Computer Science Review 37 (2020) 100279

Contents lists available at ScienceDirect

Computer Science Review

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

Review article

Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges and future directions

Jagdeep Singh, Sunny Behal *

Department of Computer Science and Engineering, Shaheed Bhagat Singh State Technical Campus, I.K.G. Punjab Technical University, Kapurthala, Punjab, India

ARTICLE INFO

Article history:
Received 22 April 2020
Received in revised form 1 June 2020
Accepted 8 June 2020
Available online 20 June 2020

Keywords:
DDoS
Software Defined Networking
Detection
Mitigation
Review

ABSTRACT

Many security solutions have been proposed in the past to protect Internet architecture from a diversity of malware. However, the security of the Internet and its applications is still an open research challenge. Researchers continuously working on novel network architectures such as HTTP as the narrow waist, Named Data Networking (NDN), programmable networks and Software-Defined Networking (SDN) for designing a more reliable network. Among these, SDN has emerged as a more robust and secure solution to combat against such malicious activities. In SDN, bifurcation of control plane and data plane provides more manageability, control, dynamic updating of rules, analysis, and global view of the network using a centralized controller. Though SDN seems a secured network architecture as compared to the conventional IP-based networks, still, SDN itself is vulnerable to many types of network intrusions and facing severe deployment challenges. This paper systematically reviews around 70 prominent DDoS detection and mitigation mechanisms in SDN networks. These mechanisms are characterized into four categories, viz: Information theory-based methods, Machine learning-based methods, Artificial Neural Networks (ANN) based methods and other miscellaneous methods. The paper also dowries and deliberates on various open research issues, gaps and challenges in the deployment of a secure SDN-based DDoS defence solution. Such an exhaustive review will surely help the researcher community to provide more robust and reliable DDoS solutions in SDN networks.

Contents

* Corresponding author.
E-mail address: sunnybehal@sbsstc.ac.in (S. Behal).

https://doi.org/10.1016/j.cosrev.2020.100279
1574-0137/

1. Introduction

Over the last two decades, there is considerable growth in the use of Internet-based services and applications. At present, around 57% of the global population is the user of Internet [1]. Consequently, there is a significant rise in the concern of Internet security. The Internet has always been vulnerable to different security threats. The Worms, port scans, denial of service attacks, and Trojans etc. are some common anomalies on the Internet. Many researchers have expressed their interest in denial of service (DoS) attacks. A DoS attack is a deliberate attempt to stop benign users from accessing particular network resources. On February 7, 2000, 15 years old hacker, named mafiaboy, launched a series of DoS attacks against various e-commerce sites, including Amazon and eBay [2]. After that, more advancements took place in the modus operandi of a DoS attack.

The attacker start recruiting geographically distributed multiple devices for launching such attacks, called a Distributed Denial of Service (DDoS) attack. In a DDoS attack, the intruder finds vulnerabilities in the network and injects a malicious program, known as Trojan Horse, in the computer systems without the awareness of users. By replicating this malicious program in the multiple devices connected to the network, intruder create an army of compromised computer systems which they control to initiate DDoS attacks. These compromised machines are often known as bots, and the group of these bots is called a botnet [3]. The whole botnet is remotely under the control of a human operator called bot-master [3–5]. To launch a DDoS attack, the assailant send commands to all the compromised machines, and then these computer system start sending useless traffic towards the victim. Depending on the number of compromised devices (which is in millions now a days), the victim likely to get overwhelmed with the useless traffic packets. Consequently, the victim resources are not accessible to legitimate users, and the victim is said to be under a DDoS attack.

With rapid developments in the field of IT infrastructure, size of networks and hence, its complexity has increased manifolds. With this, the essential properties of a network such as integrity, confidentiality, authentication, availability of information, and non-repudiation are becoming challenging to ensure [13]. In the past years, many researchers and industries shift their focus in designing more robust, scalable and secure networks [14]. The latest advancements such as SDN (Software Defined Networking) is a step towards the establishment of a dynamic and centralized nature of the network as compared to the static and distributed environment of traditional networks. Current widespread adopted networks are complicated and hard to manage [15]. To implement the high-level network policies in the conventional IP-based networks, network administrators need to configure every individual network device using the vendor-specific commands [16]. Consequently, it becomes very challenging to implement desired policies and reconfiguration of network devices in current IP-based networks. Another aspect of the rigidness of these networks is vertical integration. The control plane (which decides how to handle network traffic) and data plane (which accomplishes traffic forwarding according to the decision taken by control plane) are tightly coupled in the network as shown in Fig. 1(a) which reduces the flexibility and hampering the innovation of networking infrastructure. As an immediate solution, network infrastructure should be scaled to overcome the requirements of computer networks, but this expansion would further increase the overall complexity of the network [17]. In the recent years, network industries and networking research communities deployed many solutions to design a better future network [18] such as HTTP as the narrow waist [19], Named Data Networking (NDN) [20], programmable networks [21], and Software-Defined Networking [22]. Among these, SDN is touted as the most capable solution to handle the aforementioned problems of the current network. SDN is an emerging network architecture which gives a hope for efficient network infrastructure. Firstly, it abolishes the vertical integration by bifurcating control plane (which is the network's control logic) from the data plane (underlying routers and switches that forward the network traffic as per network control logic) as shown in Fig. 1, (b) Secondly, with this separation, control logic of networks installed in logically centralized controllers and network switches become simple packet forwarding devices which adds flexibility, speed of implementation, programmability, and simplifies the network management.

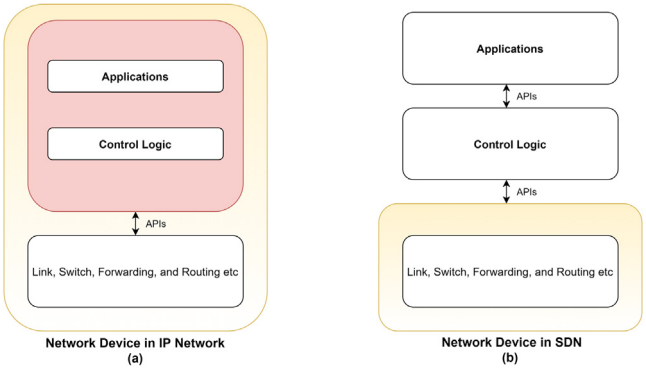

Fig. 1. Comparison of network device in IP network and SDN.

Though SDN architecture is capable of enhancing the security of a network with centralized controllers, global visibility of network, and on-demand creation of traffic forwarding rules [23]. However, still SDN has its own concerns and challenges in the form of network security, scalability and supportability. Among all these problems, security is at the forefront. Since the centralized controller is responsible for the management of the network, the failure of this controller would hamper the whole network. The centralized control and communication between controller and switches might be the target of sophisticated DDoS attacks.

Shin et al. [24] and Fonseca et al. [25] have demonstrated that how DDoS attacks downgrade the performance of SDN networks. SDN architecture divides the network into data plane, control plane, and application plane. The DDoS attacks in SDN can also be categorized accordingly into application-layer DDoS attacks, Control layer DDoS attacks, and data layer DDoS attacks depending on the target plane [8].

Table 1
Comparison of our work with other similar survey papers.

	Bawany et al. [6]	Joelle et al. [7]	Dong et al. [8]	Fajar et al. [9]	Xu et al. [10]	Kalkan et al. [11]	Singh et al. [12]	Our Work
Architecture of SDN	✗	✓	✓	✗	✗	✗	✗	✓
SDN security challenges	✗	✗	✗	✗	✗	✗	✗	✓
Types of DDoS attack in SDN	✗	✓	✗	✓	✗	✓	✓	✓
Categorization of defence solutions	✓	✓	✓	✗	✓	✗	✓	✓
Plane implementation	✗	✗	✗	✗	✗	✗	✓	✓
Controller implementation	✗	✗	✗	✗	✗	✗	✓	✓
Research challenges	✓	✗	✗	✗	✗	✗	✓	✓
Research gaps	✗	✗	✗	✗	✗	✗	✓	✓

Many survey papers on DDoS defence solutions are available in the literature [6–12,23,26–28], which closely related to our work, but only a few authors focused on DDoS attack detection and mitigation in SDN context. However, some authors like Bawany et al. [6], Joelle et al. [7], Dong et al. [8], Fajar et al. [9], Xu et al. [10], Kalkan et al. [11], and Singh et al. [12] attempted to review DDoS attack detection and mitigation mechanisms. But our paper has performed a more comprehensive review with more technical details than these reviews. Besides this, our paper is multi-dimensional (see Table 1). We have summarized a list of Research gaps which these review paper have not considered.

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

- In this paper, we present a detailed SDN architecture along with its key features that make it more robust and secure than conventional IP-based networks. It has been observed that even SDN is a reliable option, but still, it has many loopholes which make it vulnerable to many types of threats. Further, vulnerabilities present in the SDN networks are summarized so that researchers could understand the problem and can devise a better mechanism to design secure SDN-based networks.
- We provide a detailed review of various types of DDoS attacks launched in the context of SDN.
- We present a state-of-art comprehensive survey of existing DDoS attack detection and mitigation techniques in SDN networks. Further, we categorize these defence solutions into four categories and provide more technical details of these mechanisms.
- After comprehensive of DDoS based solutions in SDN, we enlisted a list of research gaps in the existing SDN based DDoS solutions.
- At the end, a brief summary of research challenges in the deployment and security of SDN is provided.

