDoS detection mechanisms utilized in SDN is presented. DDoS attacks in SDN are classified into two types and five subtypes based on the features of DDoS and SDN. For each kind of DDoS, how the attackers can exploit the vulnerabilities of SDN to launch DDoS attacks is discussed. Subsequently, the DDoS detection mechanisms used in SDN are reviewed and categorized into five types and forty-six subtypes. These kinds of DDoS detection mechanisms are compared and analyzed, where we draw a conclusion that the machine learning-based DDoS detection mechanisms and threshold-based DDoS detection mechanisms are the two most popular technologies utilized to detect DDoS attacks in SDN. More importantly, for each kind of DDoS detection mechanism, the generational process, advantages, and disadvantages are discussed. Additionally, the open problems and future directions of DDoS detection in SDN are discussed. By presenting these review, discussion and analysis, we hope it can facilitate the understanding of DDoS detection in SDN.

1. Introduction

As a passageway that connects communication terminals around the world, the traditional network architecture carries massive network traffic. Nevertheless, new emerging technologies such as edge computing, cloud computing, mobile computing have brought significant variation to network traffic patterns. More specifically, the development of mobile technology makes users access the Internet from their mobile equipments anytime and anywhere. Furthermore, in the edge computing and cloud computing, a large number of 'east–west' traffics are appeared, which are obviously different from the classic 'north–south' traffic pattern. Meanwhile, numbers of clouds also create passive traffic over the wide area network. For the traditional network architecture which can be represented by the conventional switches and routers, it is challenging to meet the needs of rapid changes of network traffic, due to the fact that the network managers must log in the switches or routers and modify the associated configuration, using the vendor-dependent management interface. In addition, the traditional network has disadvantages on scalability, which makes it hard to satisfy the scalability requirements of many network service providers.

In order to overcome the issues of the traditional network architecture, SDN has been presented (Casado et al., 2006; Anon, 2012a). Taking advantage of decoupling the data plane and control plane of the traditional network, SDN can control the network devices via the applications running on application servers, which makes the network become'software-defined'. Due to the network programmability, unified control ability, and global visibility provided by SDN, it gains great interest from the academia and industry. Especially, Google has shown how they use SDN to re-design its global data center network in Jain et al. (2013), which may be regarded as a milestone of utilizing SDN in commercial networks. At present, although SDN has been used in the wide area network, data center, and wireless network, there are still some security barriers which hinder SDN from widespread application in the global network (Jain et al., 2013; Anbalagan et al., 2020; Akyildiz et al., 2015; Yuan et al., 2017).

As one of these intractable network security issues, the DDoS is a quite devastating attack. Specifically, hackers often use DDoS to send enormous requests to a system with the intention of overwhelming the normal service provided by that system. In addition, there have been a certain number of easily obtainable DDoS attack tools, which

* Corresponding author.
E-mail address: yhcui@gzu.edu.cn (Y. Cui).

https://doi.org/10.1016/j.jnca.2021.103156
Received 2 November 2020; Received in revised form 26 March 2021; Accepted 28 June 2021
Available online 10 July 2021

makes initiating a DDoS attack become quite easy. For instance, HULK, Tor's Hammer, Slowloris, LOIC, Xoic, DDOSIM, RUDY, and PyLoris can be employed to start a DDoS attack by an inexperienced attacker. In conclusion, DDoS has characteristics of easy to initiate, hard to defend, and strong destructiveness. Unfortunately, despite the defense ability of DDoS is improving in traditional network architecture, the lack of network programmability of traditional network architecture makes it hard to deploy some state-of-the-art technologies when defending DDoS. Considering the network programmability provided by SDN, how to develop more efficient defense mechanisms against DDoS in SDN has attracted strong interest in recent years.

In this work, we focus on the extensive survey of the DDoS detection mechanisms in SDN. 143 DDoS detection works published from 2010 to 2020 (Cui et al., 2016; Chen and Yu, 2016a; Braga et al., 2010; Phan and Park, 2019; Phan et al., 2016, 2017; Wang and Chen, 2017; Xu and Liu, 2016; Zhao and Liu, 2018; Nam et al., 2018; Pillutla and Arjunan, 2019; Phan et al., 2019; Wang et al., 2019; Cui et al., 2018; Mihai-Gabriel and Victor-Valeriu, 2014; Liu et al., 2019; Santos et al., 2020; Gharvirian and Bohlooli, 2017; Wang et al., 2020a; Chen and Yu, 2016b; Dayal and Srivastava, 2018; MohanaPriya and Shalinie, 2017; Gong et al., 2019; Rahman et al., 2019; Bakker et al., 2018; Musumeci et al., 2020; Cui et al., 2019; Myint Oo et al., 2019; Ye et al., 2018; Yu et al., 2018; Hu et al., 2017; He et al., 2018; Yang and Zhao, 2018; Latah and Toker, 2018; Shang et al., 2017; Shen et al., 2020; Oo et al., 2017; Li et al., 2015; Kokila et al., 2014; Liu et al., 2017a; Chen et al., 2017; Mehr and Ramamurthy, 2019; Mowla et al., 2018; Polat et al., 2020; Hyder and Lung, 2018; Alshamrani et al., 2017; Han et al., 2018; Tan et al., 2020; Zhu et al., 2018; Sun et al., 2018; Ahmed et al., 2017; Gao et al., 2018; Kalliola et al., 2015; Shakil et al., 2019; Arivudainambi et al., 2019; Wang et al., 2020b; Haider et al., 2020; Li et al., 2018; Narayanadoss et al., 2019; SaiSindhuTheja and Shyam, 2021; Tang et al., 2016; Asad et al., 2019; Chen et al., 2018a; Wang et al., 2016a; Conti et al., 2017; Mahrach et al., 2018; Birkinshaw et al., 2019; Özçelik et al., 2017; Zerbini et al., 2019; Dong et al., 2016; Wang et al., 2018a; Kalkan et al., 2017; Dang-Van and Truong-Thu, 2017; Sudar and Deepalakshmi, 2020; Chen et al., 2018b; Kalkan et al., 2018; Wang et al., 2018b; Zheng et al., 2018; Conti et al., 2019; Viet et al., 2017; Mousavi and St-Hilaire, 2018; Kumar et al., 2018b; David and Thomas, 2019; Guesmi and Saidane, 2017; Gurusamy and MSK, 2019; Lin et al., 2017; Xu et al., 2017; Sambandam et al., 2018; Liu et al., 2017b; Boite et al., 2017; Huong and Thanh, 2017; Yang et al., 2017; You et al., 2017; Wang et al., 2017; Pandikumar et al., 2017; Tsai et al., 2017; Buragohain and Medhi, 2016; Chen et al., 2016; Xing et al., 2016; Piedrahita et al., 2015; Van Trung et al., 2015; Wang et al., 2015a; Hommes et al., 2014; Duy and Pham, 2018; Lu and Wang, 2016; Murtuza and Asawa, 2018; Jiang et al., 2016; Rebecchi et al., 2019; Wang et al., 2016b; Bhushan and Gupta, 2018; Wang et al., 2018c; Sahoo et al., 2018a,b; Dehkordi and Soltanaghaei, 2020; Wu et al., 2020; Mishra et al., 2021; Rahouti et al., 2021; Agrawal and Tapaswi, 2021; Lukaseder et al., 2018; Hong et al., 2017; Lukaseder et al., 2017; Shtern et al., 2014; Mohammadi et al., 2017; Gkountis et al., 2017; Dao et al., 2015, 2016; Rathore et al., 2019; Ujjan et al., 2019; Shao et al., 2019; Shu et al., 2020; Nguyen et al., 2019; Guo et al., 2019; Xiao et al., 2016; Chin et al., 2015; Manso et al., 2019; De Assis et al., 2017; Yin et al., 2018; Yan et al., 2016b; Wang et al., 2018d; Ivannikova et al., 2017; Aleroud and Alsmadi, 2016; Wang et al., 2015b; Wei et al., 2016) are carefully read and listed. We meticulously classify these DDoS detection mechanisms for up to three classification layers. In the first classification layer, the DDoS detection mechanisms are categorized into five types including the machine learning-based, statistical-based, combination of multiple methods-based, threshold-based, and other method-based DDoS detection mechanisms. We further classify these five kinds of DDoS detection mechanisms into fine-grained subtypes. For instance, the machine learning-based DDoS detection mechanisms are further categorized as the neural network-based, classifying-based, clustering-based, deep learning-based, and ensemble learning-based DDoS detection mechanisms. Moreover, these subtypes are classified into more fine-grained subtypes once again. Take the neural network-based mechanisms as an example, this kind of mechanisms are further classified into the Back Propagation Neural Network (BPNN)-based, Extreme Learning Machine (ELM)-based, Multi-Layer Perception (MLP)-based, Radial Basis Function (RBF)-based, Restricted Boltzmann Machine (RBM)-based, and Self Organizing Maps (SOM)-based DDoS detection mechanisms.

We also analyze the advantages and disadvantages of the machine learning-based, combination of multiple methods-based, statistical method-based, threshold-based, and other method-based DDoS detection mechanisms. A conclusion drawn from the analysis is that the machine learning-based DDoS detection and threshold-based DDoS detection mechanisms are the two most popular technologies utilized to detect DDoS attacks in SDN. In addition, for the machine learning-based DDoS detection mechanisms, the classifying-based detection mechanisms and neural network-based detection mechanisms are the most popular approaches.

Some survey papers about the DDoS detection mechanisms in SDN have been published. In Table 1, a qualitative comparison of our work with the existing related works is listed. The number of technologies (e.g. SVM, SOM, and BPNN) mentioned in the work, number of references, number of related papers, integral analysis, separate analysis, issues and challenges are chosen as the parameters of the comparison. A more comprehensive review with more detailed analysis has been done in this work, as shown in Table 1.

The prime contributions of this work are summarized below:

- We give a comprehensive review of DDoS detection mechanisms in SDN.
- An inclusive and detailed classification of the DDoS detection mechanisms is proposed, which includes five types and forty-six subtypes.
- The generational process, advantages, and disadvantages of the neural network, classifying, clustering, deep learning, ensemble learning, statistics, combination of multiple methods, threshold, and other method based DDoS detection mechanisms have been discussed.
- The open issues and possible future research directions of DDoS detection in SDN are discussed.

The paper is organized as following: Section 2 gives the description of SDN. Section 3 characterizes DDoS attack in SDN. In detail, the machine learning-based, statistics-based, combination of multiple methods-based, threshold-based, and other method-based DDoS detection are expounded and analyzed in Sections 4, 5, 6, 7, and 8, respectively. Subsequently, the analysis of DDoS detection mechanisms is carried out in Section 9. Open problems and future directions are discussed in Section 10, followed by the conclusions in Section 11.

2. Background of SDN

2.1. Definition and architecture of SDN

The Open Network Foundation (ONF) has issued the SDN white paper -Software-Defined Networking: The New Norm for Networks- in 2012 (Anon, 2012a; ONF, 2021). It gives an official definition of SDN - "Software Defined Networking (SDN) is an emerging network architecture where network control is decoupled from forwarding and is directly programmable" (Anon, 2012a). From this definition, we can observe that there are two significant features in SDN: (1) it decouples the control function and forwarding function of network devices, and (2) it provides network programmability.

ONF also presented the architecture of SDN (Anon, 2012a). According to their presentation, SDN contains three layers and two interfaces including (1) infrastructure layer, (2) control layer, (3) application layer, (4) southbound interface, and (5) northbound interface, as shown

Table 1
Qualitative comparison with the existing reviews.

Reference	Year	Number of technologies	Number of references	Number of related papers	Tables with rationales for detection	Integral analysis	Separate analysis	Issues and challenges
Singh and Bhandari (2020)	2020	15	103	39	3	N	N	**Y**
Behal and Singh (2020)	2020	30	161	62	4	N	N	**Y**
Al-Adaileh et al. (2020)	2020	15	70	29	2	Y	N	**Y**
Tayyab et al. (2020)	2020	19	91	34	3	Y	N	**Y**
Swami et al. (2019)	2019	15	92	51	2	N	N	**Y**
Dong et al. (2019)	2019	–	124	48	3	N	N	**Y**
Sahoo et al. (2019)	2019	10	162	57	2	N	N	**Y**
Mumpela and Young-Hoon (2018)	2018	12	79	46	–	N	N	N
Kubra et al. (2017)	2017	3	15	7	–	N	**Y**	N
Bawany et al. (2017)	2017	10	101	39	2	N	**Y**	**Y**
Dayal et al. (2016)	2017	11	120	29	3	N	N	**Y**
Yan et al. (2016a)	2016	7	131	29	1	N	**Y**	**Y**
Ashraf and Latif (2014)	2014	6	35	7	1	**Y**	**Y**	N
This work	–	**46**	**216**	**143**	**9**	Y	Y	Y

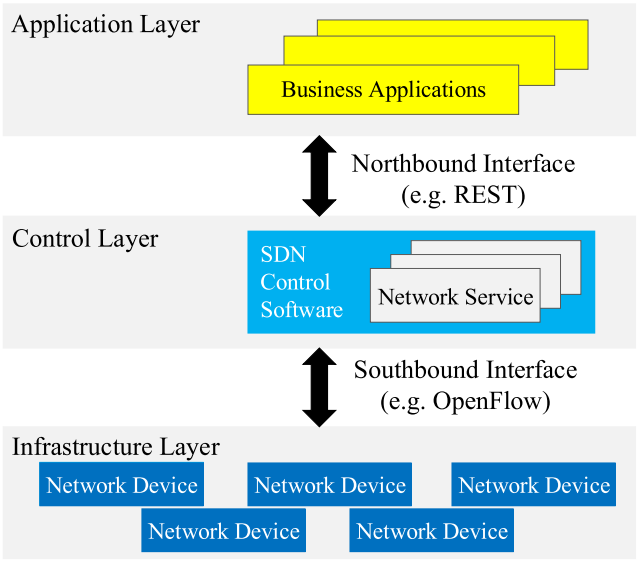


Fig. 1. The SDN architecture.

in Fig. 1. The infrastructure layer comprises network devices, which forward the received packets under the instructions made by the SDN controller in the control layer. There are different SDN applications (e.g. network security applications) in the application layer, which make decisions about how to deal with the packets received by the SDN-enabled switches. The control layer provides the network information to the applications via the northbound interface, translates the decisions made by applications to instructions that can be accepted by the SDN-enabled switches, and performs some basic control and management on the network devices (e.g. managing network topology) via the southbound interface.

Taking advantage of this architecture, the network intelligence is logically decoupled and centralized into the control layer. This architecture also simplifies the network devices. Meanwhile, since the network devices can be controlled via programs running on the controller, the service providers can efficiently operate their network and conveniently develop new network functions on the controller according to their needs.

2.2. Layers of SDN

2.2.1. Infrastructure layer

In SDN, the infrastructure layer is composed of different network devices. A network device is a component that provides the communication pipeline for the packets sent from the terminal devices and other network devices. Essentially, a network device receives packets from its ports and performs necessary actions on these packets. Possible actions include forwarding the received packets via its ports, modifying the received packets (usually on the packet header) and even dropping the received packets. In the traditional network, there are many kinds of network devices, such as the switch, router, hub, access point, repeater, gateway, and bridge. In contrast, the network device used in SDN often refers to the SDN-enabled switch (Anon, 2012b, 2013, 2014; McKeown et al., 2008; Mahmud and Rahmani, 2011; Sood et al., 2015; Detti et al., 2013; Rotsos et al., 2012).

As the most famous SDN organization, ONF has issued six versions of the OpenFlow protocol used in SDN (Anon, 2009, 2011a,b, 2012b, 2013, 2014). ONF issued the first version of OpenFlow switch specification in 2009 (Anon, 2009). Among these versions, the OpenFlow v1.3 is a famous one. At present, it is widely accepted and applied by the academia and industry. Accordingly, the OpenFlow protocol introduced in this work is referred to OpenFlow v1.3 (Anon, 2012b).

The OpenFlow switch specification clearly stipulates the components and basic functions of the SDN enabled switch. In OpenFlow v1.3, an OpenFlow switch consists of three components: (1) one or more flow tables, (2) a group table and (3) an OpenFlow channel. The flow table and group table are designed to perform packet matching and forwarding while the OpenFlow channel is employed to establish the communication channel to the control layer.

The SDN-enabled network devices include the physical network devices and virtual network devices. Due to the network programmability, centralized control, low management complexity, fine-grained network management, and high rate of innovation provided by SDN, it has been regarded as the next generation network architecture which is most likely to displace the traditional network. Accordingly, more and more vendors begin to produce SDN hardware switches. Examples of SDN hardware switches are CISIO Nexus 9000 series switches, JUNIPER EX4600, EX9200 and QFX5100 switches, PICA8 P-3297 switches, NEC QX-S1000, QX-S4100 series switches, BROCADE BCM56960 series switches, EDGECORE AS5710-54X/AS5712-54X switches, ADTRAN SDX 6310 switches, CIENA 5162 switches, and CENTEC v350, v580 switches.

Besides these physical network devices, there are also some virtual network devices utilized in SDN, including the OpenvSwitch, Indigo virtual switch, and CPqD OFSoftswitch, which are open source and widely used by researchers and manufacturers.

Table 2
Description of OpenFlow Messages.

Message	Initiator	Sub-type	Description
Controller-to-switch	Controller	Features	The controller can obtain the capabilities of a switch using this message.
		Configuration	It is designed for the controller to query or set the configuration of a switch.
		Modify-state	The controller uses this kind of message to add, delete or modify the flow entry and group entry in the switch.
		Read-state	It is used for the controller to query the switch state.
		Packet-out	The controller encapsulates some packets in this message and sends this message to the switch to forward the encapsulated packet via a specified port of the switch.
		Barrior	It is used to ensure the processing order.
		Role-request	It is used for the controller to query or set the role of its role.
		Asynchronous-Configuration	The controller uses this message to filter the asynchronous messages that it does not want to receive.
Asynchronous	Switch	Packet-in	The switch encapsulates the received packets that it cannot processed in this message and sends it to the controller.
		Flow-Removed	The switch uses this message to notify the controller a flow entry was deleted.
		Port-status	When a port status of the switch is changed, this message is used to notify the controller the changes of its ports.
		Error	It is used to notify the controller the error occurred in the switch.
Symmetric	controller or switch	Hello	As soon as the TCP connection between the controller and switch is established, the controller and switch both send a hello message to indicate the highest OpenFlow version it can support.
		Echo	It is used to test the liveness or time delay of the connection between the controller and switch.
		Experiment	It is used to achieve functions that OpenFlow protocol cannot support.

2.2.2. Control layer

Different from the conventional network architecture, SDN decouples the control function and forwarding function. Essentially, the control layer mainly focuses on two things: (1) it receives messages sent by network devices and extracts information from these messages to monitor the status of the network, and (2) it provides the abstract network view to the application layer and transfers the instructions made by the application layer to the network devices. Therefore, the control layer needs to monitor the status of the network, such as the real-time network topology of all network devices under its control, availability of all network devices, port status of all network devices, link status of all network devices, bandwidth utilization, and time delay of all links and network devices. Based on these real-time monitoring information, the control layer can dynamically make relevant abstract network views and send these network views to the application layer. Accordingly, some instructions will be created by the application layer and sent to the control layer. Then the control layer can relay these instructions to the underlying network devices to execute these decisions.

Designing and deploying SDN controller has attached attentions of academia and industry. Nowadays, many SDN controllers have already been issued, including the NOX (Gude et al., 2008), POX (Anon, 2021a), OpenDaylight (Anon, 2021b), ONOS (Berde et al., 2014), RYU (Anon, 2021c), Floodlight (Anon, 2021d), FlowVisor (Sherwood et al., 2009), Beacon (Erickson, 2013), OpenMul (Anon, 2021e), OpenContrail (Anon, 2021f) and so on.

2.2.3. Application layer

The application layer is responsible for analyzing, managing, and configuring the network devices in the infrastructure layer via the northbound interface. Meanwhile, it also receives the network information provided by the control layer. The application layer consists of one or more applications, which are software programs that can be easily developed.

According to different scenarios, the applications of the application layer can be summarized as six aspects, which are (1) network security applications, (2) network monitor and analysis applications, (3) network maintenance applications, (4) traffic engineering applications, (5) failover applications and (6) other applications. Specifically, the network security applications include the firewall (Caprolu et al., 2019), DDoS detection system (Cui et al., 2016), intrusion detection and prevention system (Sultana et al., 2019; Chen and Yu, 2016a), network scanning detector (Yuwen et al., 2016), authentication system (Duan and Wang, 2015), access control scheme (Matias et al., 2014), and moving target defense system (Jafarian et al., 2012).

2.3. Interfaces of SDN

2.3.1. Southbound interface

The southbound interface is capable of connecting the control layer and infrastructure layer. The control layer sends some monitoring and managing messages to the infrastructure layer via the southbound interface, while the infrastructure layer packages some messages to report the current network status to the control layer.

As mentioned above, besides specifying the components and basic functions of SDN enabled switch, the OpenFlow switch specification also introduces the OpenFlow protocol, a de-facto southbound interface in SDN. In OpenFlow 1.3.0, there are three types of messages, which are (1) the controller-to-switch messages, (2) asynchronous messages, and (3) symmetric messages. Among these messages, the controller-to-switch messages are provided to the controller to manage or monitor the switch. These messages can only be initiated by the controller and sent to the switch. On the contrary, the asynchronous messages can only be initiated by the switch and sent to the controller to report the changes or events occurred in the switch. Different from these two kinds of messages, the symmetric messages can be initiated by the controller or switch, which are employed to achieve some special functions. For instance, establishing the controller-switch connection, checking the liveness of the controller-switch connection, and providing a staging area for future OpenFlow versions. The detailed descriptions of these three messages are shown in Table 2.

2.3.2. Northbound interface

The control layer provides an abstract network view and a programmable interface for the application layer via the northbound interface. Specifically, taking advantage of the northbound interface, the application layer can obtain the network topology and detailed description of each switch, such as the flow table, group table, port stats, queue description, meter stats, and controller role. Meanwhile, the application layer can also add, modify and delete flow entry, group entry, meter entry, and modify the behavior of ports in the switch and modify the controller role.

Some companies and organizations have already issued a number of northbound interfaces. For instance, the API provided by NOX, POX, and RYU, the REST API supported by most controllers, the tinyNBI that has full compatibility with all versions of OpenFlow, the Frenetic that provides a programming language. However, different from the southbound interface, there is no leading specification in the northbound interface currently.

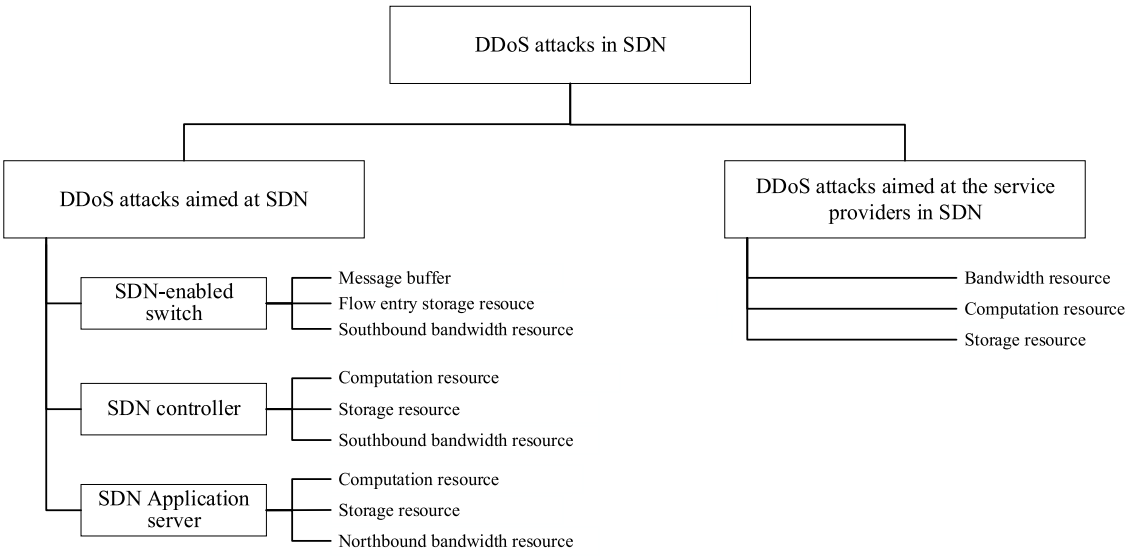


Fig. 2. DDoS attacks in SDN.

3. DDoS attacks in SDN

3.1. DDoS

The DDoS attack tries to overwhelm the available resource of a victim to block the victim from providing service for normal users, by sending massive malicious requests from a huge number of hijacked machines. In general, there are two kinds of requests which can be utilized to initiate DDoS attacks. The first kind is the normal request which is the same as the one sent by normal users. When using this kind of request, the DDoS attackers control a lot of hijacked machines to simultaneously send normal requests, resulting in completely exhausting the bandwidth or computing resource of the victim. The DDoS attackers can also forge some malicious requests based on the application or protocol vulnerability of the victim. Once the victim receives plenty of forged requests, its bandwidth or computing resource will be immediately depleted.

In order to initiate a DDoS attack, the adversary needs to control a lot of devices to send DDoS packets to the target. These machines are called zombie computers or zombies. Thus, the first step of DDoS is scanning, which aims to discover vulnerability of the online devices such as the computers, cellphones, and closed circuit television cameras. In this step, the random scanning, hitlist scanning, permutation scanning, and local subnet scanning can be deployed to locate as many vulnerable devices as possible. Subsequently, according to the scanning results, the adversary can flexibly design different attack plans with strong pertinency to hijack the vulnerable devices. After successfully kidnapping a number of zombies, the adversary can send control messages to these zombies to command them to send malicious requests to the victim to launch a DDoS attack. Many DDoS tools can achieve that purpose, such as the TFN, TFN2K, HULK, Stacheldraht, LOIC, HOIC, Shaft, Trinoo, Knight, and Mstream.

An example of how an adversary named Mallory uses TFN2K to launch DDoS to attack a game server is given to clearly show the entire steps of a DDoS attack. In this example, we assume that the IP address of the game server is 160.18.2.33.

- Step 1. Mallory initiates a random scanning to find the computers with SQL injection vulnerability.
- Step 2. Mallory uses SQL injection to hijack the computers with SQL injection vulnerability.
- Step 3. Mallory uploads .td process to all hijacked computers.
- Step 4. Mallory inputs the message ’./tfn -f hosts.txt -c 8 -i 160.18.2.33’ to command all the hijacked computers to send ICMP/UDP/TCP DDoS traffic to the game server.

Although it seems hard to initiate a DDoS attack due to the difficulty of finding and hijacking zombies, committing a DDoS attack is quite easy nowadays. It is caused by the fact that selling the zombies has already became a complete industry chain and the DDoS attackers can easily buy many zombies. Once the adversaries control these zombies, they can easily launch a DDoS attack using the above DDoS tools.

As an emerging network architecture, SDN has quite a big difference from the traditional network. Specifically, the hierarchical architecture of SDN brings new vulnerabilities that can be exploited by the adversaries to start a DDoS attack. In consequence, the DDoS attacks in SDN are quite different from the DDoS in the traditional network. As shown in Fig. 2, based on the target of DDoS attacks, in this work, the DDoS attacks in SDN are classified into two kinds: (1) DDoS attacks aimed at the SDN network and (2) DDoS attacks targeted at the service providers. Here, the service provider refers to the devices which connect to the SDN network and can provide service for users. Next, we try to give a more detailed description of these two kinds of DDoS attacks in the following subsection.

3.2. DDoS attacks aimed at the SDN network

Recall the previous description of SDN, it is composed of the infrastructure layer, control layer, application layer, southbound interface, and northbound interface. According to the hierarchical architecture of SDN, the DDoS aiming at the SDN network can be classified into three kinds, including (1) DDoS against the infrastructure layer, (2) DDoS targeted at the control layer and (3) DDoS threatened the application layer, as shown in Table 3.

Basically, to clearly differentiate these three kinds of DDoS, it is important to give precise definitions of them. Table 3 lists the differences between these kinds of DDoS. Firstly, the target of the infrastructure layer DDoS attack is to overwhelm one or a few specific network devices in the infrastructure layer. Accordingly, the adversary commands a set of zombies that connect to the target network devices to send DDoS packets. Differently, the DDoS attacks targeted at the control layer try to overwhelm the computation, storage or bandwidth resource of the controller. Thus, the adversary usually makes all or most of network devices send a lot of messages (e.g. *packet-in* messages) to the controller by commanding the zombies to send numerous new

Table 3
DDoS attacks aimed at SDN network.

DDoS type	Target	Example
DDoS against infrastructure layer	SDN enabled network devices	The adversaries try to overwhelm the flow table storage resource of the SDN-enabled switch by sending a number of new packets to the target switch.
DDoS targeted at control layer	SDN controller	The adversaries send a lot of new packets to all or most SDN-enabled switches to overwhelm the computing resource or bandwidth resource of the controller.
DDoS threatened application layer	SDN applications	The adversaries try to overwhelm the specific application by sending specific packets to all or most SDN-enabled switches.

packets to these network devices. For the DDoS attacks threatened the application layer, the target is the application server. Hence, the adversary will command the zombies to elaborate a large amount of specific new packets (e.g. ARP request packets) to all or most SDN-enabled switches. Finally, the application utilized to handle ARP packets receives a lot of *packet-in* messages, resulting in the exhaustion of computation resource, memory resource, and bandwidth resource of the application server. In conclusion, the differences between the DDoS attacks aimed at the infrastructure layer, control layer, and application layer is that they have different attack targets and different attack patterns.

3.2.1. DDoS against the infrastructure layer

When a new packet arrives at the SDN-enabled switch, the switch will look up its flow entries to match that packet. If a flow entry matches that newly arrived packet, that packet will be processed under the guideline of the action field of the matched flow entry. Otherwise, the switch will encapsulate that packet into a *packet-in* message and send that *packet-in* message to the controller. Then the controller will make some instructions to determine how to process that kind of packets in the switches. The most common instructions include sending a relative *packet-out* message and installing flow entries on the corresponding switches.