The rest of the paper is organized as: Section 2 explains the underlying architecture of SDN and various security challenges. Section 3 compare the SDN network with traditional IP networks along with DDoS attacks on multiple layers of SDN. Section 4 provides a comprehensive survey of existing DDoS detection and mitigation techniques. In Section 5, some prominent research challenges and research gaps in the existing work are summarized, and finally, Section 6 concludes our work by highlighting future directions.

2. Background of software defined networking

Widespread adopted today's vertically integrated network design is complex and hard to manage [16]. Integration of the control plane and data plane into network devices makes current network design inflexible. The network operators need to configure each network device using low-level vendor-specific commands to implement high-level network policies. Because of the complexity of the current network design, it take 5 to 10 years to design, evaluate and deploy a new routing algorithm [15].

Software-Defined Networking (SDN) is a novel network paradigm which gives a new hope to overcome the limitations of the vertically integrated current network infrastructure. This new prototype of the network eradicates the vertical integration by implementing control logic separately from the data plane. In this networking style, the control logic is implemented (logically centralized) at a different layer (control plane) then the network devices (data plane). Logically centralized controller, or network operating system, is responsible for implementing network traffic rules and network devices such as switches become simple packet forwarding devices [29]. Fig. 2 shows how the OpenFlow switch does processing at receiving of the new packet. When the new packet arrives at the OpenFlow-enable switch, it will find the flow rule entry for the packet in its flow table. If switch finds the entry of flow rule, it will directly forward the packet as per defined norms. Otherwise, the switch asks the controller for flow rule using *packet_in* message. Then the controller dynamically devises a flow rule for the incoming packet and send new rule using *packet_out* message and rule will get stored in the flow table of the switch for some time for future packets [30]. As control plane and data plane is separate from each other, the communication between two established using well-defined programming interfaces (APIs). Using these APIs, the control plane has direct control over the state of the network in data plane elements. OpenFlow [31,32] is a widely adopted protocol among others which establishes communication between OpenFlow switches and logically centralized controller. Initially, SDN and OpenFlow started as academic experiments only [31], but in the past few years, it gains significant traction in the industry. Now, commercial switches from many vendors have the support of OpenFlow API in it. Due to the strong momentum of SDN, many large companies such as Google, Facebook, Microsoft, Verizon, and Deutsche Telekon start funding to Open Networking Foundation (ONF) for promotion and adoption of Software-Defined Networking [15]. Specific architectural aspects are surveyed by some papers [33–35], and overview of OpenFlow can be found in [33] and [34].

2.1. Existing state of networking

Mainly, the functionality of the network divided into three planes such as data plane, control plane and management plane. The data plane consists of all the networking devices which are responsible for forwarding network traffic. The control plane represents a set of rules to populate the forwarding tables of switches in the data plane. The management plane contains the software services such as SNMP [36] based services which are used to configure and monitor the control functionality of network [23]. Any network policy is defined in the management plane, the control plane enforces this policy to the network, and it is executed by data plane by forwarding network traffic accordingly.

Table 2
Comparison of traditional vs SDN networks.

Parameter	Traditional network	SDN
Network management	**Difficult** Each and every device need to update individually.	**Easy** Programmability of separated planes is easy using centralized control
Performance	**Limited Information** Network devices are relatively statically configured.	**Cross-Layer Information** Devices have cross-layer information so can be dynamically configured.
Configuration	**Difficult** Devices need to configure using vendor specific commands manually.	**Easy** Each device remotely configured using centralized controller.
Innovation	**Slow** Hardware has limited capacity	**Fast** Policy deployment is fast
Testing environment	**Limited** Rigidness of network devices makes it hard to test new policies.	**Extended Features** Devices can be dynamically programmed.
Global view	**Not Available** Decentralized nature of network	**Available** With centralize controller, administrator have global view of network.

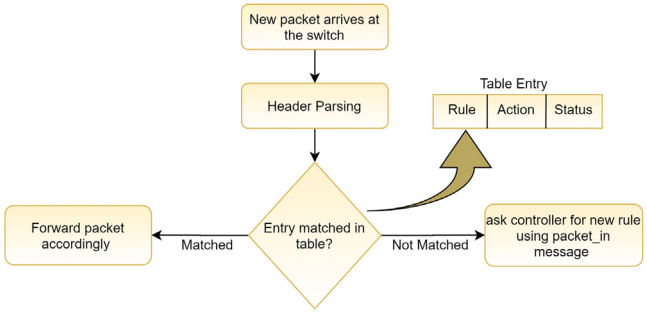

Fig. 2. Typical processing at a switch while receiving a new packet.

In the conventional IP-based networks, the data and control planes are integrated and embedded into network devices with specific algorithms to monitor, control and route network traffic. Hence the whole structure of the network becomes very decentralized. It was an essential concern for the design of the Internet at the time when it was designed. It seemed the best way to ensure network resilience, which was a fundamental design goal. However, with the rapid growth of networking, this integration and complexity make it worse for the maintenance of a stable state of network security. For example, updating or change of network policy in these systems is practically very costly and unmanageable. Although this rigid approach has been relatively effective in terms of network performance, this complicated and static architecture has been often reported in networking literature by many authors [16,18,37–39].

The rigidness and complexity are the other two main characteristics which are responsible for this vertically integrated industry where innovation is very tough to proceed. For the management of the current network, few vendors offer the proprietary solution of specialized hardware, operating systems, and network applications. The network operators have to purchase and maintain these management solutions, and corresponding skilled teams are required. With this management idea, there is a need for huge capital and operational investment. There is an enormous number of middle-boxes, such as IDSs (Intrusion Detection Systems), Firewalls and deep packet inspection engine, available to manage the network. A recent survey of 57 enterprises networks shows that the number of middle-boxes is already on an equality with the number of routers in the network [15,40].

2.2. Software-defined networking

The term Software-Defined Networking initially used to represent ideas and work around academic experiments at Stanford University [41]. As initially stated, SDN refers to network architecture where the control plane is separated from the data plane and remotely controlling the forwarding state in the data plane. Kreutz et al. [15] defined Software-Defined Networking as

- By bifurcating data plane and control plane, control functionality detached from network devices as a separated entity called SDN controller or Network Operating System (NOS) and network devices become simple packet forwarding elements.
- Forwarding decisions are flow-based, instead of destination-based, and defined by field values of the packet as a match criterion and set of actions. In SDN, a flow is a sequence of packets between sender and recipient devices. This flow-based abstraction unified behaviour of various types of network devices such as switches, routers, firewalls, and middle-boxes [42].
- The network is programmable through APIs running on the top of the network operating system that interacts with underlying network devices in the data plane.

Software-Defined Networking is defined by three fundamental abstractions of forwarding, distribution and, specification [43]. Abstraction is a pervasive feature of many computer architectures and is an essential tool for research in computer science [44].

3. SDN as a better approach than conventional IP-based networks

The core idea of SDN architecture is the separation of data plane and control plane, which introduces a high potential of innovation in the design of network [45]. This bifurcation brings greater programmability in the SDN networks. Resultantly, there is an improvement in network traffic handling and configuration of SDN-enabled devices. In conventional networks, devices are embedded with the control logic. If there is need of any modification in the policy, logic on every device needs to change manually

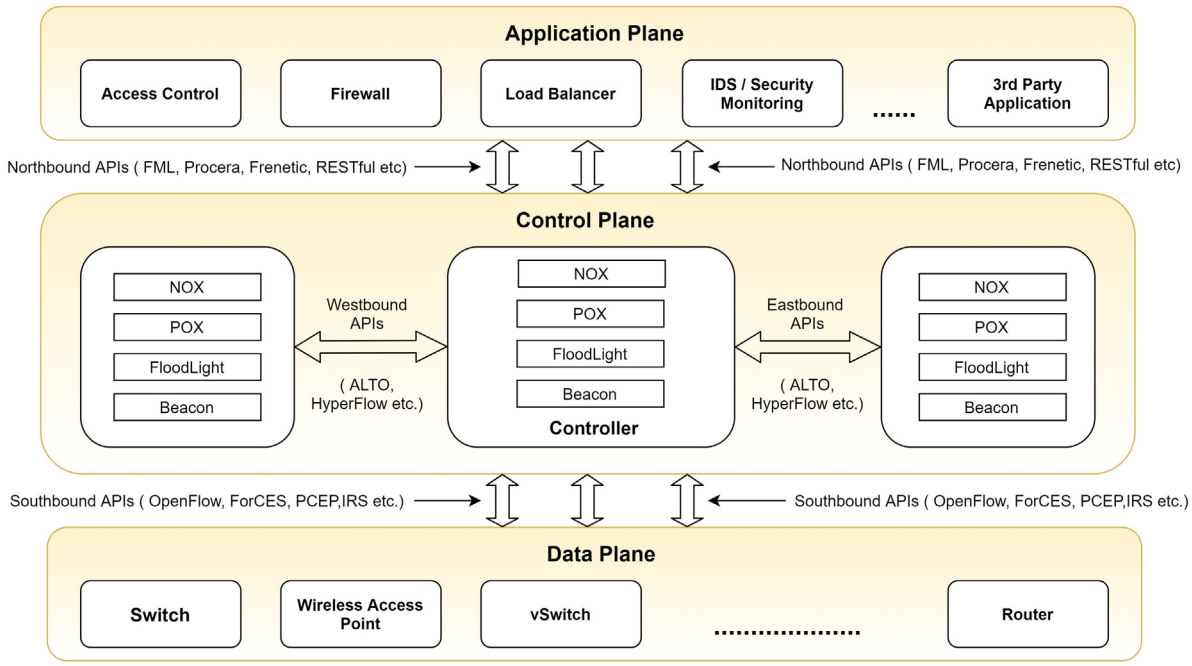

Fig. 3. Layered architecture of software-defined networking.

using vendor-specific commands, which is a very tedious task to perform. On the other hand, in SDN, the control logic is separated, and under centralized supervision, it makes it easy for the administrators to change control logic remotely using southbound APIs. New innovative policies are complicated to implement in the traditional switches as these devices are vendor-specific and have limited hardware capabilities. On the other hand, in SDN, policy up-gradation is very speedy task due to the centralized controller (Policy Maker). There is a minimal area to test the new policies in the traditional network, but SDN has more significant opportunities for testing as compared to the traditional network. The core aspects of SDN, in contrast to a conventional IP-based network, is summarized in Table 2.

There is a need to understand some commonly used terminology related to the working of SDN as described below:

- *Data Plane*: It includes various packet forwarding devices such as routers, wireless access points, switches, and virtual switches. In SDN, they are OpenFlow switches. These devices contain flow tables to store rules for packet forwarding, which are defined by the control plane and devices purely focus on packet forwarding.
- *Southbound Interfaces*: These are protocols which facilitate efficient control over the data plane. There are many protocols available such as OpenFlow [31], ForCES [46], OpFlex [47] and Protocol-Oblivious Forwarding (POF) [48], but many organizations are working for standardization of OpenFlow as it has become the de facto protocol [49].
- *Control Plane*: It involves the SDN controller or NOS (Network Operating System) which is the brain of the whole network e.g. NOX [50], POX [51], FloodLight [52], and Ryu [53] etc. It is responsible for making packet forwarding decisions and implementing them into network devices.
- *Northbound Interface*: It is an interface between the control plane and the application plane. These APIs provided by the NOS to the application developers. It helps to program the network and hides the internal details of the network. For example FML [54], Procera [55], NetKAT [56], and Frenetic [57] etc are commonly used APIs.
- *Application Plane*: It is also known as management plane and is the first plane in the SDN architecture. All the applications, which are written by developers to manage the network, execute on this plane. In SDN, fault monitoring and configurations are done through application plane.
- *Eastbound and Westbound*: In SDN, all network controllers are logically centralized, but it may physically be distributed. These controllers communicate through eastbound and westbound interfaces. As a single controller is only able to handle a small network, so if it fails, the whole network would be compromised. In this case, multiple controllers are required, and if one fails, it can inform other controllers to take over the traffic handling. HyperFlow [58], ALTO [59], and Onix [29] are suggested strategies.

3.1. Architecture of SDN

SDN replaces the fixed and complex conventional network into a dynamic programmable network. It is the biggest revolution that happened in the field of networking [60]. SDN architecture is the future of networking. The basic principle of this architecture is the separation of data plane (network devices) and control plane (control logic). This division makes the management and development of various network devices more accessible than a traditional IP network and easy administration of applications. This SDN architecture is divided into three-layers; Application Layer, Control Layer, and Data Layer, as shown in Fig. 3.

- **Application Layer** It is a topmost layer in SDN architecture. This layer handles all business and security applications. Metering, routing, QoS, load balancer, Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS), firewall implementation, and mobility management are examples of essential software services controlled by this layer. This layer communicates with a lower layer using the northbound application interfaces [55].
- **Control Layer** This layer is the mediator in the other two layers, the application layer and data layer. This layer consists Network Operating System (NOS), called the network

Table 3
Security Challenges of SDN at different planes.

Challenge	Application plane	Control plane	Data plane
Threats from applications	✗	✓	✗
Threats due to scalability	✗	✓	✗
DoS & DDoS attacks	✗	✓	✗
Authentication and authorization	✓	✗	✗
Access control and accountability	✓	✗	✗
Differentiation between genuine and malicious flow rules	✗	✗	✓
Limited number of flow entries	✗	✗	✓
Dependence of control plane	✗	✗	✓
Switch-controller link	✗	✗	✓

controller, to control the overall functionality of the network. The responsibility of a logically centralized controller is to manage the entire network and takes decisions on routing, flow forwarding, and packet dropping through programming [61,62]. The controller is logically centralized and physically distributed environment which communicate with each other using west-bound, and east-bound interfaces and this layer communicate with the layer beneath it using south-bound APIs such as OpenFlow [30] and NetConf [15] etc.

- **Data Layer** The primary function of this layer is forwarding the packets according to policies/rules assigned and developed by the controller. This layer consists of physical network devices such as switches, routers, and access points as well as virtual switches (OpenvSwitch, Indigo, Pica8, Nettle, Open Flowj) etc. [63–65].

3.2. Security challenges of SDN

The global visibility of the SDN network state and run-time manipulation of traffic forwarding rules enhance the security of the network as any conflict can be resolved remotely using a logically centralized controller [66–68]. Alongside, this centralization of the controller and its visible nature increases the potential of threats to security, DoS, and DDoS attacks are prime examples [45]. The SDN controller can be a single point of failure and would be a single point of network failure. There are expectations that security challenges may increase with the gradual deployment of SDN technologies. These vulnerabilities in SDN are concentrated around different layers of SDN architecture. Hence, security challenges existing in different planes are summarized in Table 3 and discussed below:

3.2.1. Application plane

The ability to control the network by software and centralization of the network controller are two principal properties of SDN architecture, which make the foundation of networking innovations and basis of security challenges. There is a lack of standards to OpenAPIs for applications to control the functionality and services of the network through the control plane [69]; with this, applications may cause severe threats to network security. Although OpenFlow deployed some algorithms in the form of security applications for flow-based security detection, however, there are no convincing OpenFlow applications [70]. Due to different independent development environments, different paradigms and programming models could create interoperability limitations and collision in security policies. Some of the security concerned posed by SDN applications are:

- *Authentication and Authorization*: In OpenFlow SDN, most of the functions of the controller are implemented as applications generally developed by third parties. These applications have the privilege to access the network resources and to manipulate the various network services [71]. Hence, it is necessary to authenticate every request to secure network resources. However, authentication of the large number of applications in SDN is a very challenging task.
- *Access Control and Accountability:* As most of the services implemented through application in SDN, there is a need for control access and accountability mechanism to ensure the security of the network. There are some applications which use an instance of other applications to run; hence a malicious application can bypass the access control. There are three types of application identified by Hartman et al. [72]. They divide applications into three classes as Network Sensitive Applications, Services for the network, and Packaged network services. Nested applications (which access control using an instance of other applications) are a severe challenge to SDN security [73].

3.2.2. Control plane

In SDN, the logically centralized controller is only responsible for decision making for packet forwarding. Hence, it is highly targeted point for attacks and for carrying malicious activities in the SDN network. Main security aspects of control plane are as follows.

- *Threats from Applications*: Applications implemented in the management plane may cause a grave threat to SDN security. Generally, the controller is responsible for authentication of applications and authorization of resources needed to applications with proper isolation, tracking and auditing [72]. So there is need of separating applications according to their security implications before granting access to any resource. Therefore, there should be a customized security check for different types of applications in northbound APIs of the controller. Such customization have not been demonstrated yet [23].
- *Threats due to Scalability*: In OpenFlow, the centralized controller is the most complex part of architecture where forwarding decisions are taken. The controller needs to implement flow rule for every new flow in the data path, and if there are enough number of new flows, then the controller can quickly become a bottleneck [74]. Today's implementations of controllers are not able to handle a vast number of new flows using OpenFlow in high-speed networks having 10 Gbps links [75]. At this point, controller incapability of scalability becomes a security threat. Hence, controller's lack of scalability makes it a preferred choice of distributed DoS and DoS attacks.
- *DoS & DDoS Attacks*: Denial of service attacks and Distributed-Denial of service attacks are the most challenging network threat. This attack is an intentional attempt to make network resources unavailable to legitimate users. A DoS attack in software-defined networking is demonstrated by Shin et al. [24] that exploits the control-data planes

separation logic of SDN. A secondary controller suggested by some authors [25] to overcome the problem, but the secondary controller is also susceptible to these attacks. However, the use of multiple controllers is not the solution of DoS and DDoS attacks as it can lead to cascading failure of all controllers as described in [76].