Although the packet processing method of SDN provides wonderful network programming ability for network managers, it also brings vulnerabilities which may be employed by the adversaries to initiate the DDoS attack against the SDN-enabled switches. In general, the vulnerabilities of an OpenFlow switch include (1) the restricted computation resource caused by low-end CPU resource, (2) the limited storage resource caused by Ternary Content-Addressable Memory (TCAM), and (3) the processing bottleneck caused by OpenFlow. Next, we will clearly describe these vulnerabilities.

Restricted computation resource: Once the switch receives a new packet, it has to look up its flow tables to match that packet. If it cannot match the received packet, it will generate a new *packet-in* message and send that message to the controller. However, due to the constraints of implementation, the switches often utilize low-end CPU as the kernel module to execute these operations. Accordingly, an OpenFlow switch has limited processing ability. For instance, as reported in Wang et al. (2014), the Pica8 Pronto switch can only handle up to 200 flow entry installing rules per second. Therefore, the adversary can send a lot of new packets to a switch to deplete its restricted computation resource.

Limited storage resource: The SDN-enabled switches often utilize TCAM to perform flow entries storing and flow tables looking up to match the newly arrived packets. However, due to the high hardware cost, high power consumption and low storage efficiency of TCAM, the SDN-enabled switches usually have limited storage resources. Generally, an SDN-enabled switch can maintain from a thousand to tens of thousands flow entries (Wu et al., 2016). For instance, as reported in the previous work, the H3C S5820V2 can possess only 3000 flow entries. Therefore, it is easy for an attacker to run out the storage resource of an SDN-enabled switch (Leng et al., 2017).

Processing bottleneck caused by OpenFlow: In general, there are mainly two bottlenecks caused by OpenFlow: (1) the fine-grained control and (2) the resident flow entries. On one hand, different from the traditional network, SDN provides fine-grained control for the network manager. Specifically, in OpenFlow v1.5.1, there are 45 match fields, while 14 of them are required in a flow entry. Unfortunately, the fine-grained control makes the number of flow entries become quite big. On the other hand, OpenFlow utilizes two timeout mechanisms to control how to expire the flow entries, including the *hard_timeout* and *idle_timeout* mechanisms. The *idle_timeout* is usually utilized in the existing works (Wu et al., 2016; Liang et al., 2015). However, by elaborately sending DDoS packets, the adversary can make the flow entries caused by DDoS attacks always stay in the switch.

In conclusion, it is possible for an adversary to initiate a DDoS attack against the infrastructure layer. Fig. 3 illustrates that kind of DDoS attack, which can overwhelm the target switch using the following steps.

- Step 1. The adversary sends control and command messages to a set of zombies, to command these zombies to send packets with time-varying packet headers to a target switch. For instance, the source IP addresses of these packets will be changed using IP spoofing.
- Step 2. The zombies send DDoS packets to the target switch.
- Step 3. The target switch parses these DDoS packets and tries to match these packets with its flow entries.
- Step 4. The target switch cannot match the DDoS packets due to the time-varying packet headers of these packets. Hence, it encapsulates the un-matched packets in *packet-in* messages and sends these *packet-in* messages to the controller.
- Step 5 and 6. The controller parses these *packet-in* messages to get the un-matched packets and sends these packets to related application server.
- Step 7 and 8. The application server decides how to process the un-matched packets and sends the decision to the controller.
- Step 9 and 10. The controller encapsulates one or more OpenFlow messages (e.g. *flow-mod* messages) based on the decision made by the application server and sends these messages to the switch.
- Step 11. The switch adds one or more flow entries based on the received *flow-mod* messages.
- Step 12. The adversary makes the zombies continually send packets that can match the new-added flow entries to make the new-added flow entries exist for a long time at the switch, thus causing the exhaustion of its flow entry storage resource.

3.2.2. DDoS targeted at the control layer

Recall the above description, when a DDoS packet arrives at a switch, the switch will look up its flow entries to match that DDoS packet. If there is no matching flow entry, the switch will encapsulate that DDoS packet into a *packet-in* message and send that *packet-in* message to the controller. After receiving the *packet-in* message, the controller usually stores that *packet-in* message in a buffer. Meanwhile, the controller also maintains one or more programs to pull and handle the stored *packet-in* messages.

However, the above process gives opportunities for the adversaries to launch a DDoS attack targeted at the SDN controller. It is possible for the adversaries to send numerous DDoS packets to make the arriving rate of the *packet-in* messages bigger than the processing speed of *packet-in* messages, which will cause serious problems. For example, when a DDoS attack occurs, there will be a lot of *packet-in* messages

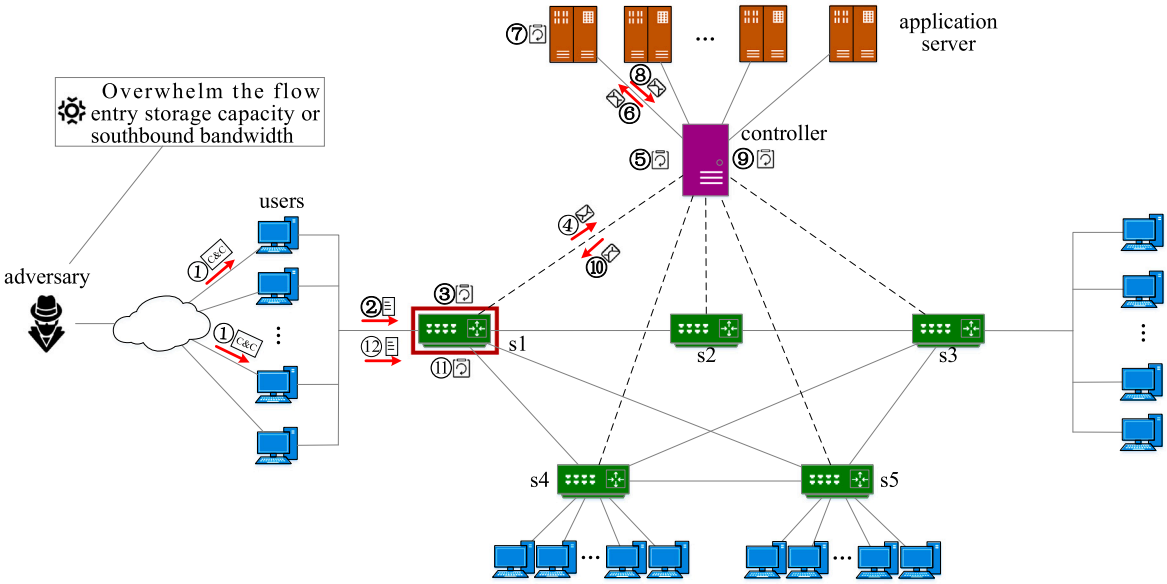


Fig. 3. DDoS attacks against the infrastructure layer.

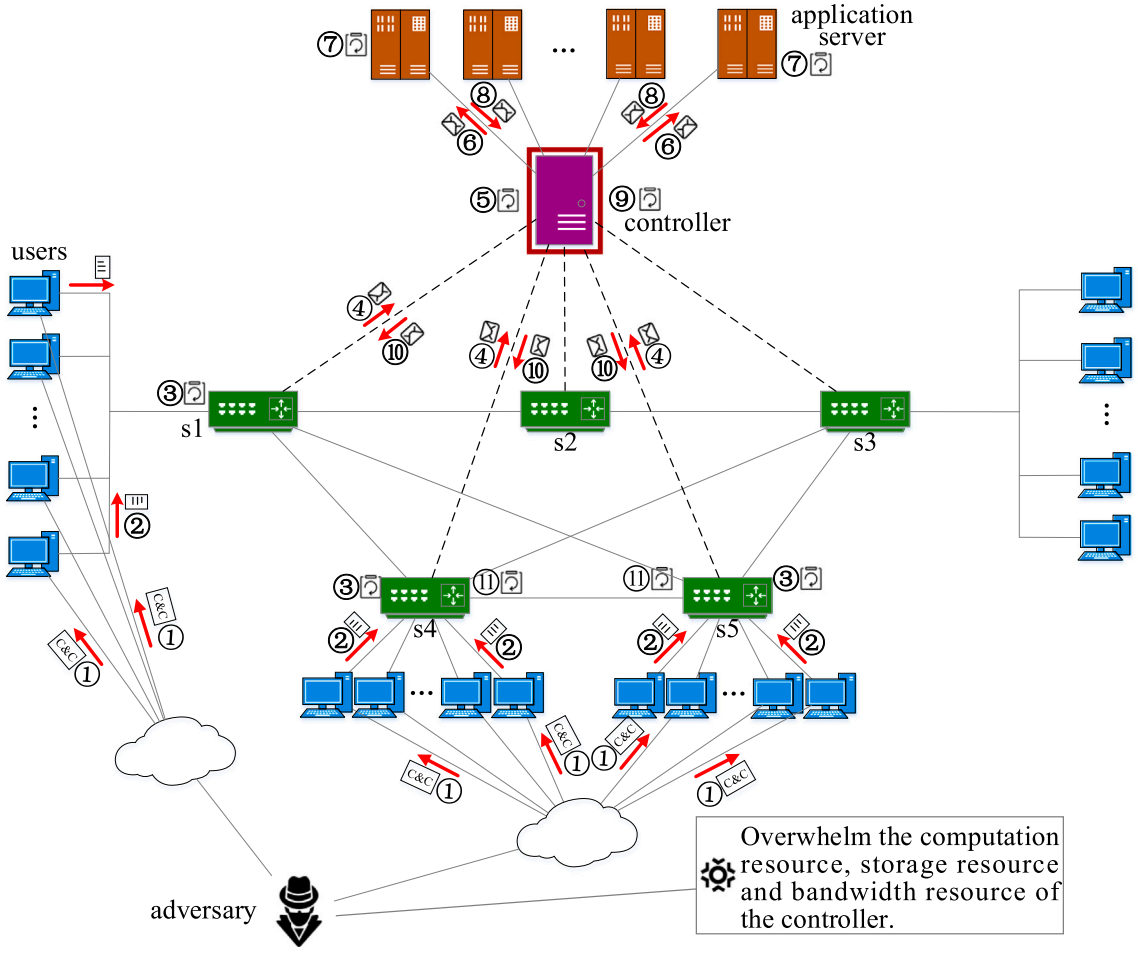


Fig. 4. DDoS attacks targeted at the control layer.

generated by the DDoS attack in the buffer. Therefore, the waiting time of normal *packet-in* messages generated by the benign users will be increased to an unacceptable level, which causes the service not available. More seriously, the adversaries can also crash the storage buffer of the SDN controller.

As shown in Fig. 4, main steps (from step 1 to step 11) of the DDoS attack targeted at the SDN controller are the same with the DDoS attack against the infrastructure layer. But these are some differences between these two kinds of DDoS attack. Here, we only list these differences: (1) At step 1, the adversary commands the zombies send an extremely large

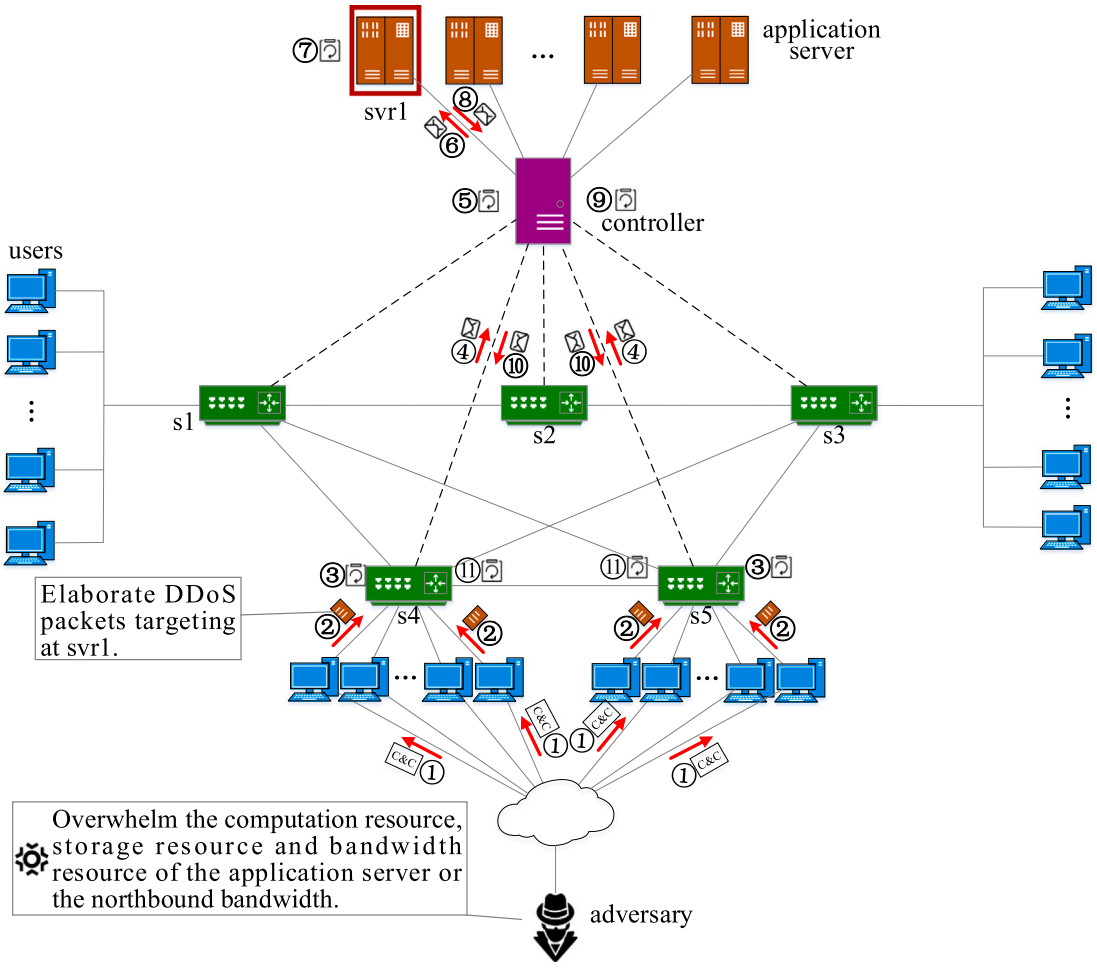


Fig. 5. DDoS attacks threatened the application layer.

number of packets with time-varying packet headers to many switches, instead of one target switch. (2) From step 3 to step 11, the numerous *packet-in* messages will be encapsulated and sent to the controller from related switches. (3) In the DDoS attack aimed at the SDN controller, the adversary does not need to make the new-added flow entries exist for a long time at the switch. Hence, the adversary does not make the zombies send packets matched the new-added flow entries.

3.2.3. DDoS threatened the application layer

In SDN, the network analysis and management are usually achieved by the applications in the application layer. These applications employ the northbound interfaces to communicate with the SDN controller. Recall the above descriptions about northbound interfaces developed in SDN, the API, REST API, and specially designed interfaces (e.g. tinyNBI and Frenetic) are the most popular northbound interfaces utilized in SDN. These different northbound interfaces make the applications running on different devices. For instance, when using the API as the northbound interface, the applications will be running on the controller. When using other northbound interfaces, the applications always run on one or more independent servers.

However, no matter whether the applications are running on the controller or independent servers, the target of DDoS attacks threatened the application layer is the specific application. In other words, in this kind of attack, the adversary tries to overwhelm an application that achieves a specific function to indirectly crash the whole SDN network. For instance, the Address Resolution Protocol (ARP) proxy is usually utilized to avoid the broadcast storm generated by loop links in SDN (Cho et al., 2015). The ARP proxy is always implemented as an application. Hence, an adversary can overload that ARP proxy application to break the ARP proxy service, causing the whole SDN network to break.

Fig. 5 illustrates how to launch a DDoS attack threatened the application layer. Main steps of initiating such a DDoS attack are the same with the DDoS attack targeted at the SDN controller. Similar to the first step of the DDoS attack targeted at the control layer, the adversary must control some zombies that connect to the target network. Then, the adversary also simultaneously commands these zombies to send DDoS packets. However, these DDoS packets are not the same with the one used in DDoS attack targeted at the control layer. In the DDoS attack threatened the application layer, the DDoS packets are elaborately generated by the zombies to make the SDN controller send these attack packets to one application server, while there is no such restriction for the DDoS packets utilized in the DDoS attack targeted at the SDN controller. Again, taking the DDoS attack aimed at the ARP proxy application as an example, the DDoS packets used in this attack are ARP requests or ARP responses with a spoofing IP address. Consequently, these ARP requests or ARP responses will be encapsulated into the *packet-in* messages and sent to the controller. The controller will send these *packet-in* messages to the ARP proxy application. At last, the ARP proxy application will be crashed.

3.3. DDoS attacks aimed at the service providers in SDN

Same as the DDoS attacks in the traditional network, the DDoS attacks aimed at the service providers also exist in SDN. Considering that a large proportion of services are provided by servers, in this

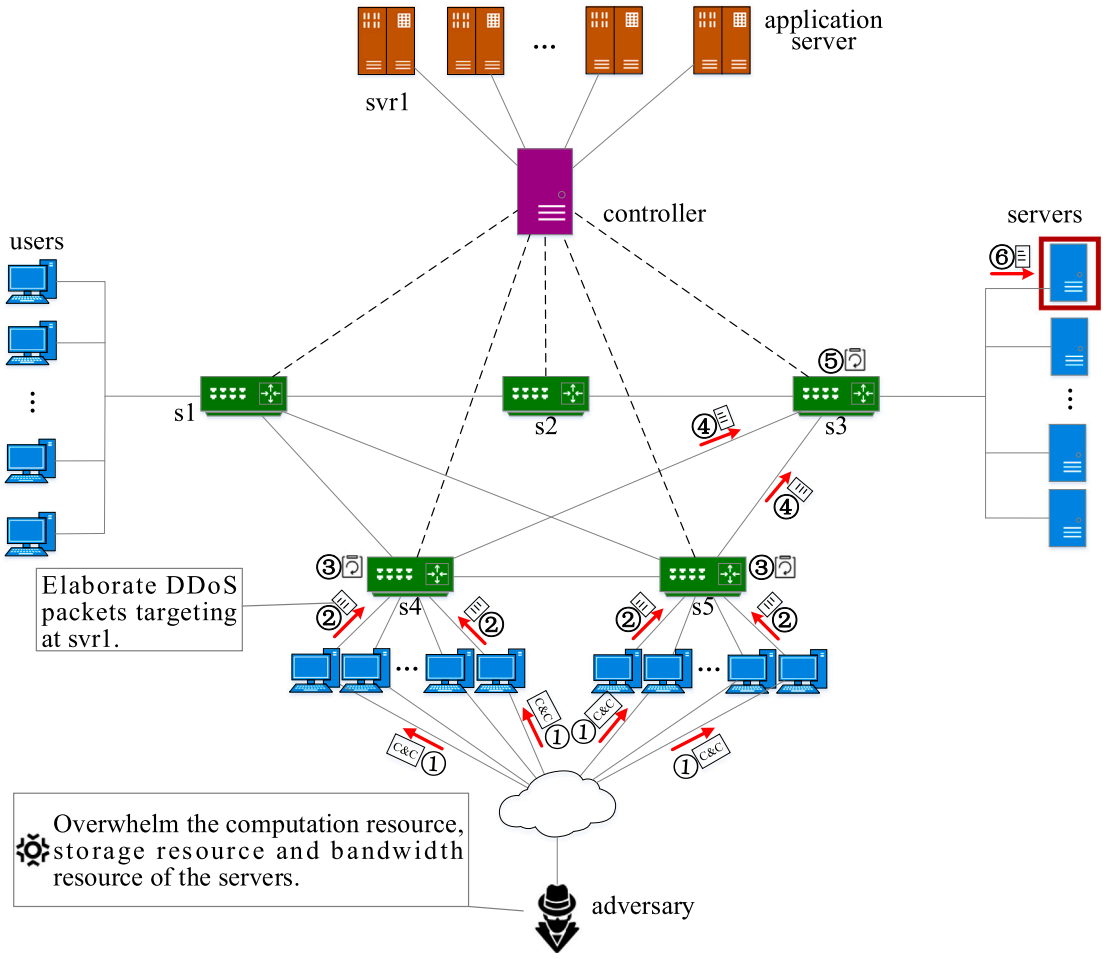


Fig. 6. DDoS attacks aimed at the service providers.

work, the service providers refer to servers. Utilizing the DDoS attacks aimed at the service providers, the adversary attempts to crash the corresponding servers, and eventually makes the service unavailable. As shown in Fig. 6, in step 1 and 2, the adversary also commands the zombies to send DDoS packets. Different with the DDoS attack aimed at SDN, the DDoS packets are not required to be un-matched packets. From step 3 to step 6, these DDoS packets will be transmitted to the target server, which will overload the target server.

Although there are many kinds of taxonomies about DDoS attacks aimed at the service providers (Somani et al., 2017; De Donno et al., 2017; Bhardwaj et al., 2016), in this section, we mainly describe these DDoS attacks based on the way chosen by the adversaries to launch a DDoS attack. In essence, for the purpose of crashing servers, the adversary can choose to overload the bandwidth or deplete the resource of servers. Therefore, the DDoS attacks aimed at the service providers in SDN are further classified into the bandwidth depletion DDoS attack and server resource depletion DDoS attack.

3.3.1. Bandwidth depletion DDoS attack

According to the previous works, the bandwidth depletion DDoS can be further classified into flood attacks and amplification attacks. For the flood attack, an adversary can command the zombies to send plenty of normal packets to the target server. The representative flood attacks are UDP flood attack and ICMP flood attack (Kolahi et al., 2015; Chauhan and Saini, 2015), where the UDP and ICMP packets are employed to congest the bandwidth of the target server.

Conversely, in the amplification attack, instead of commanding the zombies to directly send DDoS packet to the target server, the adversary makes the zombies send numerous requests to some specific servers such as the DNS servers or NTP servers. The source IP addresses of these requests are set as the IP address of the target server. After receiving these requests, these specific servers will generate the response packets and send these packets to the target server, which will significantly increase the attack rate and congest the target server's bandwidth. The typical amplification attacks are the Smurf attack and Fraggle attack (Bouyeddou et al., 2018; Deshmukh and Devadkar, 2015).

3.3.2. Server resource depletion DDoS attack

In contrast to the bandwidth DDoS attacks, the resource depletion DDoS attacks try to exhaust the server resource by using vulnerabilities of network protocols or applications. Based on the primary vulnerabilities exploited to launch the DDoS attack, the server resource depletion DDoS attacks can be classified into two kinds: (1) protocol exploitation DDoS attacks and (2) malformed packets exploitation DDoS attacks.

In protocol exploitation DDoS attacks, the adversary exploits the implementation bugs or specific functions of a network protocol to deplete the available resources of the target server. A typical protocol exploitation DDoS attack is the TCP SYN attack (Kumar et al., 2018a), which exploits the weakness of the three times handshake when establishing a TCP connection between two terminals. By sending numerous SYN packets, the target server will rapidly run out of computation and storage resources. Thus it cannot provide services for legitimate requests. For starting a malformed packets exploitation DDoS attack, the adversary intentionally elaborates some malformed packets to consume the available resource of the target server. For instance, the adversary can forge a packet by setting the same source IP address and destination IP address in that packet and send that malformed packet to the target server, which will cause the victim to crash.

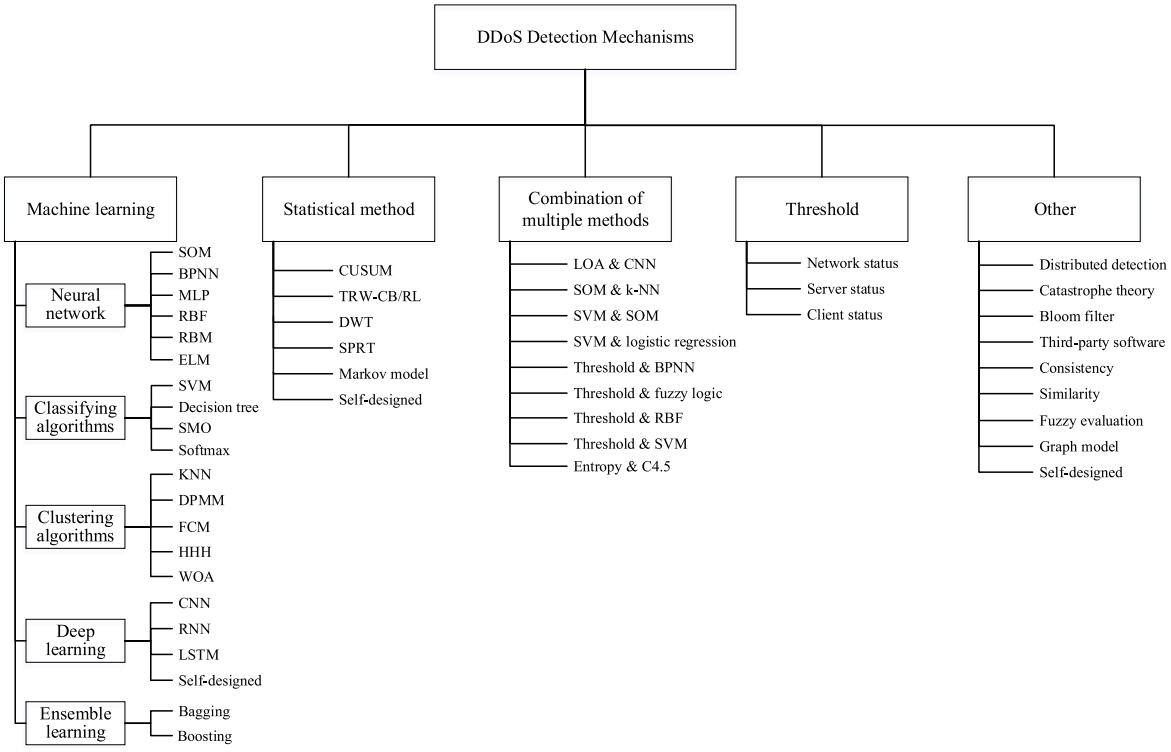


Fig. 7. Classification of DDoS detection mechanisms in SDN.

In conclusion, the DDoS attack is an unavoidable security threat in SDN. As an essential technology used to detect DDoS attacks, DDoS detection mechanisms attract plenty of academic interests. More precisely, in this work, 143 related papers that provide DDoS detection mechanisms used in SDN are reviewed. These papers were published from 2010 to 2020. These DDoS detection mechanisms are described in this section. Based on the method utilized in these mechanisms, these DDoS detection mechanisms are classified into the machine learning-based, statistics-based, combination of multiple methods-based, threshold-based, and other method-based DDoS detection mechanisms, as shown in Fig. 7. In the following sections, we will clearly describe these kinds of DDoS detection mechanisms one by one.

4. Machine learning-based DDoS detection mechanisms

In the 143 published DDoS detection mechanisms, 70 (about 49%) mechanisms are designed based on the machine learning method, which means that the machine learning-based DDoS detection mechanism is one of the most popular DDoS detection mechanisms. According to the machine learning method employed in these mechanisms, we further classify machine learning-based DDoS detection mechanisms into neural network-based, classifying-based, clustering-based, deep learning-based, and ensemble learning-based DDoS detection mechanisms, as shown in Fig. 7.

4.1. Neural network-based DDoS detection mechanisms

In machine learning technologies, the neural network consists of a set of algorithms, which loosely models the human brain. In this work, various neural networks including the Self Organizing Maps (SOM) (Braga et al., 2010; Phan and Park, 2019; Phan et al., 2016, 2017; Wang and Chen, 2017; Xu and Liu, 2016; Zhao and Liu, 2018; Nam et al., 2018; Pillutla and Arjunan, 2019; Phan et al., 2019), Back Propagation Neural Network (BPNN) (Cui et al., 2016; Chen and Yu, 2016a; Wang et al., 2019; Cui et al., 2018; Mihai-Gabriel and Victor-Valeriu, 2014; Liu et al., 2019), Multi-Layer Perception (MLP) (Santos et al., 2020; Gharvirian and Bohlooli, 2017; Wang et al., 2020a), Radial Basis Function (RBF) (Chen and Yu, 2016b; Dayal and Srivastava, 2018), Restricted Boltzmann Machine (RBM) (MohanaPriya and Shalinie, 2017), and Extreme Learning Machine (ELM) (Gong et al., 2019) have been considered as the basic methods when designing the DDoS detection mechanisms.

4.1.1. SOM-based DDoS detection mechanisms

A lightweight DDoS detection method has been proposed in Braga et al. (2010). It consists of the flow collector, feature extractor, and classifier. The flow collector is responsible for gathering the flow entries from the switches by periodically sending status request messages from the controller to the switches. The feature extractor receives these gathered flow entries and extracts the average of packets per flow, average of bytes per flow, average of duration per flow, percentage of pair-flows, growth of single-flows, and growth of different ports. A SOM model is trained and used to detect the DDoS attacks based on these features in the classifier. In this work, Braga et al. extracted the flow features from the flow entries existing in OpenFlow switches, instead of calculating flow features from the captured packets, which leads to a new research hotspot of how to develop new DDoS detection mechanisms using SDN features.

By combining SOM and SVM, Phan et al. proposed a new hybrid machine learning model (Phan and Park, 2019). The flow duration and packet number are firstly recorded and sent to the SVM classifier. If a flow is recognized as suspicious, then the flow duration, packet number, byte number, and protocol of that flow are calculated and detected by a SOM classifier. Accordingly, the SOM detection results are gained and recorded as O_{i1}. Otherwise, if the SVM classifier identifies a flow as normal, that flow will be forwarded to an enhanced History-based IP Filtering scheme (eHIPF). In each observation, the eHIPF forecasts a score X_i for the next observation. When detecting DDoS, based on the collected features in the current observation, the actual score X_j will be computed. In case that $X_i < X_j$, eHIPF will report that it detects a DDoS attack and records the detection result as O_{i2}. At last, if O_{i1} and

O_{i2} both indicate that there is a DDoS attack, the final detection result is made as True. Otherwise, it is made as False.