3.2.3. Data plane

In OpenFlow networks, control plane and data planes are separated entities and data plane devices are simple traffic forwarding devices. Whenever a first packet arrives from the new host at OpenFlow switch, new flow rule would be installed by the controller in the flow table of the device. Every switch has a limit on flow table entries due to space constraint. This scenario creates new security challenges, as described below.

- *Differentiation between Genuine and Malicious Flow Rules*: As network switches in SDN are only packet forwarding devices and rule making is done at the control plane, recognition of genuine and false flow rule is foremost security challenge for the data plane.
- *Limited Number of Flow Entries*: As every switch need to maintain a table of flow rules. Due to limited memory, there is the number of flow entries where data plane prone to saturation attacks.
- *Dependence on Control Plane:* In SDN, the security of the data plane is directly influenced by the security of the control plane [75]. Hence, if the network controller is compromised, the various data plane devices will be compromised.
- *Switch-Controller Link*: In SDN architecture, the data plane receives flow rules from the control plane. However, if due to any reason, such as control plane failure and disconnection of the control plane, the data plane becomes offline. Hence, the link between the data plane and control plane can be a desired choice for attacking the SDN network.

3.3. SDN defeating DDoS attacks

In the traditional network, DDoS attacks are increasing in size, frequency, severity, and are more sophisticated. It means current IDS for DDoS attacks are limited to combat the problem. Attackers have more refined methods to bypass the existing protection shields [77,78]. On the other hand, SDN has some features which offer advantages to defeat the DDoS attacks up to some extent [17,33,79–81].

(1) *Separation of the Control Plane and Data Plane*: It is complicated to perform a large-scale experiment in traditional networks. Moreover, it is too difficult to examine newly deployed algorithm in traditional networks because of control logic are embedded in devices, and every device needs to update individually. However, SDN decouples bottom two planes and making it easier to test extensive mechanisms. SDN has great functionality of dynamic configuration, which leads to permit experiment environments. Progressive ideas can be easily deployed and smoothly shifted from trial phase to the working phase [82].
(2) *Global View of Network*: The controller (Centralized Network Operating System) has a global view of the network and monitors the traffic in the network. Centralizing of SDN controller makes it easy to isolate the compromised host from the legitimate host using information obtained by requesting the end hosts [49].
(3) *Traffic Analysis based on Software*: Various application utilities are running in the application plane of SDN to monitor and configure the network devices. Many software and algorithms are available to analyse the network traffic, which reduces the burden of switches to parsing the traffic [49].
(4) *Programmability of Network*: Application plane contains applications to program the controller, which further control the behaviour of the network. The programmability of the SDN network makes it more flexible because more intelligence can be deployed at any time [83]. As the network is programmable, the incoming traffic can be processed to find malicious traffic or hosts to maintain the performance of the network.
(5) *Dynamic Network Policy Updation*: The dynamic modifications of flow rules on the OpenFlow switches is the immediate response in DDoS attack mitigation. Based on the analysis of the traffic, new innovative algorithms to block the traffic can be propagated instantly [84]. In a traditional network, it is a complicated task to implement a new rule to every device, but in SDN, it is straightforward to update switches dynamically.

3.4. SDN as a victim of DDoS attacks

By decoupling the data plane and the control plane, SDN is capable of mitigating DDoS attacks. However, the security issues of SDN itself remains to be unaddressed [85–87]. Due to its architecture, SDN itself may be a targeted choice for DDoS attacks. As discussed, SDN is vertically separated into three functional layers. These layers are vulnerable to DDoS attacks. These vulnerabilities (see Table 4 and Fig. 4) are summarized below:

- **Buffer Saturation** When a new packet comes from the host, flow rule entry is matched in the flow table. For all unmatched entries, switch requests to the controller to draw new flow rule for the flow. Firstly, the switch will send part of the packet to the controller to bring flow rule for a new connection, and another part of the packet will buffer in the memory. If buffer memory gets full, then it sends the whole packet to the controller. Buffered packets will usually be processed via a *packet_out* or Flow-mod message from a controller, or automatically expired after some time [30]. The attacker can exploit this feature of SDN by sending fake flows to the switch. At this place, the attacker can send malicious or counterfeit packets to the switch whose entries are not in the flow table, and the switch has to send *packet_in* messages to the controller. With a large number of fake flows, the buffer will exhaust in a very short time, and the switch will be unable to buffer the legitimate packets too, which leads to buffer saturation attack in SDN [88].
- **Flow Table Overflow** In SDN, OpenFlow switches maintain a memory called Ternary Content Addressable Memory (TCAM) to store the flow tables. Whenever a new packet comes from the host, its entry is matched by the switch in the flow table. The SDN switches send a request to the controller for new flow rule if the match is not found in the flow table [88]. Each flow rule has a specific timing to remain in the flow table. The attacker uses this feature to launch the DDoS attack. The attacker sends a large number of fake packets to the switch, and the switch requests a new rule for each packet, and after some time all the entries in the flow table will be replaced with new entries and memory gets full with flow rules for fake flows. At this stage, legitimate entries find no space in the flow table and get dropped, and resources will not be available to authorized users. The switch can send an only limited number of *packet_in* messages as switches have limited TCAM memory because of its cost (400 times more than RAM) and power usage (100 times more than RAM) [89].

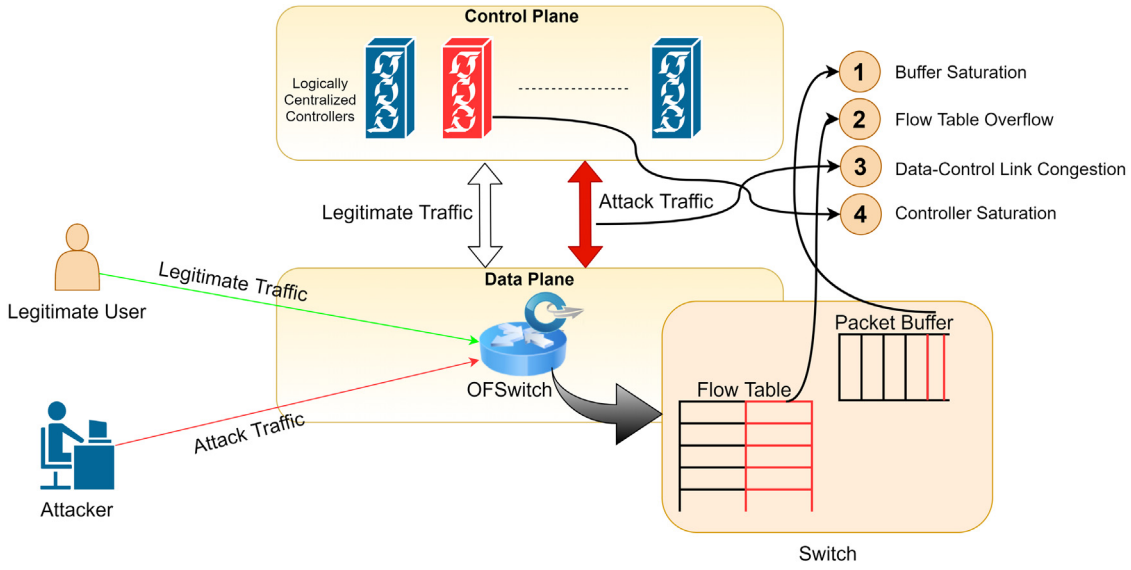

Fig. 4. Conceptual mapping of vulnerabilities in SDN architecture.

Table 4
Summary of vulnerabilities in SDN architecture.

Type of Vulnerability	Remarks
Buffer saturation	Switches need to buffer portion of packet header field. OpenFlow switch has limited memory to buffer the information.
Flow table overflow	Switches have TCAM memory to store the flow tables. This memory is limited because of high cost and power uses of memory.
Congestion of control and data plane link	For every new traffic flow, switch requests new flow rules using *Packet_in* messages from controller and receive the reply using *Packet_out* messages, which cause high traffic on the link.
Controller saturation	For every switch in the network, the centralized controller is responsible for drawing new flow rule. Multiple malicious requests can overload the controller.

- **Congestion of Control-Data Plane Link** When a packet comes whose entry is not in the flow table, the switch will ask the controller for the flow rule by sending some part of the packet using *packet_in* request and another part will be buffer. When there is no buffer space left, then the switch will send the complete packet to the controller. Posting the entire packet towards the controller using single-channel will create constrictions in the channel. When an attacker sends multiple fake packets to the switches and switches forward these packets to the controller, it will create a bottleneck for legitimate requests too.
- **Controller Saturation** Finally, when the attacker floods fake packets towards switches and switches send all requests to the controller, the controller gets busy in satisfying the fake request. This processing of fake packets exhausts the throughput and processing power of the controller, which leads to downgrading the whole SDN architecture. The attacker usually creates a significant amount of abnormal packets to exhaust the controller.

3.5. Type of DDoS attacks in SDN

The SDN architecture for the network is a new concept to decouple the data plane from the control plane to enhance the manageability and security of the network. Separation of Control plane from the data plane, global view of the network, the programmability of the network, traffic analysis based on the software, and dynamic updating of network policies are some inherent characteristics of the SDN to enhance the security of the network [49]. Nevertheless, several attacks exist which infect the data plane, control plane and, interfaces between planes. These attacks are spoiling the architecture of the SDN and making it as challenging network architecture [90]. Fig. 5 shows which plane is vulnerable to a particular attack in SDN architecture. Such attacks are discussed as follows:

(1) *Packet_in flooding:* Whenever vswitch receive the unmatched flow, it will request (using *packet_in* message) to the centralized controller to draw a forwarding rule for the new flow via the southbound interface. The attacker sends numerous packets to the vswitch with spoofing the IPs and forces the vswitch to send bulk *packet_in* messages to the controller. Resultantly, this flood overloads the controller and makes it unreachable to the legitimate users as the controller will be busy in handling only fake flow requests [91].
(2) *Spoofing of Switch*: In this attack, attacker can spoof the IP address of switch and control messages can send using this switch with the modified address. When a switch establishes a connection with the controller and communicates, at same time, a second malicious switch (switch with spoofed IP having same hardware and name) will be turned on and establishes a connection with the controller. The controller will terminate the connection with a legitimate switch and communicates with the malicious one, gradually degrade the performance of the network [92]. This attack can cause fake requests to the controller, and it should be controlled with some mechanism.

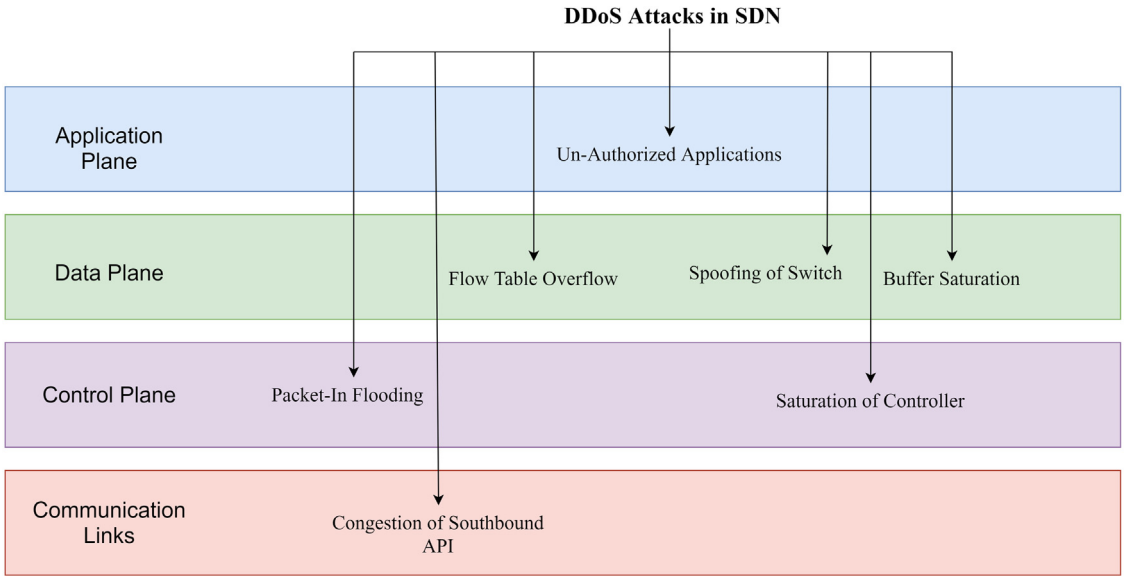

Fig. 5. Types of DDoS attacks at different layers of SDN.

(3) *Flow Table Overflow*: When OpenFlow switch requests for the new flow rule from the controller, the new rule sent by controller gets store in the flow table of the switch. Every flow rule has fixed time out value and after that time, the old rules evicted by the switch from the flow table [93]. TCAM is very limited in capacity to store the flow table entries, because of it is very costly and power hungry [89]. The attacker uses this feature to overwhelm the switch, the attacker sends many new fake flows to the switch, and consequently, switch's flow table will run out of memory within short time and will have only fake rules in it. All legitimate entries get erased from the flow table and downgrade the performance.

(4) *Congestion of Southbound APIs*: When OpenFlow switch sends *packet_in* request to get new rule from the controller, it sends some part of the packet, and another part is stored in the buffer. Nevertheless, if the buffer is full, then the switch is liable to send the whole packet to the controller [88]. At this point, by sending multiple fake flows to the switch, an attacker can easily overload the single bandwidth used in southbound APIs and create congestion in it to make it unavailable to the legitimate users [93].

(5) *The Controller Saturation*: When multiple *packet_in* requests (because of fake flows generated by the attacker) come at the controller, the controller makes the queues to handle these all requests. However, if there are numerous fake packets, controller will remain busy in handling fake requests and its performance will eventually degrade, which is a barrier for the SDN based networks [94]. Attacker can easily generate enough number of fake flows, which can exhaust the processing capacity of the controller. A secondary controller is suggested by some authors [25] to overcome the problem, but the secondary controller is also susceptible to these attacks. However, the use of multiple controllers is not the solution of DoS and DDoS attacks as it can lead to cascading failure of all controllers as described in [76].

(6) *Buffer Saturation*: When a switch is sending *packet_in* message to the controller, it sends some part of the packet to the controller and other stores in buffer memory. The attacker uses this feature to attack the victim, the attacker sends multiple fake packets to the switch, and in a short time, the buffer will be exhausted. When the switch runs out of internal buffering, must sends the full packet to controllers as part of the event which is another bottleneck for SDN. At this stage, legitimate users are getting trouble to process their flow requests, resultantly, the attacker succeeds in downgrading the performance.

(7) *Un-authorized Applications*: There are many applications in the application plane which gain access to the network resources to provide services to the controller and network. Some applications can use the instance of other applications to access network resources. However, there is no authentication and authorization of applications to check their validity; for this reason, some malicious applications gain access through instances of the other applications and modify the behaviour of the network and become the cause of degradation of network performance.

4. Review of DDoS defence solutions in SDN

In recent years, DDoS detection and mitigation have been a primary concern for researchers. As the new architecture of network, SDN possess some features to combat the problem of distributed denial of service attacks, however, SDN itself has been the target for attackers. Many of the fellow researchers proposed DDoS attack detection mechanisms, and many gave solutions for mitigating the effect of DDoS attacks. This section presents the systematical analysis of recent work done in DDoS attack detection and mitigation in the SDN context.

4.1. Review of DDoS detection solutions in SDN

This section presents a comprehensive review of existing DDoS detection solutions in SDN. All of these solutions are divided into four different categories depending on the type of detection metric and detection mechanism used, namely: Information theory-based DDoS defence solutions, Machine learning-based DDoS defence solutions, Artificial Neural network-based defence solutions and other defence solutions (see Fig. 6). We identify eight different parameters to compare these solutions, viz: year of publication, the scope of solution (Detection or Mitigation), detection metric used, parameters used in the algorithm for detection, target plane, controller type, the dataset used for validation and other key features of the solution. The technical details of all of these mechanisms are given in the subsequent sections.

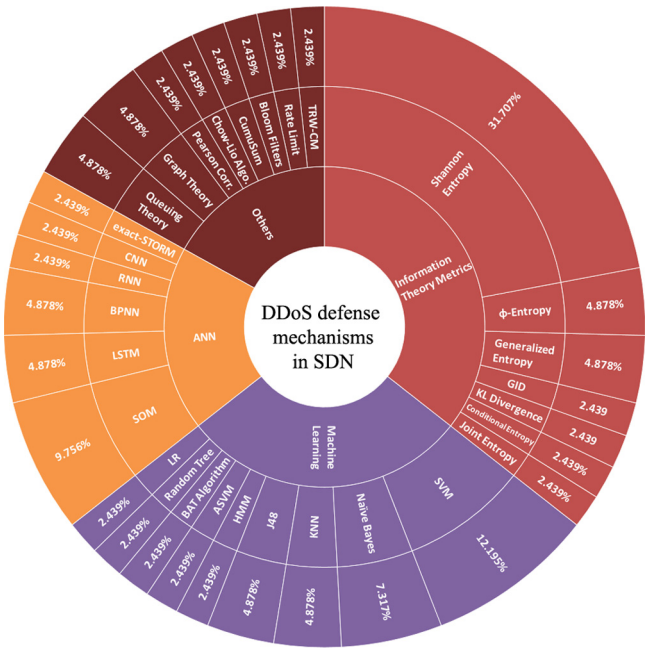

Fig. 6. Classification of DDoS attack defence approaches in SDN.

4.1.1. Information theory-based DDoS defence solutions in SDN

Information theory-based Entropy and divergence metrics are widely used for DDoS attack detection. Entropy represents the randomness in the network features, whereas a divergence metric represents the similarity of two probability distributions. The concept of uncertainty's measurement is coined initially by Claude Shannon in 1948 [95]. The information distance or divergence [96] metric calculated using different probability distributions of traffic flows used to find the abnormality of network traffic. By using the entropy measure, it can be seen how the current network behaviour deviates from normal network behaviour, which leads to the detection of a DDoS attack. Many fellow researchers provided DDoS defence approaches using entropy metric as given in Table 5 and briefly discussed below.