In Phan et al. (2017), another SOM-based DDoS detection method, named DSOM, was proposed by Phan et al. DSOM consists of some extension modules including the flow collector, feature extractor, training database, DSOM map, fast policy enforcement, and DSOM switch agent. Among these modules, the DSOM map executes the same operations as a single SOM. The number of flows, number of packets per flow, number of bytes per flow, duration, growth of client ports, and protocol are extracted and transmitted to train a DSOM model. Then the DSOM model can detect a DDoS attack. Once a DDoS attack is identified, the DSOM map sends the anomalous client information to the DSOM switch agent. After that, the DSOM switch agent will send these information to the fast policy enforcement module and the controller to perform further actions.

A novel DDoS defense mechanism named SGuard was proposed in Wang and Chen (2017). SGuard contains three modules: the access control module, classification module, and data plane cache module. The classification module is responsible for detecting the DDoS attacks. It is composed of a data collector, a feature extractor, and a classifier. The flow entries of each switch will be collected by the controller at predetermined time intervals. Subsequently, the feature extractor calculates the percentage of flows with a small number of packets, percentage of flows with small average bytes, percentage of flows with short time duration, percentage of reversible flows, growth rate of irreversible flows, and growth rate of ports. The SOM is utilized as the classifier. The SOM model contains an input layer and a competitive layer, which can classify the flow entries into anomalous and normal ones.

In Xu and Liu (2016), Xu et al. summarized the DDoS detection as two procedures: victim detection and attacker detection. Before detecting victim and attacker, in order to maximize the coverage and minimize the granularity of DDoS detection, the IP addresses with the common prefix constitute an IP range. Meanwhile, each pair of source and destination IP ranges are cognized as a flow. Then the SOM model can be utilized to estimate the victim likelihood of each IP range. If an IP range has a low likelihood of being a victim, that IP range will be extended. When detecting the DDoS attacker, the SOM model can identify whether an attacker is in an IP range. If yes, that IP range will be further detected to find the exact IP address of the DDoS attacker. By dynamically adjusting the scope of IP ranges, the proposed DDoS detection method can adaptively capture the traffic rate feature and traffic rate deviation/asymmetry feature, which makes it efficiently detect DDoS attack.

A DDoS attack detection method designed based on SOM has been proposed by Zhao et al. in Zhao and Liu (2018). In that method, the occurrence probability $p(v_a)$ of the *packet-in* messages is firstly calculated. If $p(v_a)$ reaches to a certain threshold, the number of source IP address, source IP address generating speed, destination port number, ports generating speed, and median of bytes per flow will be extracted as features and passed to the SOM model. Then the SOM model can classify whether these features belong to a DDoS attack. The most important contribution of this work is that the authors employ the occurrence probability of the *packet-in* messages as the DDoS detection trigger.

In Nam et al. (2018), Nam et al. presented an SOM-based DDoS detection method. In their method, four additional modules including the monitor, algorithm, alert, and mitigation module are introduced. The monitor module gains information from the OpenFlow switches and calculates five features: the entropy of source IP address, entropy of source port, entropy of destination port, entropy of packet protocol, and total number of packets. Then these features are normalized and used as the weight vectors of a neuron in SOM. By using SOM as a layer between KNN and the original data, an algorithm combining the SOM and KNN is presented and used as the classifier. Except that, instead of finding the Best Matching Units (BMUs) for every input sample, a SOM with a center-distributed classification mechanism that only needs to calculate the distances between each input instance and a universal reference is proposed. The experimental results show that these two proposed mechanisms both can decrease the processing time, with a little decrease in the detection rate and false positive rate.

Pillutla et al. proposed a fuzzy SOM-based DDoS mitigation mechanism in Pillutla and Arjunan (2019), which is named as FSOMDM. In FSOMDM, the weight sum rules in traditional SOM are replaced by the fuzzy rules. Five traffic flow features including the mean number of packets per flow, mean number of bytes per flow, mean duration time per flow, mean percentage of flow pairs, and rate of growth per flow are calculated and passed to the FSOMDM. Then it can classify the related traffic into malicious traffic and normal traffic.

In Phan et al. (2019), Phan et al. utilized SOM to detect low-rate DoS in SDN. Specially, in the proposed method, the Q-learning is employed to select features used in DDoS detection. After that, the SOM, SVM, and Random forest model can identify low-rate DoS attacks. The performance of the proposed method is tested using MaxiNet, OpenvSwitch, and ONOS. The results indicate that the proposed method performs well in terms of precision, recall, accuracy, F-score, and false alarm rate.

4.1.2. BPNN-based DDoS detection mechanisms

In Cui et al. (2016), Cui et al. proposed a Software-Defined Anti-DDoS (SD-Anti-DDoS) mechanism against DDoS attacks in SDN. In SD-Anti-DDoS, the flow entries are periodically gathered from the switches. Five features including the number of packets, number of bytes, survival time, packet rate, and byte rate of each flow entry are extracted. After that, a BPNN that contains five neurons in the input layer, ten neurons in the hidden layer and one neuron in the output layer is used to detect DDoS attack. The evaluation results demonstrate that the proposed detection mechanism can efficiently detect TCP/SYN flood, UDP flood, and ICMP flood attack.

Chen et al. designed a DDoS attack detection method based on BPNN (Chen and Yu, 2016a). The main idea of this method is to make the switches act as neurons of BPNN. Similar to BPNN, the established DDoS detection model consists of an input layer, several hidden layers, and an output layer. The switches selected as the input layer neurons monitor the passing network traffic and extract features from these traffic. Simple mathematical operations are executed at the switches which are chosen as the hidden neurons. The final decision is made by the switches of the output layer. The performance of the proposed method is measured in terms of detection accuracy, communication overhead, and computation overhead, which shows that it can achieve about 90% detection rate, with an acceptable level of communication and computation overhead.

In Wang et al. (2019), a Safe-Guard Scheme (SGS) has been proposed, which aims to protect the control plane of SDN from the DDoS attack. In SGS, the DDoS attack detection is executed on the switches. More specifically, three modules including the feature extracting, detection reacting, and results executing are designed. Based on the features extracted by the first module, the trained BPNN is able to detect DDoS attacks. It should be noted that after detecting the DDoS attack, the switches connected to the attack source will be migrated to another controller. That controller will send *packet-out* messages to those switches to block the DDoS attack.

In Cui et al. (2018), a DDoS detection mechanism named TDDAD is designed based on the time feature. TDDAD periodically collects flow entries from the switches and extracts the duration, packet count of each flow entry, and the number of flow entry of each port as the input instance of the trained BPNN. Then the BPNN model can judge whether the input instance is abnormal. If the number of abnormal flow entry exceeds the predefined threshold, a DDoS attack is detected. Considering the complexity of updating the BPNN model, the author uses the static BPNN model in TDDAD. In evaluation, the DARPA 1999 Intrusion Detection Data Set was used to test TDDAD. The evaluation

results show that TDDAD can detect all six kinds of DDoS attacks with an error rate at 0.

In Mihai-Gabriel and Victor-Valeriu (2014), Mihai-Gabriel et al. proposed a DDoS detection mechanism, which mainly consists of a Floodlight controller and a server that performs traffic monitoring and risk calculating based on BPNN. The BPNN model used in this mechanism has two input layers, ten hidden layers, and one output layers. Taking advantaging of that BPNN model, the proposed mechanism can calculate the risk of being attacked by a DDoS. Once the calculated risk is recognized as high, the controller will turn to the proactive mode, where no flows will be sent to the controller.

4.1.3. MLP-based DDoS detection mechanisms

In Santos et al. (2020), Santos et al. introduced MLP to DDoS detection. In their work, the MLP-based DDoS detection is implemented and evaluated. The GridSearch are employed to find the best hyper-parameters set. Meanwhile, the feature values are standardized to avoid the overfitting of the MLP model. However, the evaluation results show that although MLP performs well, its detection precision is a little worse than Random Forest and Decision Tree.

Gharvirian and Bohlooli also proposed an MLP-based DDoS detection mechanism (Gharvirian and Bohlooli, 2017). In their mechanism, the flow initiation rate is calculated in determined time intervals and compared with a threshold. Meanwhile, the number of packets of each flow is counted and recognized as a data set. The fast entropy of that data set is calculated and compared with another threshold. If both compared results show the existence of a DDoS, the attack is confirmed. If both results reject the existence of the DDoS, no attack is detected. If one of the compared results show the existence of the DDoS while the other one not, the MLP-based detection will be further executed. The packet count per flow, byte count per flow and flow duration were calculated and sent to the MLP model to further identify the DDoS attack.

In Wang et al. (2020a), an approach has been presented by Wang et al. to detect DDoS attacks in SDN. The main idea of this work is utilizing Sequential Backward Selection (SBS) to select optimal features and establish an MLP model to detect DDoS attacks. A training dataset and a feedback dataset are designed to dynamically train the MLP model. Once the number of new labeled samples in the feedback dataset exceeds a pre-defined value, the MLP model will be retrained. Subsequently, every feature is temporally removed from the feature set to calculate its feature saliency and find the optimal feature set. The proposed approach was evaluated on Matlab, in terms of detection accuracy, precision, detection rate, and false alarm rate. According to the outcomes obtained from the NSL-KDD dataset, the proposed approach can achieve quite an acceptable performance with 97.66% on detection accuracy, 94.88% on detection rate, and 0.62% on false alarm rate.

4.1.4. RBF-based DDoS detection mechanisms

Chen et al. proposed a collaborative intrusion detection system against DDoS in SDN (Chen and Yu, 2016b). In the proposed system, the SDN-enabled switches are recognized as one or more neurons of the RBF model. The RBF model has three layers including the input layer, hidden layer, and output layer. The neurons of the input layer are responsible for monitoring the flows passing through the related switches. Meanwhile, the neurons of the hidden layer receive the preprocessed results sent by the input neurons and execute the activation functions on these results. At last, the linear mapping operation is carried out on the neurons of the output layer and the final detection results will be calculated by these output neurons. The authors modified the source code of OpenvSwitch to implement the proposed system. The detection accuracy and communication overhead of the proposed system have been evaluated, which demonstrated its effectiveness.

In Dayal and Srivastava (2018), the flow entries of each switch is fetched every 10 s. Then the entropy of all destination IP addresses is calculated. If that entropy is less than a predefined threshold, a more detailed DDoS detection designed based on RBF will be executed. In this work, the training of the RBF network is optimized using Particle Swarm Optimization (PSO) algorithm. The average packets per flow, average bytes per flow, number of flows per second, entropy of destination IP addresses per second, entropy of source IP addresses per second, and entropy of IP protocol per second are extracted and sent to the trained RBF model to detect the DDoS attack.

4.1.5. RBM-based DDoS detection mechanisms

In MohanaPriya and Shalinie (2017), an RBM-based DDoS detection method has been presented. The proposed method consists of the data collection phase and attack detection phase. In the attack detection phase, the hit count H_c and energy consumption rate $E_{con}(SW)$ are firstly calculated. For a specific MAC address, if H_c is higher than the average threshold value and $E_{con}(SW)$ is higher than another threshold, the RBM model will be initialized to further detect the DDoS attack. The evaluation was performed on Mininet and POX, which shows that the proposed method can achieve about 92% detection accuracy with an 8% false positive rate.

4.1.6. ELM-based DDoS detection mechanisms

An Intelligent Trust Model (ITM) used for detecting hybrid DDoS has been proposed in Gong et al. (2019). In ITM, ELM is employed as the detection algorithm. The authors choose the ingress port, source MAC/IP address, destination MAC/IP address, counters, action, and some VLAN characteristics as the input features of the ELM model. Then the ELM model can map these features to normal traffic, UDP flood, ICMP flood, SYN flood, slow attack, and http attack. Hence, instead of just judging whether there are DDoS attacks, ITM can also identify the types of the current DDoS attack, which gives the network manager a more precise view of the DDoS attack.

4.1.7. Discussion of neural network-based DDoS detection mechanisms

Neural network-based DDoS detection mechanisms play quite an important role in detecting DDoS in SDN. Generally, using a neural network to detect DDoS includes two processes: the training process and testing process. In the training process, the neural network model is trained and built using the training dataset. In the testing process, the trained neural network model can be utilized to identify the DDoS attack. 23 DDoS detection mechanisms designed based on the neural network have been published. Table 4 lists the summary of neural network-based DDoS detection mechanisms. Among these mechanisms, the most popular neural networks are the SOM and BPNN. In this work, DR, FA, DA, PRE, F1, FN, AUC refer to the detection rate, false alarm rate, detection accuracy, detection precision, F1 score, false negative rate, and area under curve. Meanwhile, the network topologies are classified to small network (less than 24 switches), middle network (from 25 switches to 49 switches), and large network (larger than 50 switches).

The neural network-based DDoS detection mechanisms perform well in the detection rate and detection time. For instance, the mechanisms proposed in Braga et al. (2010), Wang and Chen (2017) and Nam et al. (2018) can achieve the detection rate at 99%, while the detection time stays at the millisecond level. Although the neural network-based DDoS detection mechanisms can achieve excellent detection performance, there are some obstacles that need to be overcome when using the neural network to detect DDoS. The most serious one is that some important parameters significantly affect the performance of DDoS detection, such as the number of layers, number of neurons of each layer, and learning rate. Before using a neural network to detect DDoS attack, these parameters should be determined.

Table 4
Summary of neural network-based DDoS detection mechanisms.

Reference	Year	Location	Method	Attack type	Experiments	Scale	Detection results
Braga et al. (2010)	2010	Controller	SOM	TCP/UDP/ICMP flood	Simulation	Small(3 switches)	DR:99.11, FA:0.46
Phan and Park (2019)	2019	Controller	SOM, SVM	TCP/ICMP flood	Experiment	Laboratory LAN	DR:99.27, DA:99.3, FA:0.67
Phan et al. (2016)	2016	Controller	SOM, SVM	TCP/ICMP flood	–	Small(1 switch)	DR:98.13, DA:97.6, FA:3.85
Phan et al. (2017)	2017	Controller and switch	SOM	TCP/UDP/ICMP flood	Experiment	Small(4 switches)	–
Wang and Chen (2017)	2016	Controller	SOM	TCP flood	Simulation	Small(2 switches)	DR:99.77
Xu and Liu (2016)	2016	Controller	SOM	–	Simulation	Internet2's network topology	–
Zhao and Liu (2018)	2018	Controller	SOM	–	Simulation	–	–
Nam et al. (2018)	2018	Controller	SOM, KNN	–	Experiment	Small(1 switch)	DR:98.24, FA:2.14
Pillutla and Arjunan (2019)	2019	Controller	SOM	TCP/UDP/ICMP	–	–	DR:98.4, FA:16
Phan et al. (2019)	2019	Application Server	SOM, SVM, Random forest	Low-rate DoS	Simulation	Small(1 switch)	–
Cui et al. (2016)	2016	Controller	BPNN	TCP/UDP/ICMP flood, mixed flood	Simulation	Middle(25 switches)	–
Chen and Yu (2016a)	2016	Controller	BPNN	TCP/UDP/HTTP flood	Simulation	Large(50 switches)	DR:92.43, FA:0
Wang et al. (2019)	2019	OpenFlow switch	BPNN	TCP/UDP/ICMP flood	Simulation	Small(1 switch)	–
Cui et al. (2018)	2018	Controller	BPNN	ipsweep, smurf, neptune	–	–	–
Mihai-Gabriel and Victor-Valeriu (2014)	2014	Application server	BPNN	TCP/ICMP flood, open relay DDoS, reflected DDoS	Simulation	Small(1 switch)	–
Liu et al. (2019)	2019	Switch and controller	BPNN, Threshold, PSO	UDP/TCP/ICMP flood	Simulation	Small(6 switches)	DR:97.47, FA:1.43, DA:98.02
Santos et al. (2020)	2018	Controller	MLP, CART, SVM, and Random forest	TCP/UDP/ICMP/HTTP flood	Simulation	Small(1 switch)	–
Gharvirian and Bohlooli (2017)	2017	Controller	MLP, Entropy	Self-designed DDoS	Simulation	Small(9 switches)	DR:95.23, FA:2
Wang et al. (2020a)	2020	Application server	MLP	–	Simulation	–	DR:99.65, DA:99.62, FA:0.41
Chen and Yu (2016b)	2016	Switch	RBF	ICMP flood	Simulation	Large(50–100 switches)	DR:96.3, FA:3.4
Dayal and Srivastava (2018)	2018	Controller	RBF, PSO, and threshold	TCP/UDP/ICMP/HTTP flood, smurf	Simulation	Claranet topology	DA:99.83
MohanaPriya and Shalinie (2017)	2017	Controller	RBM	–	Simulation	Small(1 switch)	DR:92, FA:8
Gong et al. (2019)	2019	Switch and controller	ELM	TCP/UDP/ICMP/HTTP flood, low-rate DDoS	Simulation	Small(10 switches)	DA:98.13

4.2. Classifying-based DDoS detection mechanisms

In this section, we focus on the review and analysis of the classifying-based DDoS detection mechanisms in SDN. The classifying algorithms, which aim to achieve the classification of the testing data, are one of the most essential research areas in machine learning. For the DDoS detection in SDN, the classifying algorithms are often utilized as classifiers to identify the DDoS attack. In our survey, the Support Vector Machine (SVM) (Phan and Park, 2019; Phan et al., 2019; Santos et al., 2020; Rahman et al., 2019; Bakker et al., 2018; Musumeci et al., 2020; Cui et al., 2019; Myint Oo et al., 2019; Ye et al., 2018; Yu et al., 2018; Hu et al., 2017; He et al., 2018; Yang and Zhao, 2018; Latah and Toker, 2018; Shang et al., 2017; Shen et al., 2020; Oo et al., 2017; Li et al., 2015; Kokila et al., 2014; Liu et al., 2017a; Chen et al., 2017; Mehr and Ramamurthy, 2019; Mowla et al., 2018; Polat et al., 2020), Decision Tree (Santos et al., 2020; Hyder and Lung, 2018), Sequential Minimal Optimization (SMO) (Alshamrani et al., 2017), and Softmax (Han et al., 2018) are the commonly used classifying algorithms when detecting DDoS attack in SDN.

4.2.1. SVM-based DDoS detection mechanisms

Cui et al. presented a DDoS detection and defense mechanism in Cui et al. (2019). The proposed mechanism is composed of the statistics collection module, feature computing module, DDoS attack detection module, and DDoS attack defense and recovery module. Among these modules, the statistics collection module is responsible for periodically collecting the flow entries from the switches. Then the feature computing module calculates the entropy of source address and destination address from these flow entries. Subsequently, the DDoS attack detection module trains the SVM model by some samples selected from the above entropies and identifies the DDoS attack using the trained SVM model. At last, the DDoS attack defense and recovery module discard the DDoS packets and recover the normal communication.

An Advanced Support Vector Machine (ASVM) based DDoS attack detection mechanism has been designed by Myint Oo et al. in Myint Oo et al. (2019). The traffic generation, traffic data collection, features extraction, and classification of attack or normal by using ASVM are the main four modules of the proposed mechanism. The traffic generation module creates the normal traffics, UDP flooding attack, and SYN flooding attack. The traffic data collection module periodically sends *flow-stats-request* messages to the switches. Then five features including the average number of flow packets, average number of flow bytes, variation of flow packets, variation of flow bytes, and average duration of traffics are obtained by the feature extract module. When utilizing ASVM to detect the DDoS attack, the linear kernel and one-versus-some decision function are employed. The experiment was performed using OpenDaylight and Mininet. The proposed mechanism is shown to provide an average detection accuracy of 97% with an average false alarm rate of 2%.

In Ye et al. (2018), a DDoS attack detection method is introduced by Ye et al. They focus on designing an SVM-based DDoS detection mechanism. The proposed mechanism comprises three steps, i.e., flow table collection, characteristic values extraction, and classifier. Six-tuple characteristic values including the speed of source IP addresses, speed of source port, speed of flow entries, standard deviation of flow packets, standard deviation of flow bytes, and ratio of pair flows are extracted from the flow entries and utilized to train and test the SVM model. The simulation was performed on the Floodlight and Mininet. As per the results, the proposed mechanism can achieve an acceptable average detection accuracy rate (95.24%) and a low average false positive rate (1.26%).

Yu et al. presented an efficient DDoS attack detection and rapid response framework in Yu et al. (2018). The main idea of their work is to design a feature selection algorithm based on the correlation measure technology. More specifically, the features designed by the authors include the average number of packets per flow, average number of bytes per flow, rate of flow table entries, percentage of pair flows, ports generating speed, entropy of source IP address, entropy of destination IP address, and entropy of flow count. These features are ranked and selected in the descending order for the TCP, UDP, and ICMP protocols, respectively. According to the selected features, the flow entries caused by the DDoS attack can be identified. The effectiveness of the proposed framework was verified using the Mininet and Floodlight controller. The proposed framework provides better results in terms of detection accuracy and false positive rate.

A DDoS attack defense scheme named FADM has been designed by Hu et al. in Hu et al. (2017). The authors primarily focus on the design of real-time DDoS detection and efficient DDoS mitigation. FADM utilizes the controller-based and sFlow-based network status collection methods to enhance the ability of rapid and accurate detecting DDoS attacks. Hence, it extracts five-tuple features from the network traffic and flow entries of the switches and trains the SVM classifier to detect the DDoS attack. In order to reduce the damage caused by the DDoS attack, a white-list method is introduced to stop the DDoS attack. For experimental evaluation design, the POX controller and Mininet were employed to draw a scene. The results show that FADM gives 100% detection accuracy once the attack rate is larger than 3000 packets per second, while the average response time is small.

In Santos et al. (2020), a mechanism used the SVM, MLP, Decision Tree, and Random forest as the detection algorithms has been designed by Santos et al. for detecting DDoS attack in SDN. It provides the analysis of the effectiveness of using the SVM, MLP, decision tree, and random forest on detecting DDoS attack in SDN. In Phan and Park (2019), Phan et al. proposed an SVM-based DDoS defense scheme. The detection of a DDoS attack depends on the combination of SVM and SOM models. The major advantage of this solution is that it can significantly decrease the false positive rate. In He et al. (2018), a multi-SDN-based cooperation mechanism for defending the DDoS attack in SDN is presented by He et al. It works on the idea that using the communication between SDN controllers to exchange information of DDoS attack detection, traceback, and mitigation. In this work, the SVM algorithm is utilized as the detection model. In Yang and Zhao (2018), Yang et al. proposed a mechanism to identify and defend the DDoS attacks in SDN. The DDoS detection of this mechanism comprises two steps: the IP entropy-based and SVM-based detection. The experimental results prove that the proposed mechanism can achieve a high recognition rate.

Latah et al. also designed an approach for defending against the DDoS attacks in SDN (Latah and Toker, 2018). The main idea is the combination of a threshold-based detection and SVM-based detection. In this work, the SVM-based detection, activated by the threshold-based detection results, is proposed to reduce the false positive rate. In Oo et al. (2017), a system is designed to detect the DDoS attacks in SDN. The main idea of this work is that it improves the SVM algorithm in terms of training time and testing time by introducing the AVL tree structure to sort the normalization of binary attributes. In Shang et al. (2017), Gao et al. designed a mechanism named FloodDefender to protect control plane and data plane resources against SDN-aimed DoS attacks. FloodDefender introduces three new technologies: packet filter used to identify and filter DDoS attack packets, flow rule management to monitor network traffic and block attack traffic, and table-miss engineering to save bandwidth of victim switches.

An SVM-based DDoS detection algorithm has been presented by Li et al. in Li et al. (2015). The essential idea of this algorithm is that it determines the punish parameters and kernel function parameters based on the generic algorithm and cross validation. Using SVM as the classifier, Kokila RT et al. designed a DDoS detection approach in Kokila et al. (2014). It preprocesses the source IP address, destination IP address, source port, destination port, protocol, and packet length and sends these features to the SVM model to identify DDoS attacks. In Liu et al. (2017a), a system named FL-Guard is presented for detecting and defending DDoS attacks in SDN by Liu et al. FL-Guard works on the idea of dynamically binding IP address, thus solving the IP spoofing problem in DDoS. Meanwhile, the SVM model is utilized to identify DDoS attacks in FL-Guard. In Chen et al. (2017), Chen et al. designed an amplification DDoS attack detection mechanism using the features of SDN. This mechanism adopts the network changes caused by the Distributed Reflection Denial of Service (DRDoS) attack including the count and packet size of forward and backward packets to build the SVM training model to detect DRDoS attack. In Mehr and Ramamurthy (2019), an SVM-based DDoS detection method has been designed for the RYU controller by Mehr et al. In Mowla et al. (2018), Mowla et al. presented a cognitive switch-based DDoS defense mechanism named CSDSM in SDN-driven Interconnect Content Delivery Network (CDNi). The detection sector used to monitor and extract network features, the analysis sector to identify DDoS attacks based on SVM, and the defense sector to mitigate DDoS attacks are three main components of CSDSM.

4.2.2. Decision tree-based DDoS detection mechanisms

In Santos et al. (2020), Santos et al. evaluated several machine learning algorithms, in terms of the detection accuracy and processing time. In their work, the Classification And Regression Tree (CART) based DDoS attack detection was proposed to identify the flow-table attack, bandwidth attack, and controller attack. They found that the CART-based DDoS attack detection method performs best, as it has the lowest processing time and good detection accuracy.

In Hyder and Lung (2018), Hyder and Lung presented another decision tree-based DDoS mitigation system. The Iterative Dichotomiser 3 (ID3) was introduced in the proposed system, which contains the training phase and online test phase. In the training phase, a prediction tree is constructed based on the training data set. In the online test phase, the usages of the controller resource such as the CPU, memory and bandwidth, and the packet count data are collected. Subsequently, the previously created prediction tree can distinguish the attacker IP addresses from the non-attacker IP addresses.

4.2.3. SMO-based DDoS detection mechanisms

Alshamrani et al. presented a DDoS defense mechanism based on the SMO algorithm (Alshamrani et al., 2017). The labeled NSL dataset is used for training and testing the SMO model. The proposed DDoS detection method is triggered by the *packet-in* messages. When receiving a trigger, the features of the received packets will be calculated and forwarded to the SMO model to distinguish whether there is a DDoS attack.

4.2.4. Softmax-based DDoS detection mechanisms

OverWatch, a DDoS attack defense framework used in SDN, was proposed by Han et al. in Han et al. (2018). In OverWatch, the combination of Autoencoder and Softmax classifier is utilized when designing the DDoS attack detection module. The designed model contains four layers: one input layer, two hidden layers, and one output layer, where

Table 5
Summary of classifying method-based DDoS detection mechanisms.

Reference	Year	Location	Method	Attack type	Experiments	Scale	Detection results
Phan and Park (2019)	2019	Controller	SOM, SVM	TCP/ICMP flood	Experiment	Laboratory LAN	DR:99.27, DA:99.3, FA:0.67
Phan et al. (2019)	2019	Application Server	SOM, SVM, Random forest	Low-rate DoS	Simulation	Small(1 switch)	–
Santos et al. (2020)	2018	Controller	MLP, CART, SVM, and Random forest	TCP/UDP/ICMP/HTTP flood	Simulation	Small(1 switch)	–
Rahman et al. (2019)	2019	Controller	SVM, Random Forest, J48, and KNN	TCP/ICMP flood	Simulation	Small(1 switch)	–
Bakker et al. (2018)	2018	Controller	SVM, Random Forest, and KNN	–	Experiment	Small(2 switches)	DR:14.44, DA:93.47, FA:0.2
Musumeci et al. (2020)	2020	Switch	SVM, Random Forest, J48, and KNN	TCP/ICMP flood	–	–	DA:98.5
Cui et al. (2019)	2019	Controller	SVM	Self-designed DDoS	Simulation	Small(1 switch)	–
Myint Oo et al. (2019)	2019	Application server	SVM	TCP/UDP flood	Simulation	Small(9 switches)	DR:97, DA:97, FA:2
Ye et al. (2018)	2018	Controller	SVM	TCP/UDP/ICMP flood	Simulation	Small(5 switches)	DA:95.24, FA:1.26
Yu et al. (2018)	2018	Controller	SVM	TCP/UDP/ICMP flood	Simulation	Small(7 switches)	DR:98.56, FA:0.32
Hu et al. (2017)	2017	Controller	SVM	TCP/UDP/ICMP flood	Simulation	Small(3 switches)	DR:100, FA:0
He et al. (2018)	2018	Controller	SVM	–	Simulation	Small(3 switches)	DA:98.71
Yang and Zhao (2018)	2018	Controller	SVM, threshold	DoS	–	–	DA:99.8
Latah and Toker (2018)	2018	Controller	SVM, threshold	–	Simulation	Small(4 switches)	DA:100, FA:0
Shang et al. (2017)	2017	Controller	SVM, threshold	UDP flood	Experiment and simulation	Small(5 switches)	DR:96, FA:5
Shen et al. (2020)	2020	Controller	SVM, threshold	–	Simulation	Small(7 switches)	FA:1.48
Oo et al. (2017)	2017	Application server	SVM	–	–	–	DR:95, FA:50
Li et al. (2015)	2015	–	SVM	–	–	–	TN:0.35, FN:0
Kokila et al. (2014)	2014	–	SVM	–	–	–	DA:95.11, FA:0.8
Liu et al. (2017a)	2017	Application server	SVM	–	Simulation	Small(3 switches)	DA:96.55
Chen et al. (2017)	2017	Application server	SVM	NTP amplification attack, DNS amplification attack	Experiment	Cloud computing	DA:99.99
Mehr and Ramamurthy (2019)	2019	Controller	SVM	–	Simulation	Small(3 switches)	–
Mowla et al. (2018)	2015	Controller	SVM	–	Simulation	Small(2 switches)	PRE:100, DR:92.9
Polat et al. (2020)	2020	Controller	SVM, NB, ANN, and KNN	TCP/UDP/ICMP flood	Simulation	Small(2 switches)	DA:98.3, PRE:97.72, F1:97.7
Hyder and Lung (2018)	2018	Controller	ID3, threshold	UDP reflection attack	Simulation	–	–
Alshamrani et al. (2017)	2017	Application server	SMO	–	Simulation	Small(4 switches)	DA:99.4
Han et al. (2018)	2018	Switch and controller	Softmax, threshold	Self-designed DDoS	Experiment	Small(1 switch)	DA:96

these two hidden layers are two autoencoders and the output layer is a softmax classifier. Each layer is trained using the back-propagation algorithm and then stacked together to form the attack classifier. Then it can be utilized to perform the DDoS attack classification.