As SDN network is programmable, it allows extracting and analysing the network flow statistics. Giotis et al. [97] extend this feature by decreasing the load of the controller to extract the information. In the proposed approach, data is collected and analysed using the entropy method to detect the anomalies in the network. Collector module is responsible for collecting data periodically and sending it to the anomaly detection module. After that, anomaly detection inspects all flow entries for every time window and identify the malicious flows. After detection mitigation module, implemented flow rules about malicious flows to block them. They validate their approach using the network of the National Technical University of Athens and collect benign traffic. Tcpreplay and Scapy tools are used to generate malicious traffic.

Wang et al. [98] proposed an approach which reduces flow collection overhead in the controller by implementing the scheme on OpenFlow switches. Their proposed approach is based on the flow statistics and used entropy to detect anomalies in the network. They validate their approach using Mininet network emulator. Mousavi et al. [99] introduced an effective and lightweight solution for attack detection in SDN. Their system uses entropy variation of the destination IP address for anomaly detection. The system monitors the destination IP of the incoming packet and traces the count of packets from the same IP to detect an attack in the controller. If packets from the same destination are coming, it reduces the entropy, and it will detect the anomaly in the network. They used a Mininet network emulator for experimental setup and Scapy tool to generate traffic. Botie et al. [100] gave a novel technique based on in-switch processing capabilities to detect and mitigate the DDoS attacks. After monitoring the feature of the traffic, an entropy-based algorithm process the features to detect network anomalies. When the violation is detected, new flow rules are installed on switches to drop malicious flows. Tsai et al. [101] used the entropy-based DDoS detection method in protection. In this scheme, an application running on the controller will monitor incoming traffic and calculates the entropy. If it found attack, it will help to implement a mitigation strategy to block the specific port and this application also send this information to NISD for future examination, and NISD can stop this malicious traffic. They validate their approach by experimental environment using Ryu controller, and Scapy tool is used to generate traffic.

Kalkan et al. [102] described a new scheme of security in SDN network using joint entropy. JESS (Joint Entropy-based Security Scheme), a proposed method has three phases, such as nominal, preparatory, and active mitigation stage. In the first phase, which is the attack-free phase, baseline information is generated by creating nominal pair profiles for each attribute pair. In this phase, all traffic passes to the controller. When congestion is detected, the preparatory stage is commenced, the switch sends only information of the packets to the controller and controller will calculate joint entropies of pair profiles, and it found the difference between entropies exceeds that threshold, DDoS attack is detected. When controller found an attack, it informs to the switch. Then Attack Mitigation module start dropping the attack packets, whereas the legal packets are protected. Sahoo et al. [103] proposed a Generalized Entropy (GE) and Generalized Information distance (GID) based metrics to detect the low-rate DDoS attacks on the control layer of SDN. They employed information distance as a metric to quantify the deviation of network traffic flows. They obtain better results using GE and GID as compare to Shannon entropy and KL-divergence. They validate their approach using Mininet emulator on Linux Ubuntu 14.04 LTS. Jiang et al. [105] proposed a new method EDDM (Entropy-based DDoS Defence Mechanism) to detect and mitigate DDoS attacks running on the SDN controller. This method has three phases such as Window Construction Phase, DDoS Detection Phase and DDoS Mitigation Phase. They used the entropy metric to differentiate the network traffic, whether it is normal or malicious. This method can find bots using the corresponding In-port of the In-switch to mitigate the attack. They implemented the floodlight controller on Mininet to validate their approach. Hong et al. [106] proposed a scheme which balances the load in the attack scenario and non-attack scenario using a dynamic threshold. This scheme calculates the entropy of the network features and calculates the threshold dynamically. After getting information about the higher rate of network traffic, the load balancer finds the other routes to disperse the network traffic to avoid congestion at a particular switch.

Further, Bawany et al. [107] proposed an adaptive framework called SEAL (SEcure and AgiLe), which contains three modules, d-defence, a-defence, and c-defence for detection and mitigation of DDoS attacks. Adaptability in SEAL is achieved through a customized version of EWAA (Estimated weighted moving average) filters. For a different type of applications, there are various type of filters to be used, such as proactive filters, active filters, and passive filters. They validate their approach using Mininet network emulator. Ahalawat et al. [108] introduced another method to detect and mitigate DDoS attacks using entropy metric. After detection of an attack, the mitigation module enables the limiting of the inflow of the switches to mitigate its impact. They also validate their approach using Mininet with Ryu controller. Xuanyuan

Table 5
Information theory-based DDoS Defence solutions in SDN.

Sr.No.	Authors years	Scope	Algorithm/ Detection metrics	Parameters	Plane	Controller	Data set	Key points
1	Giotis et al. [97] (2014)	Detection Mitigation	Shannon Entropy	Source IP Destination IP Source Port Destination Port	Application Control	NOX	Traffic Traces from NTU of Athens Synthetic using Scapy and Tcpreplay	– Authors implemented a separate module for data collection. – Data is analysed using the entropy method to detect the anomalies in the network. – Anomaly detection module analysed the data fed by the collector at periodic intervals. – Mitigation module implements flow rules about malicious flows to block them.
2	Wang et al. [98] (2015)	Detection	Shannon Entropy	Destination IP	Data	Floodlight	CAIDA 2007 Synthetic using Scapy	– It reduces flow collection overhead in the controller by implementing the scheme on OpenFlow switches. – It used entropy to detect anomalies in the network.
3	Mousavi et al. [99] (2015)	Detection	Shannon Entropy	Destination IP	Control	POX	Synthetic using Scapy	– The system uses entropy variation of the destination IP address for anomaly detection. – After 50 packets entropy of window will be calculated. – This solution is flexible as parameters can be changed to fit the requirements of Controller.
4	Botie et al. [100] (2017)	Detection Mitigation	Shannon Entropy	Source IP Destination IP Source Port, Destination Port	Application Control Data	RYU	Real Traffic Traces Synthetic using Hping3	– State-based monitoring module used to capture the traffic from switches which rely on stateful-in-switch processing . – An entropy-based statistical approach is used to detect the anomaly in the network. – Then mitigation module installs new flow rules for malicious flows.
5	Tsai et al. [101] (2017)	Detection Mitigation	Shannon Entropy	Destination IP	Control	RYU	Synthetic using Scapy	– In this scheme, an application running on the controller will monitor incoming traffic and calculates the entropy to find deviation in network traffic. – If it found attack, it will help to implement a mitigation strategy to block the specific port.
6	Kalkan et al. [102] (2018)	Detection Mitigation	Joint Entropy	Source IP Destination IP	Control	RYU	Traffic traces from MAWI group	– When congestion is detected, the switch sends only information of the packets to the controller and controller will calculate joint entropy of pair profiles, and it found the difference between entropy values exceeds than threshold, DDoS attack is detected. – Active Mitigation Stage performs five functions such as Suspicious Pair Profile Generation, Score Calculation, Threshold Determination, Rule Generation, and Differing Rules Determination to mitigate the attacks.
7	Sahoo et al. [103] (2018)	Detection	Generalized Entropy	Destination IP	Control	POX	Synthetic using Scapy	– Authors employed information distance as a metric to quantify the deviation of network traffic flows. – They employed two modules in the controller, such as the statistical collection module and anomaly detection module.
8	Sahoo et al. [104] (2018)	Detection	Shannon Entropy GE GID KL-Divergence	Destination IP	Control	POX	Synthetic using Scapy	– Authors used information theory metrics to distinguish flash events from high-rate DDoS attacks

(*continued on next page*)

et al. [109] used conditional entropy for detection and wildcard policy to discard the packets to mitigate the attack. However, wildcard policy is not able to reasonably distinguish between malicious traffic and normal traffic.

Table 5 (*continued*).

Sr.No.	Authors years	Scope	Algorithm/ Detection metrics	Parameters	Plane	Controller	Data set	Key points
9	Jiang et al. [105] 2018	Detection Mitigation	Shannon Entropy	Destination IP	Control	Floodlight	Synthetic	– The value of Shannon entropy of network statistics is compared with the predefined threshold to determine the attack. – Checks IP frequency to avoid dropping of legitimate traffic and blocks only IPs with high frequency.
10	Hong et al. [106] (2019)	Detection Mitigation	Shannon Entropy	Source IP Destination IP Source Port Destination Port	Control	OpenDay-Light	Synthetic using Scapy	– Collector module periodically collects the packets sent by switches and transmit results to the manager module for processing. – A manager is the decision-making module, which decides whether the network is under attack or not. – A third module, Executor, provides instructions to the controller for rule generation as per the direction of the manager module.
11	Bawnay et al. [107] (2019)	Detection Mitigation	Shannon Entropy	Destination IP	Data Control	ONOS	ISCX	– Adaptability achieved through a customized version of Estimated weighted moving average – This pattern is used to identify the attack. – Attack defence module pushes a flow entry to the corresponding OF switch, then the switch drops all packets arriving at victim particular port.
12	Ahalawat et al. [108] 2019	Detection Mitigation	Shannon Entropy	Source IP Destination IP Source Port Destination Port Protocol	Control	Ryu	Synthetic using Scapy and Hping	– A secure controller-to-controller protocol implemented to transfer the attack information with each other. – The mechanism enables the efficient notification along the path of an ongoing attack and effectively filter the traffic near the source of the attack.
13	Xuanyuan et al. [109] 2019	Detection Mitigation	Conditional Entropy	Source IP Destination IP Destination Port Length	Control	POX	LLS 2.0 from MIT Lincoln Lab	– Conditional entropy is used to detect the attack. – Wildcard filtering policy used to drop the packets to mitigate the attack. – The mechanism is not able to reasonably discard attack packets.
14	Cui et al. [110] (2019)	Detection Mitigation	Shannon entropy	Source IP Destination IP	Control	Floodlight	Synthetic using Scapy and D-ITG	– Statistics collection module collects traffic set periodically and sends to the feature extraction module for feature extraction. – Feature extraction module uses entropy to select particularly suitable features for anomaly detection. – Then attack detection module uses this set to classify the traffic as malicious or normal.
15	Li et al. [111] (2020)	Detection	ϕ-entropy	Destination IP	Control	Floodlight	Synthetic using Scapy	– The controller periodically collects the information from the switches and create a hash table of destination IP addresses. – Entropy value is calculated of destination IP addresses using $\phi - entropy$ and compared with pre defined thresholds. If entropy remains below the threshold for consecutive five windows, then DDoS attack is detected.