4.2.5. Discussion of classifying-based DDoS detection mechanisms

A summary of classifying-based DDoS detection mechanisms is listed in Table 5. The common approach of employing classifying method as the classifier in DDoS detection contains two phases: training phase and testing phase. The training phase is responsible for finding the optimal classifying standard. For example, the SVM-based DDoS detection method needs to find the classification surface in the training phase. Taking advantages of the classifying standard determined by the training phase, the testing phase can classify the tested instance into a certain category, which can label the instance as a normal instance or DDoS attack.

The classifying-based DDoS detection technology is one of the most popular DDoS detection technology. As is shown in Table 5, 27 mechanisms designed based on the classifying method have been published. That may be caused by its high detection performance and simple modeling process. For example, compared with the neural network, the SVM-based DDoS detection mechanisms do not need to construct the network model, the user only needs to determine the kernel function and penalty coefficient, which is easier than constructing the neural network model.

There are also some limitations when employing the classifying algorithms to design a DDoS detection mechanism. Several classifying methods, especially the SVM and logistic regression, essentially belong to solutions of the binary classification problem. These classifying methods can perform well when detecting whether there is a DDoS attack in the network. But these methods maybe not suitable for further identifying the different kinds of DDoS attacks, such as the TCP flooding, UDP flooding, ICMP flooding attack.

4.3. Clustering-based DDoS detection mechanisms

The clustering-based DDoS detection mechanisms utilized in SDN are reviewed and analyzed in this section. When designing the clustering-based DDoS detection mechanisms, the clustering technologies are often considered as the identifier to distinguish the DDoS attack traffic from normal traffic. Currently, the clustering technologies introduced to detect DDoS in SDN contain the K-Nearest Neighbor (KNN) algorithms (Nam et al., 2018; Rahman et al., 2019; Bakker et al., 2018; Musumeci et al., 2020; Polat et al., 2020; Tan et al., 2020; Zhu et al., 2018; Sun et al., 2018), Dirichlet Process Mixture Model (DPMM) (Ahmed et al., 2017), Fuzzy C-Means (FCM) (Gao et al., 2018), Hierarchical Heavy Hitter (HHH) (Kalliola et al., 2015), and Whale Optimization Algorithm-based clustering (Shakil et al., 2019).

4.3.1. KNN-based DDoS detection mechanisms

As mentioned above, an algorithm combining the SOM and KNN is presented and used as the classifier in Nam et al. (2018). The evaluation results demonstrate that the combination of SOM and KNN can significantly reduce the processing time, with a little decrease in the detection rate and an increase in the false positive rate.

In Rahman et al. (2019), Rahman et al. presented their work about evaluating several machine learning-based DDoS detection approaches used in SDN. In this work, the effectiveness of the KNN-based DDoS detection approach has been evaluated in terms of recall, F score, sensitivity, accuracy, precision, training time, and testing time. The evaluation results indicate that KNN can achieve high accuracy, precision, recall, and F score, while it may cause a long testing time. Bakker et al. also carried out a comparison of the effectiveness of KNN, SVM and Random Forest based DDoS detection mechanisms in Bakker et al. (2018). The evaluation results also demonstrate that KNN can achieve high accuracy.

In Zhu et al. (2018), Predis, a privacy-preserving DDoS attack detection mechanism has been designed by Zhu et al. Predis is composed of three components: the computing server (CS), detection server (DS), and SDN domains. Each SDN domain collects the traffic data and calculates the median of packets per flow, median of bytes per flow, percentage of correlative flow, growth of ports, and growth of source IP addresses. Subsequently, it sends the encrypted features of the traffic data to the CS. Meanwhile, a relative perturbation parameter will be sent to the DS. Based on a kd-tree which is built by the training dataset, the CS computes the distance between the disturbance data and the training data and passes that distance to the DS. Considering that the DS has the perturbation parameter sent from the SDN domain and preliminary distance from the CS, it can get the correct distance. Then an improved KNN will be utilized to identify the DDoS attack.

Sun et al. presented a DDoS detection method in Sun et al. (2018). To enhance the changes caused by the DDoS attack, instead of utilizing the commonly used Shannon entropy to calculate the network features, this work introduces the φ entropy. Thus, the proposed DDoS detection mechanism can improve the detection accuracy. In particular, the average byte of flow, average duration of flow, φ entropy of source IP, φ entropy of destination IP, increasing speed of flow table are chosen as the flow characteristics. The KNN is utilized to detect a DDoS attack.

4.3.2. DPMM-based DDoS detection mechanisms

In order to mitigate the DNS amplification DDoS attack and DNS flooding DDoS attack, an SDN-based DPMM clustering method has been proposed in Ahmed et al. (2017). The traffic statistic manager, learner component, and network resources manager are the main components of that method. The traffic statistic manager is responsible for gathering the network features from the network devices. The learner component detects whether there is a DDoS attack in the network using DPMM. The network resources manager monitors the CPU and memory utilization of the network devices. Three features including the total number of packets, ratio of source and destination bytes, and connection duration time are combined as the representation of a connection. Based on these features and the learner component, the attack traffic will form a separate cluster, which can be used to identify the DDoS attack from legitimate traffic.

4.3.3. FCM-based DDoS detection mechanisms

In Gao et al. (2018), an FCM based DDoS mitigation method was proposed by Gao et al. The proposed method is composed of the switches classifier, feature extractor, anomaly detection, and attack mitigation. Taking advantages of the Bayesian network, the switches classifier can detect the switches that may be compromised to the attackers. Then the controller will capture the flow entries of these compromised switches. The feature extractor can calculate and form the mean of packets per flow, percentage of pair flows, growth of foreign flows, growth of different ports, deviation of packets count, and deviation of bytes count and structure these features into a six-tuple. Subsequently, that six-tuple will be clustered into a particular set by the FCM. Once a DDoS attack is detected, some actions will be taken to mitigate the DDoS attack by the attack mitigation module.

4.3.4. HHH-based DDoS detection mechanisms

Based on HHH, Kalliola et al. presented an automated DDoS defense mechanism in Kalliola et al. (2015). Using the automatic traffic learning and blacklists to detect and mitigate the DDoS attack is the essential principle of the proposed mechanism. The source IP addresses are utilized to build the normal traffic model. Thus, in the HHH model, the prefix tree is constructed based on a set of IP addresses. A predefined threshold is also used to split the traffic into several clusters. Hence, a cluster consists of a set of IP addresses and a hit count. The proposed mechanism can incrementally learn from the normal traffic to form the resulting clusters. Then it can be utilized to detect DDoS attacks.

4.3.5. WOA-based DDoS detection mechanisms

In Shakil et al. (2019), a WOA-based clustering for DDoS detection (WOA-DD) mechanism has been proposed. In WOA-DD, the request is cognized as an instance of the solution space. Thus, the core idea of WOA-DD is that it calculates the distance between the new request and normal or DDoS attack clusters centroid. Based on that distance, it assigns the new request to one of the clusters, which are grouped to normal or DDoS attacks using the historical data. The fitness of the whales is defined as the minimum average distance between the request and the cluster centroid. The mathematical analysis and performance analysis indicate that although WOA-DD leads to a noticeable delay when processing the new request, it can efficiently detect the DDoS attacks.

4.3.6. Discussion of clustering-based DDoS detection mechanisms

Table 6 lists the summary of clustering-based DDoS detection mechanisms. A common approach of using the clustering method to detect DDoS consists of the training step and testing step. As in Gao et al. (2018), Shakil et al. (2019), and Zhu et al. (2018), the training step clusters the instances into different clusters and finds out the center of each cluster. When identifying DDoS, these mechanisms calculate the distances between the test instance and these centers. Based on the above distances, the test instances will be judged as normal or DDoS.

The clustering method-based DDoS detection mechanisms can achieve high detection rate and low detection time. For instance, in Zhu et al. (2018), the precision is up to more than 99% while the detection time is about 50 ms. Except the excellent detection performance, as the clustering method can discover the unknown cluster, the clustering method-based DDoS detection mechanisms have the ability to detect a new type of DDoS.

There are also some limitations in the clustering method-based DDoS detection mechanisms. First, some clustering methods are sensitive to the dataset, especially for the noises and outliers. Hence, when using these clustering methods, the data must be carefully processed. Second, the performance of some clustering methods extremely depends on their parameters. For instance, the initial clustering center of FCM and the number of clusters of KNN.

Table 6
Summary of clustering-based DDoS detection mechanisms.

Reference	Year	Location	Method	Attack type	Experiments	Scale	Detection results
Nam et al. (2018)	2018	Controller	SOM, KNN	–	Experiment	Small(1 switch)	DR:98.24, FA:2.14
Rahman et al. (2019)	2019	Controller	SVM, Random Forest, J48, and KNN	TCP/ICMP flood	Simulation	Small(1 switch)	–
Bakker et al. (2018)	2018	Controller	SVM, Random Forest, and KNN	–	Experiment	Small(2 switches)	DR:14.44, DA:93.47, FA:0.2
Musumeci et al. (2020)	2020	Switch	SVM, Random Forest, J48, and KNN	TCP/ICMP flood	–	–	DA:98.5
Bakker et al. (2018)	2018	Controller	SVM, Random Forest, and KNN	–	Experiment	Small(2 switches)	DR:14.44, DA:93.47, FA:0.2
Polat et al. (2020)	2020	Controller	SVM, NB, ANN, and KNN	TCP/UDP/ICMP flood	Simulation	Small(2 switches)	DA:98.3, PRE:97.72, F1:97.7
Zhu et al. (2018)	2018	Application server	KNN	TCP flood, LLDOS	Simulation	–	PRE:99.2, DR:99
Sun et al. (2018)	2018	Application server	KNN	TCP/UDP/ICMP flood	–	Small(4 switches)	DR:99.8, FA:1, F1:98.7
Ahmed et al. (2017)	2017	Controller	DPMM	–	–	–	–
Gao et al. (2018)	2018	Controller	FCM	56 types of attack	Simulation	Data center	TPR:95.68, FA:10
Kalliola et al. (2015)	2015	Application server	HHH	Naive flood, DNS reflection, botnet, valid requests, randomly spoofed	Experiment	Small(3 switches)	–
Shakil et al. (2019)	2018	–	WOA	Self-designed DDoS	Simulation	–	PRE:93.97, DR:94.59, F1:87.17, DA:82.83

4.4. Deep learning-based DDoS detection mechanisms

Deep learning, also known as hierarchical learning, is a class of machine learning algorithms which imitates the understanding and learning ability of human brains. Due to its ability of modeling high-level abstractions of the structured and unstructured data, deep learning has already been used in speech recognition, image recognition, natural language processing, and other fields. Meanwhile, deep learning models including the Convolutional Neural Network (CNN) (Arivudainambi et al., 2019; Wang et al., 2020b; Haider et al., 2020; Li et al., 2018; Narayanadoss et al., 2019), Recurrent Neural Network (RNN) (Li et al., 2018; SaiSindhuTheja and Shyam, 2021), Long Short-Term Memory (LSTM) (Li et al., 2018; Narayanadoss et al., 2019), and self-designed deep learning models (Tang et al., 2016; Asad et al., 2019) have also been utilized to detect DDoS attacks in SDN.

4.4.1. CNN-based DDoS detection mechanisms

In Arivudainambi et al. (2019), by combining the Lion Optimization Algorithm (LOA) and CNN, Arivudainambi et al. proposed a DDoS detection method named LION-IDS. LOA is a meta-heuristic algorithm that can find the optimal solution. Hence, in order to enhance the detection accuracy of the CNN model, LOA is introduced to select optimal feature subsets used to train the CNN model. More exactly, the position of each lion is considered as a distinct vector of coordinates. The lions can update their positions based on their fitness. Here, the fitness is defined as the mean detection accuracy of the CNN classifier. After searching, the best position of each lion is recorded as the feature subsets which will be utilized to train the CNN model. The trained CNN model can classify the DDoS attacks. The evaluation results show that the proposed LION-IDS performs better than the state-of-the-art methods in terms of detection accuracy and false positive rate.

Wang et al. designed a CNN-based DDoS detection mechanism in Wang et al. (2020b). The proposed detection mechanism works in a general framework for software-defined Internet of Things. A CNN model consisted of the input layer, convolutional layer, pooling layer, fully connected layer, and output layer is built. The simulation was performed on the Floodlight controller and OpenvSwitch. According to simulation results, the proposed model performs better than KNN, SVM, and DNN in terms of accuracy, precision, recall, and F1 score.

4.4.2. Hybrid deep learning: CNN & RNN & LSTM-based DDoS detection mechanisms

A hybrid deep learning model was constructed to detect DDoS attacks by Li et al. in Li et al. (2018). The proposed DDoS detection model consists of the input layer, forward recurrent layer, backward recurrent layer, fully connected hidden layer, and prediction output layer. Twenty features including the text features, numerical features, and boolean features are extracted and passed to the input layer of the hybrid deep learning model. In that layer, the batch standardization is firstly executed. Meanwhile, the stack convolutional layer is also used in the input layer. Then the output data of the input layer are simultaneously sent to the forward recurrent layer and backward recurrent layer. In order to solve the gradient disappearance problem of the forward recurrent layer and backward recurrent layer, LSTM and GRU are introduced. Subsequently, the output of these two layers is input to the fully connected hidden layer. At last, the data processed by the fully connected hidden layer are transmitted to the prediction output layer to predict the output result. Based on that result, the hybrid deep learning model can detect whether the packets are DDoS attack packets.

4.4.3. CNN & LSTM-based DDoS detection mechanisms

In Narayanadoss et al. (2019), a deep learning-based DDoS detection mechanism has been proposed to detect the crossfire attack. The designed DDoS detection mechanism is running on the controller. Two kinds of methods are considered when capturing the network status. The first one is periodically obtaining the network traffic by the SDN controller, while the second one is a event-driven approach, where the switch can ask the controller to perform traffic monitoring when the switch has high packet loss. The number of flows, aggregate flow size, and timestamp are the features extracted from the network traffic. Subsequently, a CNN and an LSTM model are constructed. In the CNN model, the input data is a 2-dimensional array, where each row is the traffic feature at a specific period. The CNN model contains two separate convolutional steps. The first one processes one row at a time while the second one simultaneously processes a few rows. A fully-connected layer is also designed to receive the output of these two convolutional steps and generate the final binary value to determine whether there is a DDoS attack. Except the above CNN model, an LSTM model that consists of two consecutive LSTM cells and a fully-connected layer is also presented. The evaluation results indicate that the LSTM model outperforms CNN on the detection accuracy.

Table 7
Summary of deep learning-based DDoS detection mechanisms.

Reference	Year	Location	Method	Attack type	Experiments	Scale	Detection results
Arivudainambi et al. (2019)	2018	–	LOA, CNN	TCP/UDP flood	Simulation	Small(6 switches)	DA:99.7, FA:0.57
Wang et al. (2020b)	2020	Controller	CNN	–	Simulation	Small(6 switches)	–
Haider et al. (2020)	2020	Controller	CNN	–	–	–	DA:99.45, DR:99.12, PRE:99.57, F1:99.61
Li et al. (2018)	2018	Application server	CNN, RNN, LSTM	–	Simulation	–	DA:99.79
Narayanadoss et al. (2019)	2019	–	CNN, LSTM	Crossfire attack	Simulation	Intelligent transport system	DA:91
SaiSindhuTheja and Shyam (2021)	2021	–	RNN	–	Simulation	–	PRE:98.18, DA:94.12, DR:95.13, F1:93.56
Tang et al. (2016)	2016	Controller	Self-designed DNN	back, land, neptune, pod, smurf, teardrop	Experiment	Small(4 switches)	DA:75.75, PRE:83, DR:75, F1:74
Asad et al. (2019)	2019	–	Self-designed DNN	Slowloris, SlowHTTPTest, Hulk and GoldenEye	–	–	F1:99

4.4.4. Self designed DNN-based DDoS detection mechanisms

In Tang et al. (2016), based on a self-designed deep learning model, a network intrusion detection mechanism was presented. The DNN model consists of an input layer, three hidden layers, and an output layer. There are 6 neurons in the input layer and 2 neurons in the output layer. Meanwhile, the number of neurons in the three hidden layers are 12, 6, and 3, respectively. Six features including the duration, protocol type, number of bytes from source to destination, number of bytes from destination to source, number of connections of the same host as the current connections in the past two seconds, and the number of connections of the same service as the current connection in the past two seconds are extracted and used to train the model. Then the trained model can detect the network intrusion. In this work, the DDoS attack is recognized as a typical type of network intrusion and utilized to evaluate the proposed network intrusion detection mechanism.

Asad et al. proposed a detection method named DeepDetect for DDoS attacks in SDN (Asad et al., 2019). A self-designed feed-forward back-propagation neural network is constructed in this work. It contains one input layer with 66 neurons, seven hidden layers which have 128, 256, 128, 64, 32, 16, and 8 neurons, and one output layer with 5 neurons. Meanwhile, the ReLu activation function is employed in the hidden layers, while the Softmax is applied for computing the loss of the output layer. The effectiveness of DeepDetect is verified on the Canadian dataset (CIC IDS 2017). As per the results, DeepDetect provides efficient detection accuracy with the F1 score at about 0.99.

4.4.5. Discussion of deep learning-based DDoS detection mechanisms

A summary of deep learning-based DDoS detection mechanisms is listed in Table 7. The number of deep learning-based DDoS detection mechanisms is less than the neural network-based, classifying-based, and clustering-based mechanisms. That may be caused by the fact that when detecting the high-rate DDoS, the neural network-based, classifying-based, and clustering-based DDoS detection can achieve high detection precision and accuracy with less computation complexity.

In contrast, for the low-rate DDoS attack and new type of DDoS such as Crossfire that have the nature of concealment, deep learning-based DDoS detection mechanisms may perform better than the neural network-based, classifying-based, and clustering-based DDoS detection mechanisms. The reason is that the deep learning-based DDoS detection mechanism has stronger data mining capability than these technologies, thus it can achieve better detection accuracy when detecting the covert DDoS attack.

When using deep learning to detect DDoS, some obstacles or limitations should be figured out. The first one is the training time of the deep learning model may be quite long. In Arivudainambi et al. (2019), the training process lasts for more than 10 h. Hence, the long training time may make the deep learning module has high requirements on hardware. The second obstacle is that many parameters should be determined when constructing a deep learning module. Take CNN as an example, the user must choose the optimal or suboptimal parameters including the number of layers, number of neurons, number of filters, number of epoch, learning rate, objective function, weight initialization, and regularization. Although deep learning-based DDoS detection mechanism has such limitations, we still recommend using of deep learning to design the DDoS detection mechanism, especially for the low-rate DDoS and Crossfire attack.

4.5. Ensemble learning-based DDoS detection mechanisms

The ensemble learning-based DDoS detection mechanisms are reviewed and analyzed in this section. Specifically, the bagging-based (Phan et al., 2019; Santos et al., 2020; Rahman et al., 2019; Bakker et al., 2018; Musumeci et al., 2020) and boosting-based DDoS detection mechanisms (Chen et al., 2018a) used in SDN are described.

4.5.1. Bagging-based DDoS detection mechanisms

In Santos et al. (2020), Santos et al. proposed a DDoS detection mechanism used in SDN. Four machine learning algorithms including the SVM, MLP, Decision Tree, and Random Forest have been utilized for detecting DDoS attacks. For the experiment, the author employed the Mininet and POX as the network simulator and SDN controller. According to the experiment results, the Random Forest achieves the best accuracy for detecting the controller-target DDoS attack, flow table-target DDoS attack, and bandwidth controller-switch-target DDoS attack, as compared to the SVM, MLP, and Decision Tree. However, the results also indicate that Random Forest also takes the most time for detecting DDoS attack, which may be a weakness of the Random Forest-based DDoS detection mechanisms.

Rahman et al. evaluated several machine learning-based DDoS detection mechanisms in Rahman et al. (2019). In their work, the Random Forest-based DDoS detection mechanism has been tested for detecting TCP flood attack and ICMP flood attack. According to the evaluation results, Random Forest-based DDoS detection mechanism can achieve high precision, accuracy, recall, and F score, while it causes long testing time than SVM and KNN.

In Bakker et al. (2018), Bakker et al. also compared the Random Forest-based DDoS detection mechanism with the SVM and KNN based mechanisms. In their evaluation, the Random Forest-based DDoS detection mechanism achieved less accuracy. However, the Random Forest-based DDoS detection mechanism performed better than SVM and KNN in terms of classifier initialization time and packet processing time.

Table 8
Summary of ensemble learning-based DDoS detection mechanisms.

Reference	Year	Location	Method	Attack type	Experiments	Scale	Detection results
Phan et al. (2019)	2019	Application Server	SOM, SVM, Random forest	Low-rate DoS	Simulation	Small(1 switch)	–
Santos et al. (2020)	2018	Controller	MLP, CART, SVM, and Random forest	TCP/UDP/ICMP/HTTP flood	Simulation	Small(1 switch)	–
Rahman et al. (2019)	2019	Controller	SVM, Random Forest, J48, and KNN	TCP/ICMP flood	Simulation	Small(1 switch)	–
Bakker et al. (2018)	2018	Controller	SVM, Random Forest, and KNN	–	Experiment	Small(2 switches)	DR:14.44, DA:93.47, FA:0.2
Musumeci et al. (2020)	2020	Switch	SVM, Random Forest, J48, and KNN	TCP/ICMP flood	–	–	DA:98.5
Chen et al. (2018a)	2018	Controller	Boosting	–	–	–	DA:98.53, FA:0.8

4.5.2. Boosting-based DDoS detection mechanisms

Except the bagging-based DDoS detection mechanisms, the boosting-based DDoS detection mechanisms have also been already utilized for detecting DDoS in SDN. Chen et al. proposed an eXtreme Gradient Boosting (XGBoost) based DDoS attack detection mechanism in Chen et al. (2018a). By iteratively splitting nodes, XGBoost can construct a specified number of trees. Subsequently, it can identify whether the features belong to DDoS attacks. Four types of features including the TCP connection basic features, TCP connective content features, time-based network traffic statistical characteristics, and host-based network traffic statistical characteristics are extracted from the network. These features form a training data set used to train the XGBoost tree. Then the trained XGBoost tree can online test the network features to identify the DDoS attack. As per the result, XGBoost outperforms GBDT, Random forest and SVM, in terms of detection accuracy, false positive rate.

4.5.3. Discussion of ensemble learning-based DDoS detection mechanisms

Table 8 lists the ensemble learning-based DDoS detection mechanisms. The bagging and boosting methods have been used to detect DDoS attacks in SDN. The bagging-based DDoS detection methods need to sample instances from the training data set and train base learners. By iteratively adjusting the weight of instances in the training data set, the boosting-based DDoS detection methods can focus on the instances that hard to classify.

The ensemble learning-based DDoS detection mechanism performs well in terms of detection accuracy. According to the research of Santos et al. (2020), the Random Forest-based DDoS detection mechanism outperforms the SVM, MLP, and decision tree-based DDoS detection mechanisms, in terms of true positive rate and false positive rate. The possible reason for the higher detection accuracy of the ensemble learning-based DDoS detection method is that it employs a number of base classifiers to execute the DDoS detection and makes the final decision based on these separate detection results.

Although the ensemble learning-based DDoS detection mechanisms can achieve higher detection accuracy, it may have longer processing time. For instance, in Santos et al. (2020), the processing time of the random forest-based DDoS detection mechanism is about 22 s, which is quite longer than that of SVM, MLP, and decision tree.

In conclusion, the ensemble learning-based DDoS detection mechanisms can achieve excellent detection accuracy while cause longer processing time. Subsequently, considering that for the high-rate DDoS attack, many machine learning-based DDoS detection mechanisms (SVM, MLP, BPNN, and so on) can achieve high detection accuracy with short processing time, it is better to use the above machine learning-based DDoS detection mechanisms, instead of the ensemble learning-based DDoS detection mechanisms. However, it is hard to detect the low-rate DDoS because of the stealthiness of the low-rate DDoS. Therefore, for the low-rate DDoS, the detection accuracy must be considered as the most crucial factor when choosing a DDoS detection mechanism. Hence, the ensemble learning-based DDoS detection mechanism may be a good choice for defending the low-rate DDoS attack.

5. Statistics-based DDoS detection mechanisms

In this section, we focus on the description and analysis of the statistics-based DDoS detection mechanisms. The statistics methods employed in DDoS attack detection in SDN include the Cumulative Sum (CUSUM) (Wang et al., 2016a; Conti et al., 2017; Mahrach et al., 2018), Credit-Based Threshold Random Walk (TRW-CB) and Rate Limiting (RL) (Birkinshaw et al., 2019; Özçelik et al., 2017), Discrete Wavelet Transform (DWT) (Zerbini et al., 2019), Sequential Probability Ratio Test (SPRT) (Dong et al., 2016), Markov model (Wang et al., 2018a), and self-designed method (Kalkan et al., 2017).

5.1. CUSUM-based DDoS detection mechanisms

Wang et al. designed a mechanism for defending DDoS attacks in SDN based on legitimate source and destination IP address (Wang et al., 2016a). The objective of this mechanism is to segregate the DDoS attack traffic from normal traffic. It works on two methods, the legitimate source–destination IP addresses database, and the CUSUM-based DDoS attack detection method. Taking advantages of introducing the mean value of the input data and a constant value decided by the network, the authors improved the traditional CUSUM algorithm. The evaluation was performed on POX. The effectiveness of that mechanism is measured in terms of detection accuracy and network parameters. According to the simulation results, the proposed mechanism can achieve 100% detection accuracy while decreasing the RTT and packet loss of network traffic.

In Conti et al. (2017), Conti et al. proposed a comprehensive detection method against the DDoS attack. The main idea of the proposed method is that it detects the DDoS attacks based on an improved CUSUM algorithms. An adaptive threshold is introduced to the CUSUM algorithm. For the experiment design, a POX controller and three OpenFlow switches were utilized in the evaluation. The detection rate, false alarm rate, detection accuracy, detection time, and overhead of the controller were considered as main parameters used to evaluate the proposed method. The evaluation outcomes prove the effectiveness of the proposed method in terms of the detection rate and false alarm rate.

In Mahrach et al. (2018), Mahrach et al. also designed a mechanism to detect the SYN flooding attacks in SDN. The proposed mechanism is composed of three modules: (1) traffic information DBs, (2) traffic anomaly detection, and (3) inspection and mitigation. Employing P4-enabled switches, the number of TCP SYN packets received by each host or server are recorded to constructed the traffic information DBs. For detecting the unexpected DDoS attacks, CUSUM is employed as the detection algorithm. The inspection and mitigation module is responsible for ensuring the existence of DDoS attacks and mitigating these attacks.

5.2. CB-TRW and RL-based DDoS detection mechanisms

In Birkinshaw et al. (2019), Birkinshaw et al. proposed an intrusion detection and prevention system against DoS attack and port scanning attack. The CB-TRW-based method and RL-based method are designed to identify the port scanning attack and DDoS attack. It first extracts the packet encapsulated in the *packet-in* messages received by the controller. Then the number of SYN packets, UDP packets, and ICMP packets are counted and compared with the related threshold. Once the number exceeds the threshold, it decides that a DDoS occurs. The experimentation was performed on the POX controller and OpenvSwitch. The results show that the proposed mechanism can efficiently detect DoS attacks.

In Özçelik et al. (2017), an software defined edge defense mechanism named ECESID was proposed to detect and mitigate IoT-based DDoS. ECESID works on the integration of CB-TRW and RL algorithms. It uses CB-TRW to calculate the likelihood ratio of a host being a malicious attacker by checking the TCP RST packets of that host. Meanwhile, the new connection requests are handled using the RL algorithm. It defines a list of recently connected hosts as the 'working set'. If the number of request that connects to the host not in the working set exceeds a threshold value, a DDoS attack is detected. The evaluation was performed on the Mininet-WiFi. The time spent to identify a DDoS attack and the throughput of a selected benign host are measured to evaluate the performance of ECESID. The outcomes show that ECESID spends an average of 6.02 s to detect the DDoS attack, while recovering the maximum throughput after mitigating the DDoS attack.

5.3. DWT-based DDoS detection mechanisms

Zerbini et al. proposed a method for detecting the DDoS attacks in SDN (Zerbini et al., 2019). It attempts to analyze the network status based on the DWT to detect DDoS. The number of bits and packets and Shannon entropy of source/destination IP/ports are gathered and decomposed into the detail and approximation coefficients by the Haar wavelet function with two decomposition levels. Then the mean μ and standard deviation ν of the detail and approximation coefficients in two time windows: W_f and W_d are calculated. At last, the 3-σ rule is employed to check whether the μ and ν are abnormal to identify the DDoS attacks. Using Mininet, a tree-like network contains 5 switches, and 80 hosts was created as the experiment scenarios. The proposed mechanism was implemented on the POX controller. Main metrics, including the precision, accuracy, false-positive rate, false-negative rate, F-measure, and recall are utilized to evaluate the proposed mechanism. According to the evaluation results, the proposed mechanism is shown to achieve better accuracy, recall, F-measure, and AUC.

5.4. SPRT-based DDoS detection mechanisms

In Dong et al. (2016), a detection mechanism against the novel attack aimed at the SDN controller was designed by Dong et al. An SPRT-based detection method was proposed to detect DDoS attack. Two boundaries, i.e., A and B are defined in the detection method. The log-probability ratio of an interface i being compromised is calculated. Then the detection result is decided by that ratio and the above boundaries. The evaluation was performed on the DARPA data set. According to the results, the proposed mechanism performs better than the detection methods based on percentage, count, and entropy in terms of promptness, versatility, and detection accuracy.