Cui et al. [110] proposed a solution for detection of DDoS attack using cognitive-inspired computing with dual address entropy. Statistic collection module periodically collects and calculates the frequency of each source and destination IP. Feature Computing module calculates destination entropy and source address entropy. When entropy of destination is lower, and entropy of source is higher than normal traffic threshold, DDoS attack is detected. DDoS attack defence discards all table entries to mitigate the DDoS attack. They validate their approach using a floodlight controller on Mininet.

4.1.2. Machine learning-based DDoS defence solutions in SDN

Machine learning algorithms are used to solve complex problems in many fields [112]. These algorithms are also applied for detection of DDoS attacks, and it has been found that they are better than signature-based detection techniques [113]. These ML-based classifiers can be trained to determine the abnormal behaviour of the network traffic with more accuracy. Some commonly used classifiers based on machine learning are Support Vector Machine (SVM), Hidden Markov Model (HMM), Decision Tree (J48), Advanced Support Vector Machine (SVM), Naive Bayes,

Table 6
Machine Learning based DDoS defence solutions in SDN.

Sr.No.	Authors years	Scope	Algorithm/ Detection metrics	Parameters	Plane	Controller	Data set	Key points
1	Niyaz et al. [114] (2016)	Detection	Stacked Autoencoder	Number of bytes per flow Number of packets per flow	Control	POX	Real Traffic Traces Synthetic using Hping3	– It used deep learning algorithm to derive a reduced set of features from a large set. – It classifies the traffic with an accuracy of 99.82% with very low FPR compared to other works.
2	Hurley et al. [115] (2016)	Detection	HMM	Length of the packet Source port Destination port Source IP Destination IP	Control	Floodlight	Real traffic traces	– This system detects DDoS in the SDN environment using HMM (Hidden Markov Models). – Hidden Markov Models trained using the Baum–Welch algorithm.
3	Alshamrani [116]	Detection	SVM J48 Naive Bayes	subset of best suitable features out of 41 features	Control	POX	NSL	– After inspection, the controller will update, the rule for forwarding or denying. If attack traffic is detected, the ML model and flow tables will be updated as well.
4	Hu et al. [117] (2017)	Detection Mitigation	SVM Shannon entropy (Feature Selection)	Source IP Destination IP Source Port, Destination Port Protocol	Control	POX	Synthetic using TFN2K	– This approach detect and mitigate flooding attacks using entropy and SVM classifier. – Entropy is used to identify changes in network. – DDoS Detection Module performs three task; information collection, feature extraction, and attack detection – Attack mitigation mechanism implemented based on white-list and dynamic updating of forwarding rules.
5	Dehkordi et al. [118] (2017)	Detection	BayesNet J48 Logistic regression RandomTree	Number of Packet_in messages Flow request rate Duration	Control	Floodlight	UNB-ISCX CTU-13 ISOT	– This system compare the current network features' correlation with predefined threshold values for DDoS attack detection. – Machine learning models combined with correlation to increase the accuracy of the existing system.
6	Li et al. [119] (2018)	Detection	Binary Bat algorithm Random forest	Dynamic set of features	Control	–	KDD Cup99	– Authors proposed a two-stage IDS which intelligently detect anomaly in the network by capturing network flows. – Bat Algorithm is used to select features from the network flows. – Random Forest is used for classification.
7	Guozi et al. [120] (2018)	Detection	KNN ϕ-entropy	Average byte Average duration ϕ-entropy of source IP ϕ-entropy of destination IP Increasing speed of flow table entries	Control	–	Synthetic using Hping3 and Nping	– Flow feature extraction module extracts the features using five characteristics from the data provided by the flow table collection module. – KNN classifier is used to classify the current network as an anomaly or normal traffic as per the outcome.
8	Deepa et al. [121] (2019)	Detection	KNN SVM Naive Bayes	Time Difference	Control	POX	CAIDA2016	– The authors proposed an ensemble method to detect anomalous behaviour of data traffic in the SDN controller. – Authors combined KNN-SOM, NV-SOM, and SVM-SOM and found that SVM-SOM produces a higher detection rate and accuracy.

(*continued on next page*)

Logistic regression, Random Trees, Binary Bat algorithm, Random forest, and K-nearest neighbour (KNN). Many fellow researchers used these efficient algorithms for DDoS attack detection, as summarized in Table 6 and discussed below.

Niyaz et al. [114] submitted a multi-vector DoS detection system which is implemented as a network application on the top of an SDN controller. They collected normal data from Home Wireless Network connected to the Internet. To generate attack traffic, they used hping in VMware ESXi host environment to launch different types of DDoS attacks with variable frequency and size. Hurley et al. [115] introduced an intrusion detection system for the SDN environment using HMM (Hidden Markov Models). They set up an experimental environment using Mininet emulator and Floodlight OpenFlow controller. Hidden Markov Models trained using the Baum–Welch algorithm. Alshamrani et al. [116] gave a method to tackle two problems such as Misbehaviour attack and NewFlow attack. Their system periodically collects the information about the network, and then they use a machine learning technique to classify this flow as normal flow or attack flow. They validate their approach using network emulation. Hu et al. [117] proposed flooding attack detection and mitigation system based on entropy and SVM classifier for

Table 6 (*continued*).

Sr.No.	Authors years	Scope	Algorithm/ Detection metrics	Parameters	Plane	Controller	Data set	Key points
9	Phan et al. [122] (2019)	Detection Mitigation	HIPF SVM SOM	Flow Number of source Active of source Average number of packets per flow	Application Control	NFV	CAIDA Synthetic using BoNeSi	– Authors ensembles SVM, eHIPF, and SOM classier to defend the network. – Raw-data Processing Module collects the traffic and extracts features then send it to the classifier module. – Ensembled classifier classifies the data as anomalous or normal. – eHIPF filtering technique is to filter the traffic for mitigation.
10	Myint et al. [123] (2019)	Detection	ASVM	Average number of flow packets Average number of flow bytes Variation of flow Packets variation of flow bytes Average duration	Application	OpenDay-Light	Synthetic using Scapy	– Traffic generation and extraction module collects traffic data and sends it to the classifier module. – Classification module uses Advanced SVM to determine the traffic as normal or attack traffic.

identification of network anomalies. They used network traffic information collected through SDN controller and sFlow agents and introduced mitigation agent to block attack traffic while legitimate users can access the network resources usually. They validate their approach using Mininet emulator.

Dehkordi et al. [118] introduced a hybrid method for DDoS attacks using statistical and machine learning techniques. The analytical method used to extract features and machine learning method used for classification. They validate their approach using UNB-ISCX, CTU-13 and ISOT datasets. Li et al. [119] proposed a two-stage IDS which intelligently detects an anomaly in the network by capturing network flows with a global view. They used Bat algorithm with Swarm Division and Binary Differential Mutation to extract typical features of the network and then Random Forest through adaptability altering the weights of samples using voting mechanism is used as a classifier to classify the network flows. Guozi et al. [120] proposed another hybrid method using ϕ-entropy and KNN machine learning method. ϕ-entropy is to select particular features from a set of features and KNN is used to classify the network traffic. Deepa et al. [121] proposed an ensemble method to detect anomalous behaviour of network traffic in the SDN controller. They used KNN, Naive Bayes, SVM, and self-organizing maps to the ensemble for better efficiency. They validate their approach using Mininet. Authors combined KNN-SOM, NV-SOM, and SVM-SOM and found that SVM-SOM produces a higher detection rate and accuracy. Phan et al. [122] introduced a novel DDoS attack defender using hybrid machine learning technique and enhanced History based IP Filtering (eHIPF) scheme, instead of HIPF, to improve the detection rate and speed to classify the traffic. When eHIPF generate an attack, the mitigation agent sends a flow mod message with drop action to drop every packet at the border of the cloud. To validate their approach, they set up the experimental environment in the laboratory network. Myint et al. [123] implemented a method using advanced support vector machine to detect DDoS attack with minimum overhead. The volumetric and the asymmetric features are used to reduce the training time and testing time. This approach successfully identifies two types of flooding attacks: UDP flood and ICMP flood. They implement the OpenDaylight controller on Mininet to validate their strategy.