5.5. Markov model-based DDoS detection mechanisms

Wang et al. designed an approach against the low-rate DDoS attacks in SDN (Wang et al., 2018a). It focuses on the work of using the hidden Markov model to detect DDoS attacks. The Renyi entropies of the destination and source IP addresses of the network traffic are chosen as features adopted in detection. The Baum–Welch algorithm is employed to train the Markov model. After that, the Viterbi algorithm is used to execute the Markov decoding. The evaluation was performed on the Mininet and POX. The performance of this approach is compared with KNN, SVM, SOM, and BPNN, in terms of the true positive rate, false positive rate, and false negative rate. As per the results, the proposed approach provides better results in the true positive rate and false positive rate.

5.6. Self-designed statistics method-based DDoS detection mechanisms

In Kalkan et al. (2017), Kalkan et al. designed a statistical defense mechanism named SDNScore against the DDoS attacks in SDN. The main idea of SDNScore is that it allocates each packet a score calculated using a self-designed equation. More specifically, SDNScore is compromised of four modules: profiler, actuator, comparator, and scorer. A normal profile of each switch is first generated by the profiler. The bandwidth usage of the network is monitored by the actuator. If a surge is located, the comparator selects two attributes that are the most different from the normal profiles, from eight attributes: source IP, destination IP, source port, destination port, protocol type, packet size, TTL value, and TCP flag. Then the scorer calculates the score of each packet using the ratio of number of packets in the current profile and normal profile that have the same selected attributes. The score is compared with a dynamic threshold calculated by the load shedding algorithm. If that score is bigger than the threshold, the packet will be dropped. SDNScore was evaluated in terms of detection accuracy, communication overhead, and storage overhead. The simulation results show that SDNScore provides excellent detection accuracy, especially for unknown DDoS attacks.

5.7. Discussion of statistics-based DDoS detection mechanisms

A summary of the statistics-based DDoS detection mechanisms is listed in Table 9. Generally, the statistics-based DDoS detection mechanism employs a statistics method to calculate a measure value based on the features of the network and users. Then the measure value is compared with a reference value to identify whether there is a DDoS attack. For instance, in Wang et al. (2016a), Conti et al. (2017), Mahrach et al. (2018), the CUSUM coefficient is first calculated by CUSUM and then compared with a threshold to detect a DDoS attack. In Zerbini et al. (2019), the approximation coefficient is computed and compared with two boundaries to classify an abnormal point.

The statistics-based DDoS detection mechanism has the advantage that it only needs to calculate the measure value using some simple equations, instead of building a complex model. Hence, the computational complexity of the statistics-based DDoS detection mechanism is low, which makes it quite suitable for applying in the high-rate DDoS attack detection.

Nevertheless, the statistics-based DDoS detection mechanism usually compares the measure value with a simple threshold to identify DDoS attacks. Whether the threshold is appropriate significantly affects the detection performance. Therefore, the dynamic threshold is appropriate for the statistics-based DDoS detection mechanisms. For instance, by dynamically adjusting the threshold using 3σ rule, the detection accuracy, precision, recall, and F-score of the statistics-based DDoS detection mechanism can be improved.

Table 9
Summary of statistics-based DDoS detection mechanisms.

Reference	Year	Location	Method	Attack type	Experiments	Scale	Detection results
Wang et al. (2016a)	2016	Server	CUSUM	–	–	–	–
Conti et al. (2017)	2017	Controller	CUSUM	Smurf, Neptune, IPsweep, Portscan	Simulation	Small(3 switches)	DR:100, FA:11.63, DA:94.23
Mahrach et al. (2018)	2018	Controller and switch	CUSUM	TCP flood	–	–	–
Birkinshaw et al. (2019)	2019	Controller	CB-TRW and RL	TCP flood	Experiment	Small(1 switch)	–
Özçelik et al. (2017)	2017	Controller	CB-TRW and RL	TCP flood	Simulation	IoT network	–
Zerbini et al. (2019)	2019	Server	DWT	TCP flood	Simulation	Small(5 switches)	DA:98, PRE:82, F1:89, AUC:98, FA:1.44
Dong et al. (2016)	2016	Controller	SPRT	56 types of attacks	–	–	–
Wang et al. (2018a)	2018	Controller	Markov model	TCP flood	Simulation	Data center	DA:94.61, FA:1, FNR:8
Kalkan et al. (2017)	2017	Controller and switch	Self-designed statistics method	TCP flood, DNS amplification, NTP attack	Simulation	Small(2 switches)	PRE:99, DR:100, DA:99, F1:99

6. Combination of multiple methods-based DDoS detection mechanisms

Except introducing only one algorithm to detect DDoS attack, the works on how to integrate multiple methods to detect DDoS attack are also considered and executed. Currently, the threshold-SVM (Yang and Zhao, 2018; Latah and Toker, 2018; Shang et al., 2017; Shen et al., 2020), threshold-fuzzy logic (Dang-Van and Truong-Thu, 2017), threshold-BPNN-PSO (Liu et al., 2019), threshold-RBF-PSO (Dayal and Srivastava, 2018), SVM-SOM (Phan and Park, 2019; Phan et al., 2016), entropy-C4.5 (Sudar and Deepalakshmi, 2020), Kmeans-KNN (Tan et al., 2020), and SOM-kNN (Nam et al., 2018) based DDoS detection mechanisms have been proposed.

6.1. Threshold-SVM-based DDoS detection mechanisms

Yang et al. introduced a threshold and SVM-based detection mechanism against the DDoS attack in SDN (Yang and Zhao, 2018). In this mechanism, the IP entropy is calculated and compared with a threshold. If a DDoS attack is detected, the SVM based detection will start to further identify the DDoS attack. The proposed mechanism was evaluated using the KDD99 dataset. According to the evaluation results, it can detect the DDoS attacks with a high rate.

In Latah and Toker (2018), Latah et al. also proposed a multi-phases-based detection mechanism for identifying the DoS flooding attacks in SDN. Similarly, the basic idea is roughly detecting DoS attacks based on the threshold method and precisely detecting DoS attacks using SVM. It collects the flow statistics every 10 s and calculates the related packet rate. If that rate is greater than 100 packets per second, the system will initiate the second stage to accurately detect the DoS attacks. In the second stage, the flow statistics will be gathered every 4 s and 9 features will be calculated and passed to an SVM classifier to identify DoS attacks. The effectiveness of the designed mechanism is tested in the form of the detection accuracy and false alarm rate. The results show that it can achieve an average detection accuracy at 96.25% with an average false alarm rate at 0.26%.

Shang et al. designed a mechanism named FloodDefender against the DoS attacks in SDN (Shang et al., 2017). FloodDefender introduces two filters to identify the DoS attacks, including a rough filter and a precise filter. The authors assume that the frequency of DoS attack flows is low. Consequently, in the first filter, the flow with a frequency less than a threshold is recorded as an attack traffic. In the second phase, the packet count, byte count, asymmetric packet count, and asymmetric byte count of flow marked as an attack in the first phase will be extracted and sent to a trained SVM model to further detect whether it belongs to a DoS attack flow. The experiment was performed using the RYU controller and commercial OpenFlow switches, Polaris xSwitch X10-24S2Q. The effects of FloodDefender are compared with FloodGuard. According to the results, FloodDefender provides better performances, in terms of the flow table utilization, time delay, and packet loss rate.

6.2. Threshold-Fuzzy logic-based DDoS detection mechanisms

Tuyen et al. presented a multi-criteria-based mechanism against the DDoS attack in SDN in Dang-Van and Truong-Thu (2017). The main idea of the proposed mechanism is that it introduces three states of the current network: normal state, suspected to be attacked state, and attack mitigating state. In the normal state, a threshold-based DDoS detection approach is used to decide whether the system turns into the suspected to be attacked state or the attack mitigating state by comparing the number of new flows with the pre-defined thresholds. In the suspected to be attacked state, the fuzzy logic works to further detect the DDoS attack through selecting inputs, fuzzifying selected inputs, evaluating rules, and defuzzifying outputs. At last, in the attack mitigating state, the DDoS attack is mitigated by dropping DDoS attack flows. The evaluation was performed on the Netnam traffic dataset, where the proposed mechanism is proved to achieve a high detection rate with a moderately low false positive rate.

6.3. Threshold-BPNN-PSO-based DDoS detection mechanisms

Liu et al. presented a DDoS detection mechanism in Liu et al. (2019). The main idea of this mechanism is to pre-detect DDoS attack by applying threshold-based detection on switches and precisely detect DDoS attack by executing BPNN-PSO-based detection on the controller. Therefore, at each switch, the entropy of destination IP address is calculated and compared with a threshold. Once it signals a DDoS attack alert, the anomaly detection module deployed on the controller starts to further detect the DDoS attacks. In this step, it utilizes the PSO algorithm to calculate the optimal initial weight and threshold of the BPNN module. For the experiment, the Floodlight and Mininet were employed as the controller and network simulator. D-ITG, a packet generator, was used to generate the benign traffic, while the Hping3 was utilized to launch DDoS attacks. According to the experiment results, the designed mechanism shows better performance as compared to BPNN and SVM, in terms of detection rate (97.47%), false alarm rate (1.43%), and accurate rate (98.02%).

6.4. Threshold-RBF-PSO-based DDoS detection mechanisms

In Dayal and Srivastava (2018), a mechanism used for early detecting the DDoS attacks in SDN has been designed by Dayal et al. The proposed mechanism works on the idea of combining the threshold-based detection and RBF-PSO-based detection. The flow statistics are

gathered periodically. Then the entropy of destination IP address is calculated and compared with a base threshold, which is set as 0.5 in this mechanism. Once the calculated entropy is less than 0.5, the RBF-PSO-based DDoS detection module will be launched to identify the DDoS attacks. The effectiveness of this mechanism was tested in the off-line and real-time experiment. The evaluation results demonstrate that the proposed mechanism can achieve high detection accuracy for TCP flooding attack, UDP flooding attack, and ICMP flooding attack.

6.5. SVM-SOM-based DDoS detection mechanisms

In Phan and Park (2019), Phan et al. designed a DDoS attack defense mechanism in the SDN-based cloud. By combining the SVM and SOM, it aims to decrease the false alarm rate of DDoS detection. The SVM classifier is first used to identify a DDoS attack by checking the flow duration and packet number. If the SVM classifier cannot determine whether the flow belongs to an attack or normal pattern, the SOM classifier starts to further analyze the input vector by inspecting its flow duration, packet number, byte number, and protocol. At last, both detection results are gathered by a scheme named eHIPF to make the final decision. Only when both detection results indicate the existence of a DDoS attack, eHIPF will report a DDoS alarm. Subsequently, eHIPF starts to identify the attack source. A score X_i of the tth observation period of the protocol i is calculated based on the active time, number of flows, and average number of packets per flow. Similarly, the score X_j of the $t+1$th observation period of the source j is calculated. If X_j is bigger than X_i, the source j is determined as an attack source. For the experiments, the BoNeSi DDoS attack tool was used to demonstrate the proposed DDoS detection approach. The detection rate, accuracy and false alarm, data plane resource consumption, and control plane resource consumption are considered as the main performance indexes of the proposed approach and compared approaches. The experiment results validate that the proposed approach performs best in all testing.

Phan et al. also proposed a DDoS mitigation mechanism in Phan et al. (2016). Similar to the one proposed in Phan and Park (2019), the SVM classifier and SOM classifier work together to detect DDoS attacks. However, instead of making the final decision based on the above eHIPF, an attack classifier is introduced to classify the DDoS attacks into bandwidth depletion attacks and resource depletion attacks based on the protocol. According to the experiment results, the proposed mechanism shows better performance as compared to SVM and SOM-based DDoS detection methods, in terms of the detection accuracy and controller overload.

6.6. SOM-KNN-based DDoS detection mechanisms

In Nam et al. (2018), a DDoS attack detection mechanism designed based on the SOM and KNN was proposed. The primary aim of the combination is to reduce the computation complexity of finding k nearest neighbors. It first uses the SOM to train all the feature vectors to find respective Best Matching Units (BMUs). When detecting DDoS, instead of calculating the distances between the input instance and all training instances, it only needs to calculate the distances from the input instance to these BMUs to find k nearest neighbors. Subsequently, the most frequent label of the these k nearest neighbors will be found out and assigned to the input instance.

6.7. Entropy-C4.5-based DDoS detection mechanisms

Sudar et al. presented a two level DDoS detection approach using the combination of the entropy and C4.5 algorithm (Sudar and Deepalakshmi, 2020). It consists of a suspicious detection module and a C4.5 classification module. The suspicious detection module compares the entropy of the source IP addresses with a threshold to roughly detect DDoS attacks. In the C4.5 classification module, the average number of packets per flow, average duration per flow, average number of bytes per flow, pair flows percentage, growth of single-flows, and growth of different ports are extracted and sent to the C4.5 model to further analysis the network traffic and detect the DDoS attack. According to simulation results, the proposed approach provides higher accuracy as compared to the random forest and Gradient Boost Decision Tree (GBDT) based DDoS detection mechanism.

6.8. KMeans-KNN-based DDoS detection mechanisms

Tan et al. designed a DDoS detection mechanism using KMeans and KNN in Tan et al. (2020). It consists of a KMeans-based training data processing module and a KNN-based traffic detection module. The KMeans algorithm is only utilized in the training phase to classify the similar instances into different categories. In the detection phase, the measured instance is first normalized. Then the distances between that instance and the cluster centers are calculated. At last, the measured instance is identified as normal or abnormal according to the labels of k points closest to the measured instance. The simulation results show that the mechanism performs with higher accuracy, higher recall, and lower false positive rate than the entropy method and distributed SOM.

6.9. Discussion of combination of multiple methods-based DDoS detection mechanisms

Table 10 lists a summary of the combination of multiple methods-based DDoS detection mechanisms. Generally, these combination of multiple methods-based DDoS detection mechanisms can be classified into two kinds. The most popular approach is first using a simple method to coarse-grained detect DDoS and then employing a complex method to fine-grained detect DDoS. For instance, in Liu et al. (2019), the entropy of all destination IP addresses is first calculated and compared with a pre-defined threshold. If the entropy is less than that threshold, the BPNN-based DDoS detection method will be initiated to further detect DDoS. Most of the existing combination of multiple methods-based DDoS detection methods (Phan and Park, 2019; Phan et al., 2016; Liu et al., 2019; Dayal and Srivastava, 2018; Yang and Zhao, 2018; Latah and Toker, 2018; Shang et al., 2017; Dang-Van and Truong-Thu, 2017) belong to this kind of detection technology.

Another kind of combination of multiple methods-based DDoS detection mechanism preprocesses the training instances to reduce the computation complexity. For instance, in Nam et al. (2018), the SOM is employed as a layer between the original data and KNN, which can significantly reduce the computation complexity.

Using a simple pre-detection method, the combination of multiple methods-based DDoS detection mechanism can significantly reduce the unnecessary execution of the DDoS detection method. Meanwhile, the different detection methods can be deployed in different locations, which can further improve the detection sensitivity and reduce the workload. For instance, in Liu et al. (2019), an entropy-based DDoS detection is deployed on the switch to execute pre-detection while a PSO-BPNN-based DDoS detection is deployed on the controller to further detect DDoS more granularly. However, the detection performance of the combination of multiple methods-based DDoS detection mechanism is notably affected by the first simple pre-detection method, especially on the false negative rate. Hence, the threshold utilized in the pre-detection method should be carefully selected to ensure a low false negative rate.

7. Threshold-based DDoS detection mechanisms

The threshold-based DDoS detection mechanism is one of the most popular DDoS detection mechanisms. From 2014 to 2020, 53 mechanisms that are designed based on threshold have been proposed, which accounts for about 35 percent of all DDoS detection mechanisms. According to the features used to compare with the threshold, these mechanisms are divided into three categories: network status-based

Table 10
Summary of combination of multiple methods-based DDoS detection mechanisms.

Reference	Year	Location	Method	Attack type	Experiments	Scale	Detection results
Yang and Zhao (2018)	2018	Controller	SVM, threshold	DoS	–	–	DA:99.8
Latah and Toker (2018)	2018	Controller	SVM, threshold	–	Simulation	Small(4 switches)	DA:100, FA:0
Shang et al. (2017)	2017	Controller	SVM, threshold	UDP flood	Experiment and simulation	Small(5 switches)	DR:96, FA:5
Shen et al. (2020)	2020	Controller	SVM, threshold	–	Simulation	Small(7 switches)	FA:1.48
Dang-Van and Truong-Thu (2017)	2017	Server	Fuzzy logic, threshold	–	–	–	FA:5, DR:97
Liu et al. (2019)	2019	Switch and controller	BPNN, Threshold, PSO	UDP/TCP/ICMP flood	Simulation	Small(6 switches)	DR:97.47, FA:1.43, DA:98.02
Dayal and Srivastava (2018)	2018	Controller	RBF, PSO, and threshold	TCP/UDP/ICMP/HTTP flood, smurf	Simulation	Claranet topology	DA:99.83
Phan and Park (2019)	2019	Controller	SOM, SVM	TCP/ICMP flood	Experiment	Laboratory LAN	DR:99.27, DA:99.3, FA:0.67
Phan et al. (2016)	2016	Controller	SOM, SVM	TCP/ICMP flood	–	Small(1 switch)	DR:98.13, DA:97.6, FA:3.85
Nam et al. (2018)	2018	Controller	SOM, KNN	–	Experiment	Small(1 switch)	DR:98.24, FA:2.14
Sudar and Deepalakshmi (2020)	2020	Controller	entropy, C4.5	–	Simulation	Small(8 switches)	DA:98.25, DR:98.1, TNR:98.4

DDoS detection mechanisms (Latah and Toker, 2018; Hyder and Lung, 2018; Chen et al., 2018b; Kalkan et al., 2018; Wang et al., 2018b; Zheng et al., 2018; Conti et al., 2019; Viet et al., 2017; Mousavi and St-Hilaire, 2018; Kumar et al., 2018b; David and Thomas, 2019; Guesmi and Saidane, 2017; Lin et al., 2017; Sambandam et al., 2018; Liu et al., 2017b; Boite et al., 2017; Huong and Thanh, 2017; Yang et al., 2017; You et al., 2017; Wang et al., 2017; Pandikumar et al., 2017; Tsai et al., 2017; Buragohain and Medhi, 2016; Chen et al., 2016; Xing et al., 2016; Piedrahita et al., 2015; Van Trung et al., 2015; Wang et al., 2015a; Hommes et al., 2014; Duy and Pham, 2018; Lu and Wang, 2016; Murtuza and Asawa, 2018; Jiang et al., 2016; Rebecchi et al., 2019; Wang et al., 2016b; Bhushan and Gupta, 2018; Wang et al., 2018c; Sahoo et al., 2018a,b; Dehkordi and Soltanaghaei, 2020; Wu et al., 2020), server status-based DDoS detection mechanisms (Lukaseder et al., 2018; Hong et al., 2017; Lukaseder et al., 2017; Shtern et al., 2014), and client status-based DDoS detection mechanisms (Mohammadi et al., 2017; Gkountis et al., 2017; Dao et al., 2015, 2016). Hereinafter, we will clearly describe these mechanisms.

7.1. Network status-based DDoS detection mechanisms

A summary of the network status-based DDoS detection mechanisms is listed in Table 11. We will introduce these mechanisms in this section.

In Chen et al. (2018b), Chen et al. presented a mechanism named FlexProtect to detect the DDoS attacks in SDN. FlexProtect utilizes the rate of uncompleted TCP connections to identify DDoS attacks. All TCP packets sent to the server are redirected to a detector. The detector works as a TCP SYN proxy to handle all TCP packets. After a 3-way handshake is completed, the TCP packets will be rerouted to the server. If the detector finds that the rate of uncompleted connections reaches the limited threshold, it will report a DDoS attack. The Mininet and OpenvSwitch were utilized in the experiment. The evaluation shows that FlexProtect can achieve quite good performance in terms of the true positive ratio and true negative ratio.

In Kalkan et al. (2018), Kalkan et al. designed JESS, a joint entropy-based DDoS defense mechanism used in SDN. JESS is composed of three stages, including the normal stage, preparatory stage, and active mitigation stage. In the normal stage, the source IP address, destination IP address, source port, destination port, protocol type, packet size, time to live value, and TCP flag are chosen as the attributes. For each pair of attributes, a normal profile is generated. Then the joint entropy of that normal profile is calculated to be ready for detecting DDoS attacks. In the preparatory stage, the current profile is first generated. After that, the joint entropy of the current profile is calculated and compared with the stored joint entropy of the normal profile. If the difference exceeds a pre-defined threshold, a DDoS attack alarm will be generated. In the active mitigation stage, the attack packets will be dropped. The simulation was performed on Mininet and Ryu. Five different attacks including the TCP SYN flood attack, DNS amplification attack, NTP attack, generic attack, and mixed attack are considered in the simulation. The performance of JESS was evaluated in terms of the true positive rate, true negative rate, false positive rate, and false negative rate. The results show that JESS performs well with low requirements on resources.

In Wang et al. (2018b), a mechanism was presented by Wang et al. to detect and mitigate the link-flooding attacks in SDN. Instead of monitoring all links, it provides a target link selection approach to reduce the computing complexity. The flow density of each link is calculated based on the flow entries of switches. The top-k links with maximum flow density will be identified as the target links. These target links will be monitored to gather some necessary network information: the packet loss rate, round-trip time, and available bandwidth. If the packet loss rate and round-trip time are larger than the related thresholds while the available bandwidth is less than the threshold, a link-flooding attack is confirmed. Then the malicious traffic will be blocked by the SDN controller and switches. The proposed mechanism was deployed on Floodlight. Three topologies, including a global Hignwinds network topology gathered from the Topology Zoo, were considered in the evaluation. The performance was measured in form of the detection accuracy, overhead, and effectiveness. According to the evaluation results, the proposed mechanism can rapidly and accurately detect and mitigate link-flooding attacks.

Zheng et al. proposed a defense mechanism named RADAR against the Crossfire attack in SDN (Zheng et al., 2018). RADAR relies on the changes of links for detecting and locating the DDoS attacks. It is composed of three modules: RADAR collector, RADAR detector, and RADAR locator. The RADAR collector is responsible for gathering statistics of network flows. A Crossfire attack will be identified by the RADAR detector if (1) the number of times that the traffic acutely changes in the link exceeds a threshold, (2) the number of congested links in a path exceeds another threshold, and (3) the accumulated congestion duration of each link equals to the congestion duration of the path. At last, the RADAR locator identifies the Crossfire attack flows based on correlation analysis. RADAR was implemented on the Floodlight controller and Mininet. The Crossfire, SYN flooding, and

Table 11
Summary of network status-based DDoS detection mechanisms.

Reference	Year	Location	Method	Attack type	Experiments	Scale	Detection results
Latah and Toker (2018)	2018	Controller	Traffic features	–	Simulation	Small(4 switches)	DA:100, FA:0
Hyder and Lung (2018)	2018	Controller	Traffic features	UDP reflection attack	Simulation	–	–
Chen et al. (2018b)	2018	Controller	Traffic features	TCP flood	Simulation	Multiple topologies	DA:98, TNR:98
Kalkan et al. (2018)	2018	Controller	Traffic features	TCP flood, DNS amplification, NTP, Mixed attack	Simulation	Small(1 switch)	DA:90, FA:10
Wang et al. (2018b)	2018	Controller	Traffic features	–	Experiment	Multiple topologies	DR:100, DA:90
Zheng et al. (2018)	2018	Controller	Device status	Crossfire, TCP flood, DNS amplification	Simulation and experiment	Multiple topologies	TPR:95, FPR:5
Conti et al. (2019)	2019	Server	Traffic features	TCP/UDP flood	Simulation	Small(9 switches)	–
Viet et al. (2017)	2017	Switch	Traffic features	HTTP flood	Experiment	Small(1 switch)	DR:93
Mousavi and St-Hilaire (2018)	2018	Controller	OpenFlow features	Self-designed DDoS	Simulation	Small(9 switches)	–
Kumar et al. (2018b)	2018	Controller	Traffic features	TCP flood	Simulation	Small(11 switches)	DR:100, FA:27
David and Thomas (2019)	2019	–	Traffic features	–	–	–	DA:99.5, DR:99.5, TNR:99.5
Guesmi and Saidane (2017)	2017	Controller	Traffic features	Self-designed DDoS	Simulation	–	–
Gurusamy and MSK (2019)	2019	–	Device status	UDP flood	Simulation	–	–
Lin et al. (2017)	2017	–	Device status	TCP flood	Simulation	Data center	–
Xu et al. (2017)	2017	Server	Device status	–	Simulation	–	PRE:86.32
Sambandam et al. (2018)	2018	Controller	Traffic features	–	Experiment	Small(2 switches)	–
Liu et al. (2017b)	2017	Controller and switch	Traffic features	DNS amplification	Simulation	Middle(26 switches)	–
Boite et al. (2017)	2017	Switch	Traffic features	TCP/HTTP flood	Simulation	Small(1 switch)	DR:100
Huong and Thanh (2017)	2017	Server	Traffic features	TCP/ICMP flood	–	–	FA:6.4
Yang et al. (2017)	2017	Switch	Traffic features	–	Experiment	Small(1 switch)	–
You et al. (2017)	2017	Controller	Traffic features	UDP flood	Simulation	Middle(25 switches)	–
Wang et al. (2017)	2017	Controller	OpenFlow features	ICMP flood	Simulation	Small(4 switches)	–
Pandikumar et al. (2017)	2017	Controller	OpenFlow features	UDP flood	Simulation	Small(3 switches)	–
Tsai et al. (2017)	2017	Server	Traffic features	Self-designed DDoS	Experiment	Small(2 switches)	–
Buragohain and Medhi (2016)	2016	Controller	Traffic features	UDP/ICMP flood	Simulation	Data center	–
Chen et al. (2016)	2016	Server	OpenFlow and Traffic features	TCP flood	Simulation	–	FA:0.14, FN:1.03
Xing et al. (2016)	2016	Controller and switch	Traffic features	DNS reflection attack	Simulation	Data center	–
Piedrahita et al. (2015)	2015	Server	Device status	Self-designed DoS	Simulation	Small(2 switches)	–
Van Trung et al. (2015)	2015	Server	Traffic features	–	–	–	FA:5, DR:97
Wang et al. (2015a)	2015	Switch	Traffic features	–	Simulation	Small(7 switches)	DR:100, FA:25
Hommes et al. (2014)	2014	–	Device status	–	Experiment	Small(8 switches)	DR:90, PRE:70
Duy and Pham (2018)	2018	Controller	Traffic features	UDP flood	Simulation	Small(9 switches)	–
Lu and Wang (2016)	2016	Server	Traffic features	–	Simulation	Small(3 switches)	–
Murtuza and Asawa (2018)	2018	–	Traffic features, network device status	TCP/ICMP flood	–	–	DA:97.31
Jiang et al. (2016)	2016	Controller	Traffic features	–	Simulation	Small(3 switches)	–
Rebecchi et al. (2019)	2019	Controller and switch	Traffic features	TCP/HTTP flood	Simulation	Small(1 switch)	DR:100
Wang et al. (2016b)	2016	Controller and switch	Traffic features	Self-designed LFA	–	Multiple topologies	–
Bhushan and Gupta (2018)	2018	Controller and switch	Traffic features and device status	–	Simulation	Small(10 switches)	–
Wang et al. (2018c)	2018	Controller	Device status	Self-designed LFA	Simulation	Multiple topologies	PRE:96.88, DR:100, F1:98.41

(*continued on next page*)

Table 11 (*continued*).

Reference	Year	Location	Method	Attack type	Experiments	Scale	Detection results
Sahoo et al. (2018a)	2018	–	Traffic features	–	Simulation	Small(10 switches)	–
Sahoo et al. (2018b)	2018	Controller	Traffic features	UDP flood	Simulation	Small(8 switches)	FN:0
Dehkordi and Soltanaghaei (2020)	2020	Controller	Traffic features	–	Simulation	–	DA:99.95
Wu et al. (2020)	2020	Controller	Traffic features	Low-rate DDoS	Simulation	Small(4 switches)	PRE:95, DA:95.8, DR:94.6, AUC:93.8
Mishra et al. (2021)	2021	Controller	Traffic features	UDP flood	Simulation	Small(9 switches)	DR:98.2, FA:4
Rahouti et al. (2021)	2021	Switch	Traffic features	TCP flood	Experiment	Small(4 switches)	–
Agrawal and Tapaswi (2021)	2021	Controller	Traffic features	Low-rate DDoS	Simulation	Middle(28 switches)	DA:97.6, FN:3.18, FA:3.29

Table 12
Summary of server status-based DDoS detection mechanisms.

Reference	Year	Location	Method	Attack type	Experiments	Scale	Detection results
Lukaseder et al. (2018)	2018	–	Availability of network service	–	–	–	–
Hong et al. (2017)	2018	Controller	Connection status of server	HTTP flood	Simulation	–	–
Lukaseder et al. (2017)	2018	Controller	Response time of server	TCP/HTTP/TLS flood	–	–	–
Shtern et al. (2014)	2014	Server	Workload of server	–	Analysis	–	–

DNS amplification attacks were considered in the experiment. The performance of RADAR was measured in terms of detection accuracy, delay, and overhead. The outcomes prove that RADAR can effectively and quickly detect different DDoS attacks.

Xu et al. also designed a threshold-based DDoS detection method (Xu et al., 2017). The main idea of this work is comparing the request rate and matching efficiency of switch with two thresholds to identify the DDoS attacks. In Wang et al. (2017), Wang et al. designed a mechanism named SECO for detecting DDoS attacks in SDN. The basic principle of SECO is that it detects the DDoS attack by comparing the number of *packet-in* message with a threshold.

The detection approach proposed in Xing et al. (2016) confirms a DDoS attack by comparing the difference of the number of DNS request and response packets with a threshold. In Hommes et al. (2014), the author uses the flow entries in the switches as an indicator of the network status. Hence, two time-instant probability distributions are calculated by the number of flow entries in a single switch and the total number of flow entries. Then the Hellinger distance and Kullback–Leibler divergence of these two probability distributions are computed with a threshold to determine whether there is a DDoS attack. In Wang et al. (2018c), Wang et al. proposed a threshold-based mechanism for detecting the link-flooding attack. In this work, each link is assigned a score. If the score of current congested links exceeds a threshold, it signals an LFA DDoS. In Mishra et al. (2021), three thresholds have been defined to determine whether the flow rate, entropy, and packet count are abnormal. Only when the flow rate and packet count are bigger than related thresholds while entropy is less than its threshold, it will report a DDoS attack. According to the record of attacks, a dynamic attack detection threshold adjusting method have been proposed in Rahouti et al. (2021). The evaluation was performed on a real-world testbed, which demonstrate the effectiveness of the proposed method.

7.2. Server status-based threshold DDoS detection mechanisms

Table 12 lists a summary of the server status-based DDoS detection mechanisms used in SDN. These mechanisms are described in this section.

In Lukaseder et al. (2018), a mechanism that aims to mitigate the slow DDoS attacks in SDN has been designed by Lukaseder et al. It is fabricated upon the idea that once a server cannot provide normal service, it will be inferred that the server is suffering from the DDoS attacks. Hence, the proposed mechanism monitors the status of servers to detect DDoS attacks. Six methods are designed to identify the DDoS attackers based on checking the long connections, low packet rate, packet distance uniformity, combination of low packet rate and packet distance uniformity, low mean packet rate, and low packet rate variance, respectively. The effectiveness of these six methods was measured in terms of the detection accuracy and detection time. According to the results, the method that jointly checks the low packet rate and packet distance uniformity achieves the best detection accuracy.

Hong et al. designed a mechanism named SHDA for defending the slow HTTP DDoS attacks in SDN in Hong et al. (2017). The main idea of this mechanism is that it uses the server status and a timeout approach to detect DDoS attacks. If the number of open HTTP connections is larger than a threshold, the server is suspected to be attacked. Then the SHDA will check whether the uncompleted HTTP request will be completed within a predefined time. An HTTP connection that is not complete within that time is recognized as a DDoS attack. For the simulation, NS3 was utilized to evaluate the effectiveness of SHDA. The outcomes demonstrate that SHDA can efficiently detect and block slow HTTP DDoS attacks.

In Lukaseder et al. (2017), a framework for detecting and mitigating DDoS attacks in SDN has been presented by Lukaseder et al. It works on the idea of identifying a DDoS attack by checking the availability of the services provided by the server. The HTTP request is utilized to estimate the response time of a service. The whole delay, partial delay, and TCP round trip time are collected and compared with different thresholds. Once one of these delays exceeds its threshold or an HTTP 503 code is received, it assumes the service is abnormal. Then the load of that server is collected. If it exceeds a pre-defined threshold, a DDoS attack is detected. For the simulation, the RYU controller was used. The detection time, mitigation time, and server downtime were considered as the measurement when testing its performance. The prototype is shown to be effective when detecting and mitigating DDoS attacks.

An approach against the Low and Slow application DDoS (LSDDoS) attack has been introduced by Shtern et al. in Shtern et al. (2014). The basic principle of this approach is to measure the discrepancies between the actual and expected workload. Hence, the CPU utilization, CPU time, disk utilization, disk time, waiting time, and throughput are recognized as the workload metrics. A performance model is built

Table 13
Summary of client status-based DDoS detection mechanisms.

Reference	Year	Location	Method	Attack type	Experiments	Scale	Detection results
Mohammadi et al. (2017)	2017	Controller	Packets sent by clients	TCP flood	Simulation	Small(11 switches)	DR:100, FA:1
Gkountis et al. (2017)	2017	Controller	Packets sent by clients	–	Simulation	Small(1 switch)	–
Dao et al. (2015)	2015	–	Packets sent by clients	–	–	Small(1 switch)	–
Dao et al. (2016)	2016	Controller	Packets sent by clients	–	Simulation	Small(1 switch)	–

based on these metrics, which can estimate the current workload. Once the discrepancies between the actual and expected workload exceed a threshold, a LSDDoS attack will be recognized.

7.3. Client status-based threshold DDoS detection mechanisms

A summary of the client status-based DDoS detection mechanisms is listed in Table 13. We will introduce these mechanisms in this section.

To defense the TCP SYN flooding DDoS attacks, the SLICOTS has been designed by Mohammadi et al. in Mohammadi et al. (2017). The main idea of SLICOTS is that it identifies the DDoS attacks by analyzing the TCP packets sent by clients. It calculates and checks the number of illegitimate SYN packets and RST packets sent from a client. If these numbers exceed the pre-defined thresholds, an SYN flooding attack will be detected. For the simulation, the OpenDayLight controller and Mininet were considered. The effectiveness of SLICOTS was compared with OPERETTA, an SYN flooding attack defense mechanism. The outcomes prove that SLICOTS performs better than OPERETTA in terms of the attack detection time, number of installed flow entries, and CPU utilization.

In Gkountis et al. (2017), Gkountis et al. designed a lightweight mechanism against DDoS attacks in SDN. It focuses on the detection and mitigation of DDoS attack initiating with IP spoofing. The number of packets sent by an IP address and the average byte per flow are compared with two different thresholds. Specifically, the number of packets is first periodically checked. If it is less than the threshold, a DDoS attack is detected. Otherwise, the average bytes per flow will be calculated and compared with another threshold. If it is less than the threshold, the detection mechanism identifies a DDoS attack. The Mininet and POX controller were utilized in the simulation. The performance of the proposed mechanism was measured in form of bandwidth utilization, number of installed flow entries, and CPU consumption. It is shown that the designed mechanism can detect DDoS attacks with lightweight overload.

Dao et al. proposed a method against the DDoS attacks in SDN in Dao et al. (2015, 2016). It works on the analysis of DDoS behavior, which can be characterized as generating massive connections in a short time, while each connection has few packets. Hence, it establishes two thresholds, k and n. When the number of connections generated by a user is larger than k, it further compares the average number of packet of each connection s with n. If s is less than n, a DDoS attack is confirmed. A topology that contains one switch and four clients was built to evaluate the DDoS attack. The number of flow entries in a switch, number of packets received by the controller, and bandwidth occupation of the controller-switch channel were considered as the metrics to evaluate the proposed method. The proposed mechanism enhances the availability of the SDN controller, which has been proved in the experiment, in terms of the number of flow entries in the switch and number of packets sent to the controller from the switch.

7.4. Discussion of threshold-based DDoS detection mechanisms

Generally, the threshold-based DDoS detection mechanisms consist of three steps. In the first step, it obtains the necessary information from the network traffic or flow entries of network devices. The second step extracts and calculates features from these information. At last, these features are compared with a threshold. If it is larger or less than the pre-defined threshold, a DDoS attack will be identified.

As mentioned above, the threshold-based DDoS detection mechanisms can be classified into the network status-based, server status-based, and client status-based DDoS detection mechanisms. Among these mechanisms, the network status-based DDoS detection mechanisms are most popular. Features used in the network status-based DDoS detection mechanisms are often extracted from the network traffic and flow entries. The main features include the number of half-open TCP connection, number of DNS request packets, number of flow entries, occupied bandwidth of switches, traffic rate, traffic asymmetry, entropy of source IP address, and entropy of destination IP address.

Compared with the network status-based DDoS detection mechanisms, the number of server status and client status-based DDoS detection mechanisms are small. The response time, workload, and number of half-open TCP connection of server are considered as the features used in the server status-based DDoS detection mechanisms. For the client status-based DDoS detection mechanisms, it always employs the number of packets or bytes sent by a host to execute DDoS detection.

Although the network status-based DDoS detection mechanisms are prevalent, only using these mechanisms may cause high false positive rate, due to its simple analysis and process. Subsequently, when designing the threshold-based DDoS detection mechanisms, it is better to employ the threshold-based DDoS detection as one step of the detection mechanism, especially for the low-rate DDoS attack.

8. Other method-based DDoS detection mechanisms

In addition to the machine learning, statistical method, threshold, and combination of multiple methods-based DDoS detection mechanisms, other technologies are also used when detecting DDoS attacks in SDN. For instance, the catastrophe theory (Guo et al., 2019), bloom filter (Xiao et al., 2016), third-party (Chin et al., 2015; Manso et al., 2019), consistency (De Assis et al., 2017), similarity (Yin et al., 2018), fuzzy evaluation (Yan et al., 2016b; Wang et al., 2018d), probabilistic transition (Ivannikova et al., 2017), graph model (Aleroud and Alsmadi, 2016; Wang et al., 2015b), and self-designed (Wei et al., 2016) method have been considered. In this section, we will elaborate on how the researchers use these technologies to detect DDoS attacks in SDN.

8.1. Distributed detection mechanisms

In Rathore et al. (2019), Rathore proposed a Blockchain-based DoS attack detection mechanism named BlockSecIoTNet. The main idea of this work is that it considers different SDN controllers as fog nodes and dynamically updates the attack detection models at the fog nodes. A central cloud server is introduced to act as a manager which manages the attack models from a set of fog nodes, while the fog nodes act as processing or proofing nodes to detect attack. The POX, Mininet, and Amazon EC2 cloud were employed to evaluate BlockSecIoTNet. The performance is measured in the form of accuracy, detection rate, F-score and so on. As per the results, BlockSecIoTNet provides good accuracy with effectively detecting DDoS.

In Ujjan et al. (2019), Ujjan. et al. designed a Blockchain-based mechanism used to detect DDoS attack in SDN. It utilizes the capabilities of Blockchain to build a collaborative detection system. The Snort nodes that are utilized to detect attack communicate with each other for sharing the rule sets. The simulation was performed on the RYU controller and Mininet. Metaslploit was used to launch the DDoS attack. The evaluation shows that the proposed mechanism achieves accurate detection rate (96%).

A smart contract-based DDoS/DoS attack detection mechanism was proposed by Shao et al. in Shao et al. (2019). The Blockchain is utilized to store the status of SDN controller. Each controller maintains an analysis center to identify the status of local network. A global view is formed by uploading data from different controllers. When a DDoS attack is detected, a primary controller will be selected to perform mitigation, using consensus set and the trust mechanism of the consensus algorithm. The effectiveness of the proposed mechanism is verified using Matlab. According to simulation results, this mechanism provides efficient consensus time and signaling overhead.

A distributed intrusion detection method has been designed for detecting DoS, user to root, remote to local, and probing attack in Shu et al. (2020). The proposed detection method consists of training phase and detection phase. In training phase, a discriminator is built by the cloud server and sent to each SDN controller. When detecting attack, the SDN controller will calculate an anomaly score for each flow. If that score is larger than a threshold, that flow will be identified as attack flow. As per the evaluation results, the proposed method shows better performance as compared to centralized detection methods in terms of accuracy, precision, recall, F1, and AUC.

In Nguyen et al. (2019), Nguyen et al. presented a collaborative intrusion detection system. The objective of the proposed system is to detect attacks including the DoS/DDoS attack, men in the middle attack, spoofing attack and so on. The designed system is composed of three layers of IDS including Edge-IDS, Fog-IDS, and Cloud-IDS. Edge-IDS utilizes lightweight machine learning-based detection algorithms to capture the traffic statistics information. Fog-IDS employs machine learning algorithms with moderate complexity to detect attacks. Cloud-IDS performs machine learning-based detection with high complexity. The simulation is performed using the MaxiNet, ONOS, and Containernet. According to the evaluation, the proposed system provides efficient results in terms of detection rate, precision, accuracy, and false alarm rate.

8.2. Catastrophe theory-based DDoS detection mechanisms

Based on the catastrophe theory, a method called CATH for detecting DDoS has been proposed by Guo et al. in Guo et al. (2019). The basic principle of CATH is that it considers the changes in the SDN network state as a catastrophe process. According to this idea, CATH estimates a cusp catastrophe model by defining the catastrophe potential function and estimating model parameters. Specifically, the features used in CATH include the number of packets in a single flow, number of bytes of a single flow, source IP growing speed, port generating speed, and flow table matching ratio. Subsequently, whether the network is in a DoS state can be inferred based on the catastrophe model. The performance of CATH was measured using the Mininet and POX controller. The results show that CATH can detect the TCP SYN flooding attack, UDP flooding attack, and ICMP flooding attack with a true positive rate up to 90%.

8.3. Bloom filter-based DDoS detection mechanisms

In Xiao et al. (2016), Xiao et al. designed a DDoS detection approach used in SDN. The main contribution of this work is they introduce the Bloom filter to detect DDoS in SDN. The duration, packet count, and byte count of DDoS flows are captured from the flow entries of the switches. Based on these features, a Bloom filter S is established to represent an anomaly DDoS attack. Subsequently, the designed approach can detect DDoS attacks by checking whether the features of a flow belong to S. For the experiment, Mininet and Iperf were utilized to simulate the network and DDoS attack. The effectiveness of this approach was evaluated in terms of the false positive rate, detection probability, insert time, and detection time. According to the outcomes, this approach can detect DDoS attacks with an accuracy more than 96%. Meanwhile, it can reduce the insert time and detection time, compared with the hashMap based DDoS detection approach.

8.4. Third-party software-based DDoS detection mechanisms

Chin et al. presented an approach for detecting DDoS attacks in SDN (Chin et al., 2015). The presented approach is composed of three modules: (1) a monitor to achieve the DDoS attack detection in the switch, (2) a correlator to further confirm the DDoS attack, and (3) a controller to perform blocking or rerouting. In the monitor, Snort, a third-party software, is utilized as the detection module. Once the Snort signals a DDoS attack, the correlator will confirm the DDoS attack by checking whether there are spoofed IP addresses in the switch port. If it does not find the spoofed IP addresses, the DDoS attack alarm generated by the monitor will be recorded as a false alarm. The performance of the designed approach was measured in terms of the detection time and detection accuracy. From the experiment results, it can be seen that less communications make the proposed approach become fast but less accurate. Hence, when using this approach, the trade-off between the detection time and detection accuracy must be determined.

In Manso et al. (2019), Manso et al. also presented a system for detecting the DDoS attacks in SDN using the third-party software. Snort is utilized as the DDoS detection module in this system, in which the authors define a set of pre-configured rules to detect DDoS. The DDoS mitigation time, average RTT and packet loss are utilized in the evaluation, which demonstrates the effectiveness of the presented system.

8.5. Consistency-based DDoS detection mechanisms

In De Assis et al. (2017), a DDoS detection method that is designed based on the information consistency has been proposed by De Assis et al. In this work, the bit rate, packet rate, flow rate, entropy of the source IP addresses, destination IP addresses, source port, and destination port are extracted as the features of the network. The Holt-Winters method is used to generate a signature. Subsequently, the signatures of the IP flows are collected, calculated and compared with the generated signature every one minute. If the observed signatures differ from the generated one, the approach further compares the observed signatures with the signatures of the known DDoS attacks. The proposed detection method was evaluated in terms of the detection accuracy and detection precision. In the experiment, it can attain the detection accuracy higher than 98%.

8.6. Similarity-based DDoS detection mechanisms

Yin et al. proposed a DDoS attack detection approach for the software-defined internet of things in Yin et al. (2018). The essential idea of this approach is to detect DDoS attacks by analyzing the rate of *packet-in* message. Hence, it first obtains the *packet-in* message rate set S. Subsequently, it divides S into two categories: X and Y. At last, the cosine similarity between X and Y is calculated. If that cosine similarity is larger than a threshold, a DDoS attack will be identified. The Floodlight and Mininet were used as the controller and network simulator in the experiment. The number of flow table items, *packet-in* messages, packets received by the controller, and the bandwidth of controller-switch channels were employed to evaluate the proposed approach. According to the outcomes, it is proved to be able to improve the performance of the internet of things under the DDoS attacks.

Table 14
Summary of other method-based DDoS detection mechanisms.

Reference	Year	Location	Method	Attack type	Experiments	Scale	Detection results
Guo et al. (2019)	2019	Controller	Catastrophe theory	Self-designed DDoS	Simulation	Multiple topologies	DR:96.2, FA:2.97
Xiao et al. (2016)	2016	Controller	Bloom filter	–	Simulation	Data center	DA:96
Chin et al. (2015)	2015	Application server	Third-party software	TCP flood	Simulation	Small(4 switches)	–
Manso et al. (2019)	2019	Application server	Third-party software	TCP flood	Simulation	–	–
De Assis et al. (2017)	2017	–	Information consistency	UDP flood	Experiment	University LAN	DA:98
Yin et al. (2018)	2018	Controller	Cosine similarity	UDP flood	Simulation	IoT	–
Yan et al. (2016b)	2016	Controller	Fuzzy evaluation	Malformed packets flood, UDP flood	Simulation	Small(1 switch)	–
Wang et al. (2018d)	2018	Controller	Fuzzy synthetic evaluation decision	TCP/UDP flood	Experiment and simulation	Multiple topologies	–
Ivannikova et al. (2017)	2017	–	Probabilistic transition	–	Simulation	Cloud computing	DA:99.58, DR:98.66, FA:0
Aleroud and Alsmadi (2016)	2016	Controller	Graph model	TCP/UDP/ICMP flood	Experiment	Small(1 switch)	–
Wang et al. (2015b)	2015	Controller	Graph model	–	Simulation	Cloud computing	DR:89.3
Wei et al. (2016)	2016	Controller	Self-designed rules	UDP flood	Simulation	Small(1 switch)	–

8.7. Fuzzy Evaluation-based DDoS detection mechanisms

A DDoS attack detection mechanism has been presented by Yan et al. in Yan et al. (2016b). In this work, the DDoS attack detection is regarded as a vague process. Considering that some characteristics which can be used to reflect the network status will change when a DDoS attack occurs, these characteristics can be employed as the factors of the fuzzy synthetic evaluation decision-making model. Accordingly, the flow request rate, entropy of destination ports, entropy of destination IP addresses, and entropy of source IP addresses are chosen as these factors. Then the judgment sets and a single-factor judgment matrix are built upon these factors. Finally, a comprehensive judgment is made to identify the DDoS attacks. For the experiment, Mininet and POX were utilized as the network simulator and controller, respectively. The malformed packets flooding attack and uncommon protocol packets flooding attack are generated by the Scapy. According to the results, the designed DDoS detection mechanism is validated to be lightweight and effective when detecting the DDoS attacks.

In Wang et al. (2018d), Wang et al. also employ the fuzzy synthetic evaluation decision-making model for detecting DDoS attacks in SDN. Different from Yan et al. (2016b), the factors used in this work are the percentage of flows with a small number of packets, percentage of flows with small average bytes, percentage of flows with a short time duration, percentage of reversible flows, growth rate of irreversible flows, and growth rate of ports. A comprehensive judgment score of each switch is used to reveal the attacking severity of each switch.

8.8. Probabilistic transition-based DDoS detection mechanisms

A probabilistic transition-based DDoS detection approach has been proposed by Ivannikova et al. in Ivannikova et al. (2017). The essential principle of this approach is that it calculates the joint probability of each session in the network to detect DDoS attacks. The duration of a session, number of packets sent in one second, number of bytes sent in one second, average packet size, and presence of packets with different TCP flags are selected as the features of a session. A clustering algorithm is adopted to assign a unique label to a group of features. The conversations with the same source and destination IP address and destination port are regarded as a user session. The conditional and marginal probabilities of each session are calculated based on the cluster labels of the session to be detected. At last, the joint probability is calculated and compared with a threshold to identify whether there is a DDoS attack. The experiment was carried out on the OpenDayLight controller and OpenvSwitch. The outcomes demonstrated that the proposed approach outperforms other compared algorithms in terms of the detection accuracy.

8.9. Graph model-based DDoS detection mechanisms

Aleroud et al. designed an approach against the DDoS attacks in SDN (Aleroud and Alsmadi, 2016). It works on the idea of combining the clustering algorithms and graph model to identify DDoS attacks. For a flow fl which is need to be detected, the most similar node a_i is first selected based on the feature similarity. Then k nodes R_k similar to a_i are chosen from the graph model. At last, in R_k, if the number of suspicious nodes exceeds the number of benign nodes, fl is determined to be a DoS attack flow. The ICMP, UDP, and TCP SYN flooding attack were generated to verify the effectiveness of the proposed approach. It is proved to perform better than KNN in terms of the true positive rate, false positive rate, and false negative rate.

In Wang et al. (2015b), Wang et al. proposed an approach named DaMask against the DDoS attacks in SDN. The main idea of this approach is that it builds a Chow-Liu tree based on the features automatically selected from the network traces. Hence, the optimal assignment of features can be approximate to a calculation of the conditional probability query. Moreover, to overcome the dataset shift problem, a graph model update method that combines the global update and local update based on the difference between the new attacks and existing ones. DaMask was evaluated on the Mininet and Floodlight controller. According to the results, DaMask has the capability to accurately detect DDoS attacks with low overhead.

8.10. Self-designed DDoS detection mechanisms

For preventing the UDP flooding DDoS attacks in SDN, a lightweight countermeasure has been presented by Wei et al. in Wei et al. (2016). It works on the idea of detecting DDoS attacks by comparing the number of packets received by a port with the number of packets sent by the same port. If the former is bigger, it signals a DDoS attack alert and drops the DDoS packets. The Mininet and RYU were employed in the experiment. The proposed method was evaluated by measuring the network throughput, which showed it performs better than the FloodGuard.

8.11. Discussion of other method-based DDoS detection mechanisms

As shown in Table 14, there are many kinds of other methods-based DDoS detection mechanisms. It is hard to summarize the common process of the other methods-based DDoS detection mechanisms.

When designing the other method-based DDoS detection mechanism, the selection of a basic algorithm is the key factor that seriously

affects the detection performance. Meanwhile, selecting a basic algorithm from a number of algorithms is also difficult. Our recommendation is that the researchers can select the basic algorithm based on the characteristics of DDoS attack. For instance, by analyzing the catastrophe characteristics of SDN flows, the authors in Guo et al. (2019) designed a catastrophe theory-based DDoS detection mechanism.

Except that, using decentralized detection methods may be a good choice, especially for the SDN with multi-domain. The decentralized detection method has advantages such as it can share attack information between different detection systems and it can also deploy attack detection method closed to the attackers, facilitating the efficient detection and quick response of DDoS attack.

9. Discussion and analysis

9.1. Discussion of traditional DDoS detection mechanisms and DDoS detection mechanisms in SDN

As described in Section 3, the DDoS attack aimed at SDN exploits the vulnerabilities of SDN, such as the separation of forwarding function and control function, which makes it performs different with the traditional DDoS attack. However, considering that it still belongs to DDoS attack, the DDoS attack aimed at SDN also has some similarity with tradition DDoS attack. The main differences between the DDoS attack aimed at SDN and traditional DDoS attack include the attack target, attack method, and attack effect. For instance, a DDoS attack aimed at the SDN controller tries to overwhelm the controller's ability of processing *packet-in* messages. Thus, the adversary sends a lot of unmatched packets to the switches, making the switches send a number of *packet-in* messages to the controller. In that case, the buffer used to store *packet-in* messages will crash, resulting in the controller cannot process the legitimate *packet-in* messages. But there are also some similarities between the DDoS attack aimed at the SDN controller and traditional DDoS attack. For example, in the both attacks, the adversary needs to send DDoS packets from distributed hosts to one target and they both block the target from working.

Due to the different attack method and attack effect between the DDoS attack aimed at SDN and traditional DDoS attack, there are also differences between the related DDoS detection mechanisms. Generally, both the DDoS detection mechanisms specific to SDN and traditional DDoS detection mechanisms can be classified into three steps, including (1) information collection, (2) feature calculation, and (3) DDoS detection. Next, we will discuss the DDoS attack detection mechanisms specific to SDN and traditional DDoS attack detection mechanisms from these steps.

The traditional DDoS detection mechanisms usually use packet sampling to collect information utilized to detect DDoS, such as the NetFlow and sFlow. Except the packet sampling, SDN provides the southbound API that can collect information for DDoS detection. For instance, when using OpenFlow as the southbound API, the SDN controller can send *flow-stats-request* messages to the switches to capture the flow entries statistics of the switches, which can be used to detect DDoS. Hence, in information collection, the DDoS detection mechanisms specific to SDN can use the SDN specified collection method that cannot utilized in traditional network to capture detection data.

Except the differences in information collection, the features used to detect DDoS attack aimed at SDN and traditional DDoS attack are also different. We take the DDoS attack aimed at the SDN switch as an example. In order to launch such a DDoS attack, the adversary will change the tuples used to match flow entries to make the controller add new flow entries on the target switch. Then, the adversary will send the same packets at a low rate to prevent deletion of the new flow entries. Therefore, when detecting that kind of DDoS, the features which can represent that process will be used. For instance, considering the changes of tuples, number of flow entries with the same source IP address may be a valuable feature utilized to detect DDoS attack aimed at the SDN switch. In contrast, when detecting the traditional DDoS attack, that feature may be useless.

Once the features are calculated, the traditional DDoS detection mechanisms and DDoS detection mechanisms used in SDN both need to identify the DDoS attack based on detection model. In this step, We think that there are no such big differences between these two kinds of DDoS detection mechanisms as the information collection step and feature calculation step. For instance, when using SVM as the detection model, both the DDoS detection mechanisms will execute the same operations, including (1) building an SVM model, (2) using labeled features to train the SVM model, and (3) employing the trained SVM model to detect DDoS attack. In the above executions, although the architecture of SVM model is effected by the features, how to determine the architecture of SVM model is the same. Therefore, we think that there are little differences between the detection models used in traditional DDoS detection mechanisms and detection mechanisms specific to SDN.

In conclusion, how to collect information and how to choose features used in DDoS detection are quite different for traditional DDoS detection mechanisms and DDoS detection mechanisms specific to SDN. However, there are no significant difference between the detection models used in these two detection mechanisms.

9.2. Analysis of DDoS detection mechanisms

Table 15 show the analysis of DDoS detection mechanisms In SDN. In Table 15, the proportion of the machine learning-based, statistics-based, combination of multiple methods-based, threshold-based, and other method-based DDoS detection mechanisms has been illustrated. A conclusion drawn from this table is that the machine learning-based DDoS detection and threshold-based DDoS detection mechanisms are the two most popular technologies utilized to detect DDoS attacks in SDN. Specifically, there are 70 machine learning-based DDoS detection mechanisms, which account for 49% of all DDoS detection mechanisms in our survey. Meanwhile, 53 threshold-based DDoS detection mechanisms have been presented, with a proportion of 35%.

For the machine learning-based DDoS detection mechanism, it can be further classified into the neural network-based, classifying-based, clustering-based, deep learning-based, and ensemble learning-based DDoS detection methods. Among these subtypes, the occurrence numbers of classifying algorithm-based DDoS detection mechanism (27/18.9%) and neural network-based DDoS detection mechanism (23/16.1%) are highest.

The most popular algorithm employed in the classifying-based DDoS detection mechanisms is SVM. Precisely, 24 SVM-based DDoS detection mechanisms have been published. This may be driven by the fact that SVM is a simple binary classification algorithm, which is natively appropriate for identifying whether there is a DDoS attack in SDN. Furthermore, as a classical classification algorithm, SVM has already been used for detecting network intrusion in the traditional network (Shon and Moon, 2007; Chen et al., 2005; Kim and Park, 2003; Mukkamala et al., 2002; Peddabachigari et al., 2007).

Among the neural network-based detection mechanisms, 10 mechanisms are SOM-based, which indicates that the SOM is the most popular neural network for the neural network-based DDoS detection. We think that there may be two reasons which lead to that phenomenon. The first one is that SOM can efficiently learn and abstract knowledge by unsupervised learning. The second one is that a number of previous works have already been employed SOM to detect network anomaly or classify network traffic. For instance, Braga et al. (a quite famous work about DDoS detection in SDN which is published in 2010) (Braga et al., 2010), Jiang et al. (2009), Powers and He (2008), Kayacik et al. (2007), Kiziloren and Germen (2007), DeLooze (2006), Liu and Yi (2006), Depren et al. (2005), and so on.

When designing a DDoS detection mechanism, the researchers should clearly aware that no algorithm can perfectly solve all questions

Table 15
Analysis of DDoS detection mechanisms in SDN.

Technologies	Number/ Percentage	Subtypes	Literatures	Number/ Percentage	Description
Machine learning	70/49.0%	Neural network	Cui et al. (2016), Chen and Yu (2016a), Braga et al. (2010), Phan and Park (2019), Phan et al. (2016, 2017), Wang and Chen (2017), Xu and Liu (2016), Zhao and Liu (2018), Nam et al. (2018), Pillutla and Arjunan (2019), Phan et al. (2019), Wang et al. (2019), Cui et al. (2018), Mihai-Gabriel and Victor-Valeriu (2014), Liu et al. (2019), Santos et al. (2020), Gharvirian and Bohlooli (2017), Wang et al. (2020a), Chen and Yu (2016b), Dayal and Srivastava (2018), MohanaPriya and Shalinie (2017) and Gong et al. (2019)	23/16.1%	BPNN, ELM, MLP, RBF, RBM, and SOM have been employed as the detection algorithms. SOM is the most popular neural network when designing neural network-based DDoS detection mechanism.
		Classifying	Phan and Park (2019), Santos et al. (2020), Rahman et al. (2019), Bakker et al. (2018), Musumeci et al. (2020), Cui et al. (2019), Myint Oo et al. (2019), Ye et al. (2018), Yu et al. (2018), Hu et al. (2017), He et al. (2018), Yang and Zhao (2018), Latah and Toker (2018), Shang et al. (2017), Shen et al. (2020), Oo et al. (2017), Li et al. (2015), Kokila et al. (2014), Liu et al. (2017a), Chen et al. (2017), Mehr and Ramamurthy (2019), Mowla et al. (2018), Polat et al. (2020), Hyder and Lung (2018), Alshamrani et al. (2017) and Han et al. (2018)	27/18.9%	The number of classifying-based DDoS detection mechanisms are largest in machine learning-based DDoS detection mechanisms. SVM is the most popular algorithms used in DDoS detection.
		Clustering	Nam et al. (2018), Rahman et al. (2019), Bakker et al. (2018), Musumeci et al. (2020), Polat et al. (2020), Tan et al. (2020), Zhu et al. (2018) and Sun et al. (2018)	8/5.6%	DPMM, FCM, HHH, and KNN have been used for detecting DDoS attack. KNN-based DDoS detection mechanisms are the majority.
		Deep learning	Arivudainambi et al. (2019), Wang et al. (2020b), Haider et al. (2020), Li et al. (2018), Narayanadoss et al. (2019), SaiSindhuTheja and Shyam (2021), Tang et al. (2016) and Asad et al. (2019)	8/5.6%	CNN, RNN, LSTM, and DNN have been considered. High accuracy can be achieved by jointly using several algorithms.
		Ensemble learning	Santos et al. (2020), Rahman et al. (2019), Bakker et al. (2018), Musumeci et al. (2020) and Chen et al. (2018a)	6/4.2%	There are few ensemble learning-based DDoS detection mechanisms in SDN. It may obtain a quite high detection accuracy for DDoS attacks.
Statistics algorithm	9/6.3%	CUSUM	Wang et al. (2016a), Conti et al. (2017) and Mahrach et al. (2018)	3/2.1%	It has a low computational complexity. The detection performance relies on the threshold used to identify DDoS attack.
		CB-TRW and RL	Birkinshaw et al. (2019) and Özçelik et al. (2017)	2/1.4%	The computation complexity is quite low. A threshold is needed to be determined to detect DDoS attack.
		DWT	Zerbini et al. (2019)	1/0.7%	It achieves high accuracy, recall, F-measure, and AUC.
		SPRT	Dong et al. (2016)	1/0.7%	It requests a small number of observations to determine a DDoS attack. It can detect a DDoS attack with the pre-defined false positive error rate and false negative error rate.
		Markov model	Wang et al. (2018a)	1/0.7%	It utilizes the Renyi entropies of the destination and source IP addresses to train the Markov model to detect DDoS attacks.
		Self-designed	Kalkan et al. (2017)	1/0.7%	It provides an elegant detection accuracy especially for unknown DDoS attacks.
Combination of multiple methods	9/6.3%	Threshold-SVM	Yang and Zhao (2018), Latah and Toker (2018), Shang et al. (2017) and Shen et al. (2020)	4/2.8%	The main idea is roughly detecting DDoS by threshold-based method and exactly detecting DDoS by SVM-based method.
		Threshold-Fuzzy logic	Dang-Van and Truong-Thu (2017)	1/0.7%	A threshold-based elementary detection is firstly applied to detect suspect attack. The fuzzy logic-based detection further works to determine DDoS attacks.
		Threshold-BPNN-PSO	Liu et al. (2019)	1/0.7%	It pre-detects DDoS by applying threshold-based detection on switches and precisely detects DDoS by executing BPNN-PSO-based detection on the controller.
		Threshold-RBF-PSO	Dayal and Srivastava (2018)	1/0.7%	The entropy of destination IP address is calculated and compared with a threshold to preliminarily detect DDoS. Once a DDoS alert is generated by the threshold-based module, a RBF-PSO-based detection module is utilized to further detect DDoS attacks.

(*continued on next page*)

Table 15 (*continued*).

Technologies	Number/ Percentage	Subtypes	Literatures	Number/ Percentage	Description
		SVM-SOM	Phan and Park (2019) and Phan et al. (2016)	2/1.4%	The detection result are determined by combining the SVM and SOM classifiers, where a method is needed to make the final decision
		SOM-*k*NN	Nam et al. (2018)	1/0.7%	It can decrease the processing time, with a little decrease on the detection rate and false positive rate.
		entropy-C4.5	Sudar and Deepalakshmi (2020)	1/0.7%	It first compares the entropy of the source IP addresses with a threshold, and further detect DDoS using the C4.5 classifier.
		Kmeans-KNN	Tan et al. (2020)	1/0.7%	Kmeans is utilized to classify similar instances into different categories, while the KNN is employed to detect the measured instance.
Threshold	53/35%	Network status	Latah and Toker (2018), Hyder and Lung (2018), Chen et al. (2018b), Kalkan et al. (2018), Wang et al. (2018b), Zheng et al. (2018), Conti et al. (2019), Mousavi and St-Hilaire (2018), Kumar et al. (2018b), David and Thomas (2019), Guesmi and Saidane (2017), Gurusamy and MSK (2019), Lin et al. (2017), Xu et al. (2017), Sambandam et al. (2018), Liu et al. (2017b), Boite et al. (2017), Huong and Thanh (2017), Yang et al. (2017), You et al. (2017), Wang et al. (2017), Pandikumar et al. (2017), Tsai et al. (2017), Buragohain and Medhi (2016), Chen et al. (2016), Xing et al. (2016), Piedrahita et al. (2015), Van Trung et al. (2015), Wang et al. (2015a), Hommes et al. (2014), Duy and Pham (2018), Lu and Wang (2016), Murtuza and Asawa (2018), Jiang et al. (2016), Rebecchi et al. (2019), Wang et al. (2016b), Bhushan and Gupta (2018), Wang et al. (2018c), Sahoo et al. (2018a,b), Dehkordi and Soltanaghaei (2020) and Wu et al. (2020)	43/30%	It compares some features of network devices with a threshold to identify the DDoS attacks.
		Server status	Lukaseder et al. (2018), Hong et al. (2017), Lukaseder et al. (2017) and Shtern et al. (2014)	4/2.8%	The CPU utilization, disk time, waiting time, throughput, response time, and open HTTP connections have been considered as the symbol of DDoS attacks.
		Client status	Mohammadi et al. (2017), Gkountis et al. (2017), Dao et al. (2015) and Dao et al. (2016)	4/2.8%	It compares the number of illegitimate SYN packet, RST packets, average bytes generated by each client with a threshold to identify the DDoS attacks.
Other	17/11.9%	Distributed detection	Rathore et al. (2019), Ujjan et al. (2019), Shao et al. (2019), Shu et al. (2020) and Nguyen et al. (2019)	5/3.5%	It uses collaborative detection mechanisms to identify DDoS attacks.
		Catastrophe theory	Guo et al. (2019)	1/0.7%	It considers the changes of SDN network state as a catastrophe process.
		Bloom filter	Xiao et al. (2016)	1/0.7%	The DDoS attack is detected by checking whether the features of a flow belongs to a Bloom filter which is established to represent an anomaly DDoS attack.
		Third-party software	Chin et al. (2015) and Manso et al. (2019)	2/1.4%	Snort has been used as the detection module in this kind of method.
		Consistency	De Assis et al. (2017)	1/0.7%	The signatures of the IP flows are collected, calculated and compared with a pre-defined signature and the signatures of known DDoS attacks to approximately and exactly detect DDoS attacks.
		Similarity	Yin et al. (2018)	1/0.7%	The *packet-in* message rate is obtained and divided into two categories. The cosine similarity of these two categories are calculated to detect a DDoS attack.
		Fuzzy evaluation	Yan et al. (2016b) and Wang et al. (2018d)	2/1.4%	Some characteristics are employed as the factors of the fuzzy synthetic evaluation decision-making model to identify DDoS attacks.
		Probabilistic transition	Ivannikova et al. (2017)	1/0.7%	It calculates the joint probability of each session to detect DDoS attacks.
		Graph model	Aleroud and Alsmadi (2016) and Wang et al. (2015b)	2/1.4%	The probabilistic inference graphical model and a self-designed graph model have been utilized to detect DDoS attacks.
		Self designed	Wei et al. (2016)	1/0.7%	A self-designed if-else rules-based detection is used to detect DDoS attacks. It works on the idea of detecting DDoS attack by comparing the number of packets received by a port with the number of packets sent by the same port.

or extremely suit for all scenarios. For instance, the neural network-based DDoS detection mechanisms do not always perform better than the clustering-based mechanisms. Therefore, when designing the DDoS detection mechanism, the researchers can choose a specified algorithm, according to the specified scenario that includes the benign network traffic, network topology, possible DDoS attacks, and datasets.

10. Open problems and future directions

10.1. How to design feature selection methods for DDoS detection in SDN

Although feature selection has a powerful impact on the performance of DDoS detection, only a few mechanisms employ the feature selection as its main component. Most researchers do not consider the feature selection when designing the DDoS detection mechanisms. According to our survey, although many features can be extracted from SDN, the number of packets of flows, number of bytes of flows, duration of flows, symmetry/asymmetry of flows, entropy of source/destination IP address, and entropy of source/destination port are the most popular features utilized in DDoS detection in SDN, which are selected based on the experience of researchers. Therefore, whether there are some better feature subsets for detecting DDoS in SDN still needs to research. Furthermore, as the network status is changeable, the ability of the selected features to reflect network status may be also changeable. Although some works focus on the feature selection for DDoS detection (Phan et al., 2019; SaiSindhuTheja and Shyam, 2021), there is still a need for dynamically selecting and updating features to improve the effectiveness of DDoS detection mechanisms.

10.2. How to exploit the potential of SDN to detect DDoS

As a network architecture different with the traditional network, SDN presents new characteristics which never exist in the traditional network. For instance, in the SDN employing OpenFlow as its southbound interfaces, the OpenFlow messages such as the *packet-in*, *packet-out*, *flow-removed*, *flow-mod*, *echo-request*, *echo-reply*, *hello*, and *error* messages, may have the potential ability to reflect the network status. In other words, these messages may be used to detect DDoS attacks in SDN. However, at current, few research has been studied for this purpose. Hence, how to exploit the potential of SDN to detect DDoS maybe another open problem.

10.3. How to early detect DDoS in SDN

As a devastating network attack, DDoS can overwhelm the target in a short time. In order to prevent the target from being crushed, the DDoS detection mechanism must detect the potential DDoS attack as quick as possible. More seriously, most attention is paid to the detection accuracy when designing the DDoS detection mechanism. Although some researchers have taken an effort on the early detection of DDoS attacks such as Mousavi and St-Hilaire (2018) and Pandikumar et al. (2017), more researches about how to design DDoS detection mechanism to early detect DDoS attack need to be done.

10.4. How to integrate other networks or computing models with SDN to design DDoS detection methods

The integration of other networks such as IoT or computing models such as edge computing brings many possibilities for achieving efficient identification of DDoS attacks. For instance, in IoT, the DDoS detection system can be deployed on the IoT gateway device, which helps to collect fine-grained detection data and reduce the DDoS response time. Similarly, the edge computing can also improve the DDoS detection efficiency by detecting DDoS attack in the edge node closed to the DDoS attackers. However, when designing the DDoS detection mechanisms with the integration of SDN and other networks or computing models, the characters of these networks or computing models should be considered. For instance, the computational capacity of the IoT devices and edge nodes may hinder the deployment of high complexity detection methods. Hence, it is severe challenge for researchers to study the DDoS detection mechanisms with integration of SDN and other networks or computing models.

10.5. Lack of research about low-rate DDoS in SDN

In this work, we found that most of the current research is focused on detecting the high-rate DDoS attack. There is a small part of researches about detecting the low-rate DDoS attack. However, the low-rate DDoS attack is also one of the main DDoS attacks which can cause fatal damage. Meanwhile, as the fact that the low-rate DDoS attack will send a small amount of packets compared with the high-rate DDoS attack, the features which can obviously reflect the high-rate DDoS attack will become invalid for detecting the low-rate DDoS attack. Therefore, it is difficult to detect the low-rate DDoS attack. Some detection methods against low-rate DDoS attack have been presented, such as Phan et al. (2019) and Sahoo et al. (2018b). However, how to exploit the features of the low-rate DDoS attack in SDN and design the detection mechanisms with high true positive rate and low false positive rate is still in urgent need.

10.6. Lack of research about new type of DDoS in SDN

Except the traditional DDoS attacks, nowadays, more and more new types of DDoS attack come out. Compared with the traditional DDoS attack, these new types of DDoS attack use different attack pattern. For instance, the link flooding attack that aims to overwhelm the network links instead of flooding servers, and the Crossfire attack that attacks the servers around the target server instead of the target server. These DDoS attacks present different forms in the network, which means that the traditional DDoS detection mechanisms may fail to detect these DDoS attacks. There are only a few of DDoS attack detection mechanisms related to the new types of DDoS attack. Hence, how to detect the new types of DDoS attack is still need to be further studied.

11. Conclusion

In this work, we provide a study of DDoS attacks in SDN, which are classified into two kinds: (1) DDoS targeted at the SDN network and (2) DDoS aimed at the service providers. For the DDoS targeted at the SDN network, it is further categorized into the DDoS against the infrastructure layer, DDoS threatened the control layer, and DDoS aimed at the application layer. We give a clear discussion about how the attackers utilize the vulnerabilities existed in different layers of SDN to launch a DDoS attack. For the second kind of DDoS attack, we also further classified it into the bandwidth depletion DDoS attack and the service provider resource depletion DDoS attack.

After that, we provide a comprehensive survey of DDoS detection mechanisms in SDN. The existing DDoS detection mechanisms are categorized into five types including the machine learning-based, statistics-based, combination of multiple methods-based, threshold-based, and other method-based DDoS detection mechanisms. For each kind of DDoS detection mechanism, we further classified it into subtypes. For instance, the machine learning-based DDoS detection mechanism is further categorized into the neural network-based, classifying-based, clustering-based, deep learning-based, and ensemble learning-based DDoS detection mechanisms.

The separate analysis of each subtype and an integral analysis of all DDoS detection mechanisms are discussed. We analyze the common process, advantages, disadvantages for each subtype. We also analyzed and compared the current DDoS detection mechanisms. A conclusion drawn from the analysis is that the machine learning-based DDoS detection and threshold-based DDoS detection mechanisms are the two most

popular technologies utilized to detect DDoS attacks in SDN. For the machine learning-based DDoS detection mechanisms, the classifying-based and neural network-based DDoS detection mechanisms have attracted a lot of interest from the researchers.

Moreover, The open problems and future directions are also discussed, such as how to design feature selection methods for DDoS detection in SDN, how to exploit the potential of SDN to detect DDoS, how to early detect DDoS in SDN, how to integrate other networks or computing models with SDN to design DDoS detection methods, lack of research about low-rate DDoS in SDN, and lack of research about the new type of DDoS in SDN.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Key Research and Development Program of China under Grant No. 2019YFB1803500, National Natural Science Foundation of China (NSFC) under the Grant No. 61902085 and 61802081, Guizhou Provincial Science and Technology Plan under the Grant No. [2020]1Y267, and Scientific Research Foundation for Introduced Talents of Guizhou University, China under the Grant No. (2019)52.

References

Agrawal, N., Tapaswi, S., 2021. An SDN-assisted defense mechduanism for the shrew DDoS attack in a cloud computing environment. J. Netw. Syst. Manage. 29 (2), 1–28.

Ahmed, M.E., Kim, H., Park, M., 2017. Mitigating DNS query-based DDoS attacks with machine learning on software-defined networking. In: MILCOM 2017-2017 IEEE Military Communications Conference. MILCOM, IEEE, pp. 11–16.

Akyildiz, Ian F., Lin, S., Wang, P., 2015. Wireless software-defined networks (W-SDNs) and network function virtualization (NFV) for 5G cellular systems: An overview and qualitative evaluation. Comput. Netw. 93, 66–79.

Al-Adaileh, M.A., Anbar, M., Hasbullah, I.H., et al., 2020. Detection techniques of distributed denial of service attacks on software-defined networking controller–A review. IEEE Access 8, 143985-143995.

Aleroud, A., Alsmadi, I., 2016. Identifying DoS attacks on software defined networks: a relation context approach. In: NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium. IEEE, pp. 853–857.

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

Anbalagan, S., Kumar, D., Faustina J, M., Raja, G., Ejaz, W., Bashir, A.K., 2020. SDN-assisted efficient LTE-WiFi aggregation in next generation IoT networks. Future Gener. Comput. Syst. 107, 898–908.

Anon, 2009. Open Networking Foundation: 'OpenFlow switch specification 1.0.0'. https://www.opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.0.0.pdf.

Anon, 2011a. Open Networking Foundation: 'OpenFlow switch specification 1.1.0'. https://3vf60mmveq1g8vzn48q2o71a-wpengine.netdna-ssl.com/wp-content/uploads/2014/10/openflow-spec-v1.1.0.pdf.

Anon, 2011b. Open Networking Foundation: 'OpenFlow switch specification 1.2'. https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-spec-v1.2.pdf.

Anon, 2012a. Software-defined networking: The new norm for networks, https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf.

Anon, 2012b. Open Networking Foundation: 'OpenFlow switch specification 1.3.0'. http://www.cs.yale.edu/homes/yu-minlan/teach/csci599-fall12/papers/openflow-spec-v1.3.0.pdf.

Anon, 2013. Open Networking Foundation: 'OpenFlow switch specification 1.4.0'. https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-spec-v1.4.0.pdf.

Anon, 2014. Open Networking Foundation: 'OpenFlow Switch specification 1.5.0'. https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.0.noipr.pdf.

Anon, 2021a. POX controller manual current documentation. [Online]. Available: https://noxrepo.github.io/pox-doc/html/.

Anon, 2021b. OpenDaylight: A linux foundation collaborative project. [Online]. Available: https://www.opendaylight.org/.

Anon, 2021c. Ryu SDN framework community, Ryu controller. [Online]. Available: https://osrg.github.io/ryu/index.html.

Anon, 2021d. Big switch networks, "project floodlight". [Online]. Available: http://www.projectfloodlight.org/floodlight/.

Anon, 2021e. OpenMUL SDN platform. [Online]. Available: http://www.openmul.org/openmul-controller.html.

Anon, 2021f. OpenContrail An open-source network virtualization platform for the cloud. [Online]. Available: http://www.opencontrail.org/.

Arivudainambi, D., V K, K.A., Chakkaravarthy, S.S., 2019. LION IDS: A meta-heuristics approach to detect DDoS attacks against Software-Defined Networks. Neural Comput. Appl. 31 (5), 1491–1501.

Asad, M., Asim, M., Javed, T., et al., 2019. DeepDetect: Detection of distributed denial of service attacks using deep learning. Comput. J. bxz064.

Ashraf, J., Latif, S., 2014. Handling intrusion and DDoS attacks in Software Defined Networks using machine learning techniques. In: 2014 National Software Engineering Conference. IEEE.

Bakker, J.N., Bryan, Ng., Winston, K.G.S., 2018. Can machine learning techniques be effectively used in real networks against DDoS attacks? In: 2018 27th International Conference on Computer Communication and Networks. ICCCN, IEEE.

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

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

Berde, P., Gerola, M., Hart, J., et al., 2014. Onos: Towards an open, distributed sdn os. In: Proceedings of the Third Workshop on Hot Topics in Software Defined Networking. HotSDN 2014, ACM, pp. 1–6.

Bhardwaj, A., Subrahmanyam, G.V.B., Avasthi, V., et al., 2016. DDoS attacks, new DDoS taxonomy and mitigation solutions—a survey. In: 2016 International Conference on Signal Processing, Communication, Power and Embedded System. SCOPES, IEEE, pp. 793–798.

Bhushan, K., Gupta, B.B., 2018. Detecting DDoS attack using software defined network (SDN) in cloud computing environment. In: 2018 5th International Conference on Signal Processing and Integrated Networks. SPIN, IEEE, pp. 872–877.

Birkinshaw, C., Rouka, E., Vassilakis, V.G., 2019. Implementing an intrusion detection and prevention system using software-defined networking: Defending against port-scanning and denial-of-service attacks. J. Netw. Comput. Appl. 136, 71–85.

Boite, J., Nardin, P.A., Rebecchi, F., et al., 2017. Statesec: Stateful monitoring for DDoS protection in software defined networks. In: 2017 IEEE Conference on Network Softwarization. NetSoft, pp. 1–9.

Bouyeddou, B., Harrou, F., Sun, Y., et al., 2018. Detection of smurf flooding attacks using Kullback-Leibler-based scheme. In: 2018 4th International Conference on Computer and Technology Applications. ICCTA, IEEE, pp. 11–15.

Braga, R., de Souza Mota, E., Passito, A., 2010. Lightweight DDoS flooding attack detection using NOX/OpenFlow. In: The 35th Annual IEEE Conference on Local Computer Networks. IEEE, pp. 408–415.

Buragohain, C., Medhi, N., 2016. FlowTrApp: An SDN based architecture for DDoS attack detection and mitigation in data centers. In: 2016 3rd International Conference on Signal Processing and Integrated Networks. SPIN, IEEE, pp. 519–524.

Caprolu, M., Raponi, S., Di Pietro, R., 2019. FORTRESS: An efficient and distributed firewall for stateful data plane SDN. Secur. Commun. Netw. 6874592.

Casado, M., Garfinkel, T., Akella, A., et al., 2006. SANE: A protection architecture for enterprise networks. In: Proceedings of the 15th conference on USENIX Security Symposium.

Casey, C.J., Sutton, A., Sprintson, A., 2014. tinyNBI: Distilling an API from essential OpenFlow abstractions. In: Proceedings of the Third Workshop on Hot Topics in Software Defined Networking. ACM, pp. 37–42.

Chauhan, V., Saini, P., 2015. ICMP flood attacks: A vulnerability analysis. In: Cyber Security: Proceedings of CSI 2015. Springer Singapore, pp. 261–268.

Chen, C.C., Chen, Y.R., Lu, W.C., et al., 2017. Detecting amplification attacks with software defined networking. In: 2017 IEEE Conference on Dependable and Secure Computing. IEEE, pp. 195–201.

Chen, M.H., Ciou, J.Y., Chung, I., et al., 2018b. FlexProtect: A SDN-based DDoS attack protection architecture for multi-tenant data centers. In: Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region. ACM, pp. 202–209.

Chen, W.H., Hsu, S.H., Shen, H.P., 2005. Application of SVM and ANN for intrusion detection. Comput. Oper. Res. 32 (10), 2617–2634.

Chen, Z., Jiang, F., Cheng, Y., et al., 2018a. XGBoost classifier for DDoS attack detection and analysis in SDN-Based cloud. In: 2018 IEEE International Conference on Big Data and Smart Computing. BigComp, IEEE, pp. 251–256.

Chen, K., Junuthula, A.R., Siddhrau, I.K., et al., 2016. SDNShield: Towards more comprehensive defense against DDoS attacks on SDN control plane. In: 2016 IEEE Conference on Communications and Network Security. CNS, IEEE, pp. 28–36.

Chen, X., Yu, S., 2016a. CIPA: A collaborative intrusion prevention architecture for programmable network and SDN. Comput. Secur. 58, 1–19.
Chen, X., Yu, S., 2016b. A collaborative intrusion detection system against DDoS for SDN. IEICE Trans. Inf. Syst. 99 (9), 2395–2399.
Chin, T., Mountrouidou, X., Li, X., et al., 2015. An SDN-supported collaborative approach for DDoS flooding detection and containment. In: MILCOM 2015–2015 IEEE Military Communications Conference. IEEE, pp. 659–664.
Cho, H., Kang, S., Lee, Y., 2015. Centralized ARP proxy server over SDN controller to cut down ARP broadcast in large-scale data center networks. In: 2015 International Conference on Information Networking. ICOIN, IEEE, pp. 301–306.
Conti, M., Gangwal, A., Gaur, M.S., 2017. A comprehensive and effective mechanism for DDoS detection in SDN. In: 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications. WiMob, IEEE, pp. 1–8.
Conti, M., Lal, C., Mohammadi, R., et al., 2019. Lightweight solutions to counter DDoS attacks in software defined networking. Wirel. Netw. 25 (5), 2751–2768.
Cui, J., He, J., Xu, Y., et al., 2018. TDDAD: Time-based detection and defense scheme against DDoS attack on SDN controller. In: Australasian Conference on Information Security and Privacy. Springer, Cham, pp. 649–665.
Cui, J., Wang, M., Luo, Y., et al., 2019. DDoS detection and defense mechanism based on cognitive-inspired computing in SDN. Future Gener. Comput. Syst. 97, 275–283.
Cui, Y., Yan, L., Li, S., et al., 2016. SD-Anti-DDoS: Fast and efficient DDoS defense in software-defined networks. J. Netw. Comput. Appl. 68, 65–79.
Dang-Van, T., Truong-Thu, H., 2017. A multi-criteria based software defined networking system Architecture for DDoS-attack mitigation. REV J. Electron. Commun. 6 (3–4).
Dao, N.N., Kim, J., Park, M., et al., 2016. Adaptive suspicious prevention for defending DoS attacks in SDN-based convergent networks. PLoS One 11 (8), e0160375.
Dao, N.N., Park, J., Park, M., et al., 2015. A feasible method to combat against DDoS attack in SDN network. In: 2015 International Conference on Information Networking. ICOIN, IEEE, pp. 309–311.
David, J., Thomas, C., 2019. Efficient DDoS flood attack detection using dynamic thresholding on flow-based network traffic. Comput. Secur. 82, 284–295.
Dayal, N., Maity, P., Srivastava, S., et al., 2016. Research trends in security and DDoS in SDN. Secur. Commun. Netw. 9 (18), 6386–6411.
Dayal, N., Srivastava, S., 2018. An RBF-PSO based approach for early detection of DDoS attacks in SDN. In: 2018 10th International Conference on Communication Systems & Networks. COMSNETS, IEEE, pp. 17–24.
De Assis, M.V.O., Hamamoto, A.H., Abrão, T., et al., 2017. A game theoretical based system using holt-winters and genetic algorithm with fuzzy logic for DoS/DDoS mitigation on SDN networks. IEEE Access 5, 9485–9496.
De Donno, M., Giaretta, A., Dragoni, N., et al., 2017. A taxonomy of distributed denial of service attacks. In: 2017 International Conference on Information Society. i-Society, IEEE, pp. 100–107.
Dehkordi, A.B., Soltanaghaei, M., 2020. A novel distributed denial of service (DDoS) detection method in software defined networks. IEEE Trans. Ind. Appl. http://dx.doi.org/10.1109/TIA.2020.3001535.
DeLooze, L.L., 2006. Attack characterization and intrusion detection using an ensemble of self-organizing maps. In: The 2006 IEEE International Joint Conference on Neural Network Proceedings. IEEE, pp. 2121–2128.
Depren, O., Topallar, M., Anarim, E., et al., 2005. An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. Expert Syst. Appl. 29 (4), 713–722.
Deshmukh, R.V., Devadkar, K.K., 2015. Understanding DDoS attack & its effect in cloud environment. Procedia Comput. Sci. 49, 202–210.
Detti, A., Pisa, C., Salsano, S., et al., 2013. Wireless mesh software defined networks (wmSDN). In: 2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications. WiMob 2013, IEEE, pp. 89–95.
Dong, P., Du, X., Zhang, H., et al., 2016. A detection method for a novel DDoS attack against SDN controllers by vast new low-traffic flows. In: 2016 IEEE International Conference on Communications. ICC, IEEE, pp. 1–6.
Dong, S., Jain, R., Abbas, K., 2019. A survey on distributed denial of service (DDoS) attacks in SDN and cloud computing environments. IEEE Access 7, 80813–80828.
Duan, X., Wang, X., 2015. Authentication handover and privacy protection in 5G hetnets using software-defined networking. IEEE Commun. Mag. 53 (4), 28–35.
Duy, P.T., Pham, V.H., 2018. A role-based statistical mechanism for DDoS attack detection in SDN. In: 2018 5th NAFOSTED Conference on Information and Computer Science. NICS, IEEE, pp. 177–182.
Erickson, D., 2013. The beacon openflow controller. In: Proceedings of the Second ACM SIGCOMM Workshop. ACM, pp. 13–18.
Gao, D., Liu, Z., Liu, Y., et al., 2018. Defending against Packet-In messages flooding attack under SDN context. Soft Comput. 22 (20), 6797–6809.
Gharvirian, F., Bohlooli, A., 2017. Neural network based protection of software defined network controller against distributed denial of service attacks. Int. J. Eng. Trans. B: Appl. 30 (11), 1714–1722.
Gkountis, C., Taha, M., Lloret, J., et al., 2017. Lightweight algorithm for protecting SDN controller against DDoS attacks. In: 2017 10th IFIP Wireless and Mobile Networking Conference. WMNC, IEEE, pp. 1–6.
Gong, C., Yu, D., Zhao, L., et al., 2019. An intelligent trust model for hybrid DDoS detection in software defined networks. Concurr. Comput.: Pract. Exp. e5264.
Gude, N., Koponen, T., Pettit, J., et al., 2008. Nox: Towards an operating system for networks. ACM SIGCOMM Comput. Commun. Rev. 38 (3), 105.
Guesmi, H., Saidane, L.A., 2017. Using sdn approach to secure cloud servers against flooding based ddos attacks. In: 2017 25th International Conference on Systems Engineering. ICSEng, IEEE, pp. 309–315.
Guo, Y., Miao, F., Zhang, L., et al., 2019. CATH: an effective method for detecting denial-of-service attacks in software defined networks. Sci. China Inf. Sci. 62 (3), 32106.
Gurusamy, U.M., MSK, M., 2019. Detection and mitigation of UDP flooding attack in a multicontroller software defined network using secure flow management model. Concurr. Comput.: Pract. Exp. e5326.
Haider, S., Akhunzada, A., Mustafa, I., et al., 2020. A deep CNN ensemble framework for efficient DDoS attack detection in software defined networks. IEEE Access 8, 53972–53983.
Han, B., Yang, X., Sun, Z., et al., 2018. OverWatch: A cross-plane DDoS attack defense framework with collaborative intelligence in SDN. Secur. Commun. Netw. 9649643.
He, B., Zou, F., Wu, Y., 2018. Multi-SDN based cooperation scheme for DDoS attack defense. In: 2018 Third International Conference on Security of Smart Cities, Industrial Control System and Communications. SSIC, IEEE.
Hommes, S., State, R., Engel, T., 2014. Implications and detection of dos attacks in openflow-based networks. In: 2014 IEEE Global Communications Conference. IEEE, pp. 537–543.
Hong, K., Kim, Y., Choi, H., et al., 2017. SDN-assisted slow HTTP DDoS attack defense method. IEEE Commun. Lett. 22 (4), 688–691.
Hu, D., Hong, P., Chen, Y., 2017. FADM: DDoS flooding attack detection and mitigation system in software-defined networking. In: GLOBECOM 2017–2017 IEEE Global Communications Conference. IEEE.
Huong, T.T., Thanh, N.H., 2017. Software defined networking-based one-packet DDoS mitigation architecture. In: Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication. ACM, p. 110.
Hyder, H.K., Lung, C.H., 2018. Closed-loop DDoS mitigation system in software defined networks. In: 2018 IEEE Conference on Dependable and Secure Computing. DSC, IEEE.
Ivannikova, E., Zolotukhin, M., Hämäläinen, T., 2017. Probabilistic transition-based approach for detecting application-layer ddos attacks in encrypted software-defined networks. In: International Conference on Network and System Security. Springer, Cham, pp. 531–543.
Jafarian, J.H., Al-Shaer, E., Duan, Q., 2012. Openflow random host mutation: transparent moving target defense using software defined networking. In: Proceedings of the First Workshop on Hot Topics in Software Defined Networks. ACM, pp. 127–132.
Jain, S., Kumar, A., Mandal, S., et al., 2013. B4: Experience with a globally-deployed software defined WAN. ACM SIGCOMM Comput. Commun. Rev. 43 (4), 3–14.
Jiang, D., Yang, Y., Xia, M., 2009. Research on intrusion detection based on an improved SOM neural network. In: 2009 Fifth International Conference on Information Assurance and Security, Vol. 1. IEEE, pp. 400–403.
Jiang, Y., Zhang, X., Zhou, Q., et al., 2016. An entropy-based DDoS defense mechanism in software defined networks. In: International Conference on Communicatins and Networking in China. Springer, Cham, pp. 169–178.
Kalkan, K., Altay, L., Gür, G., et al., 2018. JESS: Joint entropy-based DDoS defense scheme in SDN. IEEE J. Sel. Areas Commun. 36 (10), 2358–2372.
Kalkan, K., Gür, G., Alagöz, F., 2017. SDNScore: A statistical defense mechanism against DDoS attacks in SDN environment. In: 2017 IEEE Symposium on Computers and Communications. ISCC, IEEE, pp. 669–675.
Kalliola, A., Lee, K., Lee, H., et al., 2015. Flooding DDoS mitigation and traffic management with software defined networking. In: 2015 IEEE 4th International Conference on Cloud Networking. CloudNet, IEEE, pp. 248–254.
Kayacik, H.G., Zincir-Heywood, A.N., Heywood, M.I., 2007. A hierarchical SOM-based intrusion detection system. Eng. Appl. Artif. Intell. 20 (4), 439–451.
Kim, D.S., Park, J.S., 2003. Network-based intrusion detection with support vector machines. In: International Conference on Information Networking. Springer, Berlin, Heidelberg, pp. 747–756.
Kiziloren, T., Germen, E., 2007. Network traffic classification with self organizing maps. In: 2007 22nd International Symposium on Computer and Information Sciences. IEEE, pp. 1–5.
Kokila, R.T., Selvi, S.T., Govindarajan, K., 2014. DDoS detection and analysis in SDN-based environment using support vector machine classifier. In: 2014 Sixth International Conference on Advanced Computing. ICoAC, IEEE, pp. 205–210.
Kolahi, S.S., Treseangrat, K., Sarrafpour, B., 2015. Analysis of UDP DDoS flood cyber attack and defense mechanisms on web server with Linux Ubuntu 13. In: 2015 International Conference on Communications, Signal Processing, and their Applications. ICCSPA'15, IEEE, pp. 1–5.
Kubra, K., Gurkan, G., Alagoz, F., 2017. Defense mechanisms against DDoS attacks in SDN environment. IEEE Commun. Mag. 55 (9), 175–179.
Kumar, P., Tripathi, M., Nehra, A., et al., 2018a. SAFETY: Early detection and mitigation of TCP SYN flood utilizing entropy in SDN. IEEE Trans. Netw. Serv. Manag. 15 (4), 1545–1559.

Kumar, P., Tripathi, M., Nehra, A., et al., 2018b. SAFETY: Early detection and mitigation of TCP SYN flood utilizing entropy in SDN. IEEE Trans. Netw. Serv. Manag. 15 (4), 1545–1559.

Latah, M., Toker, L., 2018. A novel intelligent approach for detecting DoS flooding attacks in software-defined networks. Int. J. Adv. Intell. Inform. 4 (1), 11–20.

Leng, B., Huang, L., Qiao, C., et al., 2017. FTRS: A mechanism for reducing flow table entries in software defined networks. Comput. Netw. 122, 1–15.

Li, C., Wu, Y., Yuan, X., et al., 2018. Detection and defense of DDoS attack–based on deep learning in OpenFlow-based SDN. Int. J. Commun. Syst. 31 (5), e3497.

Li, X., Yuan, D., Hu, H., et al., 2015. DDoS detection in SDN switches using support vector machine classifier. In: 2015 Joint International Mechanical, Electronic and Information Technology Conference. JIMET-15, Atlantis Press.

Liang, Haochi, et al., 2015. Effective idle_timeout value for instant messaging in software defined networks. In: 2015 IEEE International Conference on Communication Workshop. ICCW, IEEE.

Lin, P.C., Hsu, Y.T., Hwang, R.H., 2017. Detecting and preventing DDoS attacks in SDN-based data center networks. In: International Conference on Cloud Computing and Security. Springer, Cham, pp. 50–61.

Liu, Z., He, Y., Wang, W., et al., 2019. DDoS attack detection scheme based on entropy and PSO-BP neural network in SDN. China Commun. 16 (7), 144–155.

Liu, J., Lai, Y., Zhang, S., 2017a. FL-GUARD: A detection and defense system for DDoS attack in SDN. In: Proceedings of the 2017 International Conference on Cryptography, Security and Privacy. ACM, pp. 107–111.

Liu, Z., Xu, M., Cao, J., et al., 2017b. TSA: A two-phase scheme against amplification DDoS attack in SDN. In: International Conference on Mobile Ad-Hoc and Sensor Networks. Springer, Singapore, pp. 483–496.

Liu, G., Yi, Z., 2006. Intrusion detection using PCASOM neural networks. In: International Symposium on Neural Networks. Springer, Berlin, Heidelberg, pp. 240–245.

Lu, Y., Wang, M., 2016. An easy defense mechanism against botnet-based DDoS flooding attack originated in SDN environment using sFlow. In: Proceedings of the 11th International Conference on Future Internet Technologies. ACM, 14-20.

Lukaseder, T., Hunt, A., Stehle, C., et al., 2017. An extensible host-agnostic framework for SDN-assisted DDoS-mitigation. In: 2017 IEEE 42nd Conference on Local Computer Networks. LCN, IEEE, pp. 619–622.

Lukaseder, T., Maile, L., Erb, B., et al., 2018. SDN-assisted network-based mitigation of slow DDoS attacks. In: International Conference on Security and Privacy in Communication Systems. Springer, Cham, pp. 102–121.

Mahmud, A., Rahmani, R., 2011. Exploitation of OpenFlow in wireless sensor networks. In: Proceedings of 2011 International Conference on Computer Science and Network Technology, Vol. 1. IEEE, pp. 594–600.

Mahrach, S., El Mir, I., Haqiq, A., et al., 2018. SDN-based SYN flooding defense in cloud. J. Inf. Assur. Secur. 13 (1).

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

Matias, J., Garay, J., Mendiola, A., et al., 2014. FlowNAC: Flow-based network access control. Third European workshop on software defined networks. In: Proceedings of the Third European Workshop on Software Defined Networks. IEEE, pp. 79–84.

McKeown, N., Anderson, T., Balakrishnan, H., et al., 2008. OpenFlow: enabling innovation in campus networks. ACM SIGCOMM Comput. Commun. Rev. 38 (2), 69–74.

Mehr, S.Y., Ramamurthy, B., 2019. An SVM based DDoS attack detection method for Ryu SDN controller. In: Proceedings of the 15th International Conference on Emerging Networking EXperiments and Technologies. pp. 72–73.

Mihai-Gabriel, I., Victor-Valeriu, P., 2014. Achieving DDoS resiliency in a software defined network by intelligent risk assessment based on neural networks and danger theory. In: 2014 IEEE 15th International Symposium on Computational Intelligence and Informatic. CINTI, IEEE, pp. 319–324.

Mishra, A., Gupta, N., Gupta, B.B., 2021. Defense mechanisms against DDoS attack based on entropy in SDN-cloud using POX controller. Telecommun. Syst. http://dx.doi.org/10.1007/s11235-020-00747-w.

Mohammadi, R., Javidan, R., Conti, M., 2017. Slicots: An sdn-based lightweight countermeasure for tcp syn flooding attacks. IEEE Trans. Netw. Serv. Manag. 14 (2), 487–497.

MohanaPriya, P., Shalinie, S.M., 2017. Restricted Boltzmann machine based detection system for DDoS attack in software defined networks. In: 2017 Fourth International Conference on Signal Processing, Communication and Networking. ICSCN, IEEE, http://dx.doi.org/10.1109/ICSCN.2017.8085731.

Mousavi, S.M., St-Hilaire, M., 2018. Early detection of ddos attacks against software defined network controllers. J. Netw. Syst. Manage. 26 (3), 573–591.

Mowla, N.I., Doh, I., Chae, K., 2018. CSDSM: Cognitive switch-based DDoS sensing and mitigation in SDN-driven CDNi word. Comput. Sci. Inf. Syst. 15 (1), 163–185.

Mukkamala, S., Janoski, G., Sung, A., 2002. Intrusion detection using neural networks and support vector machines. In: Proceedings of the 2002 International Joint Conference on Neural Networks, Vol. 2. IJCNN'02 (Cat. No. 02CH37290), IEEE, pp. 1702–1707.

Mumpela, J.M., Young-Hoon, P., 2018. Strategies for detecting and mitigating DDoS attacks in SDN: A survey. J. Intell. Fuzzy Systems 35, 1–13.

Murtuza, S., Asawa, K., 2018. Mitigation and detection of DDoS attacks in software defined networks. In: 2018 Eleventh International Conference on Contemporary Computing. IC3, IEEE, pp. 1–3.

Musumeci, F., Ionata, Valentina, Paolucci, Francesco, Cugini, Filippo, Tornatore, Massimo, 2020. Machine-learning-assisted DDoS attack detection with P4 language. In: 2020 IEEE International Conference on Communications. ICC.

Myint Oo, M., Kamolphiwong, S., Kamolphiwong, T., et al., 2019. Advanced support vector machine-(ASVM-) based detection for distributed denial of service (DDoS) attack on software defined networking (SDN). J. Comput. Netw. Commun. 8012568.

Nam, T.M., Phong, P.H., Khoa, T.D., et al., 2018. Self-organizing map-based approaches in DDoS flooding detection using SDN. In: 2018 International Conference on Information Networking. ICOIN, IEEE, pp. 249–254.

Narayanadoss, A.R., Truong-Huu, T., Mohan, P.M., et al., 2019. Crossfire attack detection using deep learning in software defined its networks. In: 2019 IEEE 89th Vehicular Technology Conference. VTC2019-Spring, IEEE, pp. 1–6.

Nguyen, T.G., Phan, T.V., Nguyen, B.T., et al., 2019. SeArch: A collaborative and intelligent nids architecture for sdn-based cloud iot networks. IEEE Access 7, 107678-107694.

ONF, 2021. https://www.opennetworking.org/.

Oo, M.M., Kamolphiwong, S., Kamolphiwong, T., 2017. The design of SDN based detection for distributed denial of service (DDoS) attack. In: 2017 21st International Computer Science and Engineering Conference. ICSEC, IEEE.

Özçelik, M., Chalabianloo, N., Gür, G., 2017. Software-defined edge defense against IoT-based DDoS. In: 2017 IEEE International Conference on Computer and Information Technology. CIT, IEEE, pp. 308–313.

Pandikumar, T., Atkilt, F., Hassen, C.A., 2017. Early detection of DDoS attacks in a multi-controller based SDN. Internat. J. Engrg. Sci. 13422.

Peddabachigari, S., Abraham, A., Grosan, C., et al., 2007. Modeling intrusion detection system using hybrid intelligent systems. J. Netw. Comput. Appl. 30 (1), 114–132.

Phan, T.V., Bao, N.K., Park, M., 2016. A novel hybrid flow-based handler with DDoS attacks in software-defined networking. In: 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress. UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld, IEEE, pp. 350–357.

Phan, T.V., Bao, N.K., Park, M., 2017. Distributed-SOM: A novel performance bottleneck handler for large-sized software-defined networks under flooding attacks. J. Netw. Comput. Appl. 91, 14–25.

Phan, T.V., Gias, T.M.R., Islam, S.T., et al., 2019. Q-MIND: Defeating stealthy DoS attacks in SDN with a machine-learning based defense framework. In: 2019 IEEE Global Communications Conference. GLOBECOM, IEEE, pp. 1–6. http://dx.doi.org/10.1109/GLOBECOM38437.2019.9013585.

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

Piedrahita, A.F.M., Rueda, S., Mattos, D.M.F., et al., 2015. FlowFence: a denial of service defense system for software defined networking. In: 2015 Global Information Infrastructure and Networking Symposium. GIIS, IEEE, pp. 1–6.

Pillutla, H., Arjunan, A., 2019. Fuzzy self organizing maps-based DDoS mitigation mechanism for software defined networking in cloud computing. J. Ambient Intell. Humaniz. Comput. 10 (4), 1547–1559.

Polat, H., Polat, O., Cetin, A., 2020. Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models. Sustainability 12.

Powers, S.T., He, J., 2008. A hybrid artificial immune system and Self Organising Map for network intrusion detection. Inform. Sci. 178 (15), 3024–3042.

Rahman, O., Quraishi, M.A.R., Lung, C.H., 2019. DDoS attacks detection and mitigation in SDN using machine learning. In: 2019 IEEE World Congress on Services. SERVICES, IEEE.

Rahouti, M., Xiong, K., Ghani, N., et al., 2021. SYNGuard: Dynamic threshold-based SYN flood attack detection and mitigation in software-defined networks. IET Netw. 10 (2), 76–87.

Rathore, S., Kwon, B.W., Park, J.H., 2019. BlockSecIoTNet: Blockchain-based decentralized security architecture for IoT network. J. Netw. Comput. Appl. 143, 167–177.

Rebecchi, F., Boite, J., Nardin, P.A., et al., 2019. DDoS protection with stateful software-defined networking. Int. J. Netw. Manage. 29 (1), e2042.

Rotsos, C., Sarrar, N., Uhlig, S., et al., 2012. OFLOPS: An open framework for OpenFlow switch evaluation. In: International Conference on Passive and Active Network Measurement. Springer, pp. 85–95.

Sahoo, K.S., Panda, S.K., Sahoo, S., et al., 2019. Toward secure software-defined networks against distributed denial of service attack. J. Supercomput. 75, 4829–4874.

Sahoo, K.S., Puthal, D., Tiwary, M., et al., 2018b. An early detection of low rate DDoS attack to SDN based data center networks using information distance metrics. Future Gener. Comput. Syst. 89, 685–697.

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

SaiSindhuTheja, R., Shyam, G.K., 2021. An efficient metaheuristic algorithm based feature selection and recurrent neural network for DoS attack detection in cloud computing environment. Appl. Soft Comput. 100, 106997.

Sambandam, N., Hussein, M., Siddiqi, N., et al., 2018. Network security for IoT using SDN: timely DDoS detection. In: 2018 IEEE Conference on Dependable and Secure Computing. DSC, IEEE, pp. 1–2.

Santos, R., Souza, D., Santo, W., et al., 2020. Machine learning algorithms to detect DDoS attacks in SDN. Concurr. Comput.: Pract. Exper. e5402.

Shakil, M., Fuad Yousif Mohammed, A., Arul, R., et al., 2019. A novel dynamic framework to detect DDoS in SDN using metaheuristic clustering. Trans. Emerg. Telecommun. Technol. e3622.

Shang, G., Zhe, P., Bin, X., et al., 2017. FloodDefender: Protecting data and control plane resources under SDN-aimed DoS attacks. In: IEEE INFOCOM 2017-IEEE Conference on Computer Communications. IEEE, pp. 1–9.

Shao, Z., Zhu, X., Chikuvanyanga, A.M.M., et al., 2019. Blockchain-based sdn security guaranteeing algorithm and analysis model. In: International Conference on Wireless and Satellite Systems. Springer, Cham, pp. 348–362.

Shen, Yi, Wu, Chunming, Kong, Dezhang, Yang, Mingliang, 2020. TPDD: A two-phase DDoS detection system in software-defined networking. In: 2020 IEEE International Conference on Communications. ICC, IEEE, pp. 1–6.

Sherwood, R., Gibb, G., Yap, K.-k., et al., 2009. FlowVisor: A Network Virtualization Layer. OpenFlow Switch Consortium Technology Report, p. 132.

Shon, T., Moon, J., 2007. A hybrid machine learning approach to network anomaly detection. Inform. Sci. 177 (18), 3799–3821.

Shtern, M., Sandel, R., Litoiu, M., et al., 2014. Towards mitigation of low and slow application ddos attacks. In: 2014 IEEE International Conference on Cloud Engineering. IEEE, pp. 604–609.

Shu, J., Zhou, L., Zhang, W., et al., 2020. Collaborative intrusion detection for VANETs: A deep learning-based distributed SDN approach. IEEE Trans. Intell. Transp. Syst. http://dx.doi.org/10.1109/TITS.2020.3027390.

Singh, M.P., Bhandari, A., 2020. New-flow based DDoS attacks in SDN: Taxonomy, rationales, and research challenges. Comput. Commun. 154, 509–527.

Somani, G., Gaur, M.S., Sanghi, D., et al., 2017. DDoS attacks in cloud computing: Issues, taxonomy, and future directions. Comput. Commun. 107, 30–48.

Sood, K., Liu, S., Yu, S., et al., 2015. Dynamic access point association using software defined networking. In: 2015 International Telecommunication Networks and Applications Conference. ITNAC 2016, IEEE, pp. 226–231.

Sudar, K.M., Deepalakshmi, P., 2020. A two level security mechanism to detect a DDoS flooding attack in software-defined networks using entropy-based and C4.5 technique. J. High Speed Netw. 26 (2), 1–22.

Sultana, N., Chilamkurti, N., Peng, W., et al., 2019. Survey on SDN based network intrusion detection system using machine learning approaches. Peer Peer Netw. Appl. 12 (2), 493–501.

Sun, Guozi, Jiang, W., Gu, Yu, et al., 2018. DDoS attacks and flash event detection based on flow characteristics in SDN. In: 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance. AVSS, IEEE, pp. 1–6.

Swami, R., Dave, M., Ranga, V., 2019. Software-defined networking-based DDoS defense mechanisms. ACM Comput. Surv. 52 (2), 1–36.

Tan, L., Pan, Y., Wu, J., et al., 2020. A new framework for DDoS attack detection and defense in SDN environment. IEEE Access 8, 161908-161919.

Tang, T.A., Mhamdi, L., McLernon, D., et al., 2016. Deep learning approach for network intrusion detection in software defined networking. In: 2016 International Conference on Wireless Networks and Mobile Communications. WINCOM, IEEE, pp. 258–263.

Tayyab, M., Belaton, B., Anbar, M., 2020. ICMPv6-based DoS and DDoS attacks detection using machine learning techniques, open challenges, and blockchain applicability: A review. IEEE Access 8, 170529-170547.

Tsai, S.C., Liu, I.H., Lu, C.T., et al., 2017. Defending cloud computing environment against the challenge of DDoS attacks based on software defined network. In: Advances in Intelligent Information Hiding and Multimedia Signal Processing. Springer, Cham, pp. 285–292.

Ujjan, R.M.A., Pervez, Z., Dahal, K., 2019. Snort based collaborative intrusion detection system using blockchain in SDN. In: 13th International Conference on Software, Knowledge, Information Management and Applications. SKIMA, IEEE, http://dx.doi.org/10.1109/SKIMA47702.2019.8982413.

Van Trung, P., Huong, T.T., Van Tuyen, D., et al., 2015. A multi-criteria-based ddos-attack prevention solution using software defined networking. In: 2015 International Conference on Advanced Technologies for Communications. ATC, IEEE, pp. 308–313.

Viet, A.N., Van, L.P., Minh, H.A.N., et al., 2017. Mitigating HTTP GET flooding attacks in SDN using NetFPGA-based OpenFlow switch. In: 2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology. ECTI-CON, IEEE, pp. 660–663.

Wang, S., Chavez, K.G., Kandeepan, S., 2017. SECO: SDN sEcure COntroller algorithm for detecting and defending denial of service attacks. In: 2017 5th International Conference on Information and Communication Technology. ICoIC7, IEEE, pp. 1–6.

Wang, T., Chen, H., 2017. SGuard: A lightweight SDN safe-guard architecture for DoS attacks. China Commun. 14 (6), 113–125.

Wang, X., Chen, M., Xing, C., et al., 2016a. Defending DDoS attacks in software-defined networking based on legitimate source and destination IP address database. IEICE Trans. Inf. Syst. 99 (4), 850–859.

Wang, T., Guo, Z., Chen, H., et al., 2018d. BWManager: Mitigating denial of service attacks in software-defined networks through bandwidth prediction. IEEE Trans. Netw. Serv. Manag. 15 (4), 1235–1248.

Wang, A., Guo, Y., Hao, F., et al., 2014. Scotch: Elastically scaling up sdn control-plane using vswitch based overlay. In: Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies. ACM, pp. 403–414.

Wang, Y., Hu, T., Tang, G., et al., 2019. SGS: Safe-guard scheme for protecting control plane against DDoS attacks in software-defined networking. IEEE Access 7, 34699–34710.

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

Wang, W., Ke, X., Wang, L., 2018a. A HMM-R approach to detect L-DDoS attack adaptively on SDN controller. Future Internet 10 (9), 83.

Wang, L., Li, Q., Jiang, Y., et al., 2016b. Towards mitigating link flooding attack via incremental SDN deployment. In: 2016 IEEE Symposium on Computers and Communication. ISCC, IEEE, pp. 397–402.

Wang, L., Li, Q., Jiang, Y., et al., 2018c. Woodpecker: Detecting and mitigating link-flooding attacks via SDN. Comput. Netw. 147, 1–13.

Wang, J., Liu, Y., Su, W., et al., 2020b. A DDoS attack detection based on deep learning in software-defined Internet of things. In: 2020 IEEE 92nd Vehicular Technology Conference. VTC2020-Fall, http://dx.doi.org/10.1109/VTC2020-Fall49728.2020.9348652.

Wang, M., Lu, Y., Qin, J., 2020a. A dynamic MLP-based DDoS attack detection method using feature selection and feedback. Comput. Secur. 88, 101645.

Wang, J., Wen, R., Li, J., et al., 2018b. Detecting and mitigating target link-flooding attacks using sdn. IEEE Trans. Dependable Secure Comput.

Wang, B., Zheng, Y., Lou, W., et al., 2015b. DDoS attack protection in the era of cloud computing and software-defined networking. Comput. Netw. 81, 308–319.

Wei, H.C., Tung, Y.H., Yu, C.M., 2016. Counteracting UDP flooding attacks in SDN. In: 2016 IEEE NetSoft Conference and Workshops. NetSoft, IEEE, pp. 367–371.

Wu, X., Liu, M., Dou, W., et al., 2016. DDoS attacks on data plane of software-defined network: are they possible? Secur. Commun. Netw. 9 (18), 5444–5459.

Wu, Z., Xu, Q., Wang, J., et al., 2020. Low-rate DDoS attack detection based on factorization machine in software defined network. IEEE Access 8, 17404–17418.

Xiao, P., Li, Z., Qi, H., et al., 2016. An efficient DDoS detection with bloom filter in SDN. In: 2016 IEEE Trustcom/BigDataSE/ISPA. IEEE, pp. 1–6.

Xing, X., Luo, T., Li, J., et al., 2016. A defense mechanism against the DNS amplification attack in SDN. In: 2016 IEEE International Conference on Network Infrastructure and Digital Content. IC-NIDC, IEEE, pp. 28–33.

Xu, T., Gao, D., Dong, P., et al., 2017. Defending against new-flow attack in sdn-based internet of things. IEEE Access 5, 3431–3443.

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

Yan, Q., Gong, Q., Deng, F., 2016b. Detection of DDoS attacks against wireless SDN controllers based on the fuzzy synthetic evaluation decision-making model. Adhoc Sensor Wirel. Netw. 33.

Yan, Q., Yu, R., Gong, Q., et al., 2016a. Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges. IEEE Commun. Surv. Tutor. 18 (1), 602–622.

Yang, X., Han, B., Sun, Z., et al., 2017. SDN-based DDoS attack detection with cross-plane collaboration and lightweight flow monitoring. In: GLOBECOM 2017–2017 IEEE Global Communications Conference. IEEE, pp. 1–6.

Yang, L., Zhao, H., 2018. DDoS attack identification and defense using SDN based on machine learning method. In: 2018 15th International Symposium on Pervasive Systems, Algorithms and Networks. I-SPAN, IEEE, pp. 174–178.

Ye, J., Cheng, X., Zhu, J., et al., 2018. A DDoS attack detection method based on SVM in software defined network. Secur. Commun. Netw. 9804061.

Yin, D., Zhang, L., Yang, K., 2018. A DDoS attack detection and mitigation with software-defined Internet of Things framework. IEEE Access 6, 24694–24705.

You, X., Feng, Y., Sakurai, K., 2017. Packet in message based DDoS attack detection in SDN network using OpenFlow. In: 2017 Fifth International Symposium on Computing and Networking. CANDAR, IEEE, pp. 522–528.

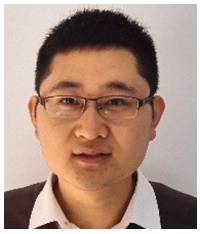
Yu, Y., Guo, L., Liu, Y., et al., 2018. An efficient SDN-based DDoS attack detection and rapid response platform in vehicular networks. IEEE Access 6, 44570–44579.

Yuan, B., Zou, D., Jin, H., Yu, S., Yang, L.T., 2017. HostWatcher: Protecting hosts in cloud data centers through software-defined networking. Future Gener. Comput. Syst. 105, 964–972.

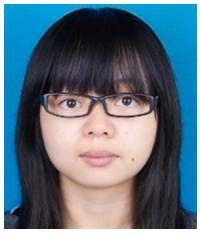
Yuwen, H., Zhang, L., Wang, Z., et al., 2016. Probability-based delay scheme for resisting SDN scanning. In: 2016 2nd IEEE International Conference on Computer and Communications. ICCC, IEEE, pp. 1096–1101.

Zerbini, C.B., Carvalho, L.F., Abrão, Taufik, et al., 2019. Wavelet against random forest for anomaly mitigation in software-defined networking. Appl. Soft Comput.

Zhao, C., Liu, F., 2018. DDoS attack detection based on self-organizing mapping network in software defined networking. MATEC web of conferences. EDP Sci. 176, 01026.
Zheng, J., Li, Q., Gu, G., et al., 2018. Realtime DDoS defense using COTS SDN switches via adaptive correlation analysis. IEEE Trans. Inf. Forensics Secur. 13 (7), 1838–1853.
Zhu, L., Tang, X., Shen, M., et al., 2018. Privacy-preserving ddos attack detection using cross-domain traffic in software defined networks. IEEE J. Sel. Areas Commun. 36 (3), 628–643.



Yunhe Cui, received his Ph.D. degree from the Southwest Jiaotong University, Chengdu, Sichuan, China. He is currently a lecturer of Guizhou University, Guiyang, Guizhou, China. His research interests include DDoS detection and mitigation, intrusion detection and prevention, software-defined networking, traffic engineering, and data centers.



Qing Qian received her Ph.D. degree from Southwest Jiaotong University, Chengdu, China, in 2018. She is an associate professor with the School of Information, Guizhou University of Finance and Economics. Her research interests include security of multimedia information, cryptography, and network security.

Chun Guo, received Ph.D. in information security from Beijing University of Posts and Telecommunications in July 2014. He is currently an associate professor in the College of Computer Science and Technology, GuiZhou University, PR China. His research interests include data mining, intrusion detection and Malware Detection.

Guowei Shen, received his Ph.D. degree from Harbin Engineering University. He is currently an associate professor of Guizhou University. His main research interests include big data, computer network and cyber security.

Youliang Tian received the Ph.D. degree in cryptography from Xidian University. He was the winner of the Youth Science and Technology Award in Guizhou province. He is currently the Distinguished Professor and doctoral supervisor of Guizhou University, the academic leader of State Key Laboratory of Public Big Data. His research interests include algorithm game theory, cryptography and security protocol, big data security and privacy protection, blockchain and electronic currency.

Huanlai Xing received his Ph.D. degree in computer science from University of Nottingham, Nottingham, U.K., in 2013. He is an Associate Professor with the School of Information Science and Technology, Southwest Jiaotong University. His research interests include evolutionary computation, network coding, multi-objective optimization, and software defined networks. He has authored and co-authored over 30 peer-reviewed journal and conference papers. Dr Xing is a Member of IEEE and ACM.

Lianshan Yan (S'99–M'05–SM'06) received the Ph.D. degree from the University of Southern California, Los Angeles, CA, USA. He is a fellow of the Optical Society of America (OSA) and a senior member of the IEEE. He is currently a full Professor at Southwest Jiaotong University, Chengdu, China. He serves as a Frequent Referee for more than 20 journals. He is the holder of 13 issued U.S. patents. He is the author and coauthor of more than 200 papers published in prestigious journals and conference proceedings, including 7 invited journal papers and more than 30 invited talks.