4.1.3. Artificial neural network-based DDoS defence solutions in SDN

The Artificial Neural Networks (ANN) have self-learning, self-organization, better fault tolerance and robustness, parallelism as of its advantages, so it got the attention of many researchers. ANN is used by many fellow researchers in detection of DDoS attacks as it can identify not only existing attack patterns but also unknown attack patterns [124]. These techniques improve the intelligence and adaptability of intrusion detection systems (IDS). Self Organizing Maps (SOM), exact-STORM, Back Propagation Neural Network (BPNN), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short Term Memory (LSTM) are some efficient algorithms used by many researchers to detect DDoS attacks in networks based on software-defined networking, as given in Table 7 and are discussed below:

Braga et al. [125] submitted a lightweight approach for DDoS attack detection with very low overhead on performance. Their strategy used flow-based information using traffic flow features. In this approach, the system monitors the NOX switches and extract existing features, such as the average of duration per flow, percentage of pair-flow, growth of single flow, growth of different ports from flow entries of all switches. Then classifier module uses a particular location of winning neuron in the topological map to classify the traffic as normal or attack. The flow collector module collects data periodically from the switches and sends to feature extractor module. Feature extractor module then extracts some essential features which are needed for attack detection and give to classifier module. Finally, the classifier module classifies the traffic flow as an attack or normal using SOM (Self Organizing Maps). They used real datasets for legitimate traffic and synthetically generated data set using Stacheldraht tool for flooding attack. Cui et al. [126] developed SD-Anti DDoS mechanism for the defence of DDoS attacks in software-defined networking. Proposed scheme contains four modules, namely, Attack Detection Trigger, Attack Detection, Attack Traceback, and Attack Mitigation. Attack detection Trigger module implemented the first time to respond more quickly against DDoS attacks. This module reduces the workload of Controller and switches. After detecting the attack, the mitigation module starts blocking the attack traffic as well as it will remove all flows related to fake traffic from the switch flow tables.

Xu et al. [68] proposed a new technique to detect the DDoS attack. This scheme has two phases, such as victim detection and attack detection. After identifying the victim, the system used SOM, neural network techniques, to classify the network traffic as normal or attack. They validate their approach using Internet network topology. Cui et al. [127] extract the temporal behaviour of an attack, and a back-propagation neural network is trained to extract attack pattern then this pattern is used to identify the attack.

When the attack detection module detects the attack, it alters to attack defence module, which blocks the port from where

Table 7
Artificial Neural Network-based DDoS defence solutions in SDN.

Sr.No.	Authors years	Scope	Algorithm/ Detection metrics	Parameters	Plane	Controller	Data set	Key points
1	Braga et al. [125] (2010)	Detection	SOM	Average of Packets per flow Average of Bytes per flow Average of Duration per flow Percentage of Pair-flows Growth of Single-flows Growth of Different Ports	Control	NOX	Synthetic using Stacheldraht	– Classifier module uses a special location of winning neuron in the topological map to classify the traffic as normal or attack. – The system monitors the NOX switches and extract existing features. – Very low overhead on performance.
2	Cui et al. [126] (2016)	Detection Mitigation	exact-STORM	Number of packets per flow Number of bytes per flow Duration Packet rate per flow Byte rate per flow	Control	RYU	Synthetic using D-ITG and TFN2K	– DDoS attack detection trigger module increase the event trigger response rate. – BPNN technique is used to detect the anomaly in the traffic after triggering by attack detection trigger module. – Mitigation module blocks the traffic to mitigate the impact of the ongoing module as per the direction of the detection module.
3	Xu et al. [68] (2016)	Detection	SOM	Packet count per source Byte count per source Packet count asymmetry from source Byte count asymmetry from source	Control	–	Internet network topology	– The victim detection procedure detects the victim using a flow volume feature and the flow rate asymmetry feature. – After detecting the victim, attacker detection procedure uses the SOM based classifier to determine the attack.
4	Cui et al. [127] (2018)	Detection Mitigation	BPNN	Number of Packets per flow Number of flows per port Duration	Control	Floodlight	Synthetic using D-ITG and Scapy	– Authors extract the temporal behaviour of an attack, and a back propagation neural network is trained to extract attack pattern. – This pattern is used to identify the attack. – Attack defence module pushes a flow entry to the corresponding OF switch, then the switch drops all packets arriving at victim particular port.
5	Li et al. [128] (2018)	Detection Mitigation	CNN RNN LSTM	Source port Destination port Source port Destination IP Source IP	Data	–	ISCX 2012	– CNN, RNN, and LSTM neural network models used for attack detection in network. – Deep Learning DDoS Detector module uses a trained deep learning model to detect whether packets entered in the current OpenFlow switch are attack packets. – If so, the attack packet will be forwarded to the Information Statistics module for statistical purposes; otherwise, it will not be processed.
6	Nam et al. [129] (2018)	Detection Mitigation	SOM	Entropy of source IP Entropy of source port Entropy of destination port Entropy of packet protocol Total number of packets	Control	POX	Synthetic using TcpReplay	– Monitor module collects the traffic information from the switches and after processing forward to the Algorithm module. – The Algorithm module classifies the network state as normal or under attack. If the network is under attack, then it generates alert to the mitigation module. – Then mitigation module generates the new policies and forwards these decisions to the switches as well as the server.

(*continued on next page*)

malicious packets are coming. However, it also blocks ports for the legitimate users then port recovery module recovers the ports dynamically. Li et al. [128] suggested a DDoS attack detection mechanism based on deep learning in SDN. This scheme has a feature processing unit which processes the raw data samples and provides data set for the training of deep learning. Based on statistics provided by Information Statistics module, the Flow Table Generator module determines the flow entries and priorities for

Table 7 (*continued*).

Sr.No.	Authors years	Scope	Algorithm/ Detection metrics	Parameters	Plane	Controller	Data set	Key points
7	[130] (2020)	Detection Mitigation	LSTM Fuzzy Logic Shannon Entropy	Source IP Destination IP Source Port Destination Port	Application	Floodlight	CICDDoS 2019 Synthetic using Scapy	– Authors used entropy metric to quantify the network flow features. – LSTM is applied to each flow attribute previously calculated using the entropy metric, and LSTM predicts signature of normal behaviour for each attribute. – Then Fuzzy logic is used to detect anomaly in the network. And EventCondition-Action (ECA) model is used to create dynamic policies for mitigation of DDoS attacks.

various attack packages that can be dropped and send them to the OpenFlow switch. Authors used ISCX data set for training purpose and verified DDoS defence architecture through real-time DDoS attacks. They designed their experimental setup using hardware.

Nam et al. [129] combined the statistical method with a neural network technique to identify the abnormal behaviour of the network. They used the entropy metric to select specific features among a set of features and then used SOM to classify the network behaviour. They validate their approach using emulation with POX controller. Novaes et al. [130] proposed a hybrid scheme for DDoS and Port scan attacks detection and mitigation in SDN networks. For attack detection, this scheme works in three phases, such as characterization, anomaly detection, and mitigation. Authors used entropy metric to quantify the network attribute, and LSTM (Long Short-Term Memory) is then used to model the signature of each attribute of normal traffic. Then they used fuzzy logic to detect the anomaly in the network. They validate their approach using Mininet simulation by implementing floodlight controller.

4.1.4. Other methods

Apart from above-discussed techniques, many researchers used other proficient methods such as SYN cookie algorithm, TRW-CB, Rate Limiting, Graph theory, Queuing Theory, Bloom Filters, and Cumulative Sums etc., for detection and mitigation of DDoS attack in SDN. In Table 8, all techniques based on these methods are discussed below:

Dotcenko et al. [131] introduced an intrusion detection system for software-defined networks using TRW-CM and rate-limiting algorithms. They validate their approach using Beacon controller with implementation on Mininet. Authors introduced an intrusion detection system for software-defined networks using TRW-CM and rate-limiting algorithms. They validate their approach using Beacon controller with implementation on Mininet. Chin et al. [132] proposed attack mitigation and detection approach for TCP SYN flood attacks. They used two modules, monitor and Correlator, for selective inspection of network packets. The monitor is running a real-time communicator algorithm which continuously listens and notifies the associated Correlator about the specific type of attack. The Correlator runs another algorithm which communicates with monitor and OVS. Correlator performs three functions to detect and mitigate the attacks. First, the presence of an attack is to be verified after an alert of a specific attack type is received. Second, once the attack is confirmed, the Correlator performs additional queries to a second hash table created based on the current flow table. Finally, using the OpenFlow API, the Correlator issues a rule directing to drop the rogue traffic. Wang et al. [85] gave a DDoS attack mitigation architecture named DaMask (*D*DoS *a*ttack *m*itigation *a*rchitecture using *s*oftware-defined networ*k*ing). DaMask-D module is responsible for generating attack alerts to the DaMask-M module. Xiao et al. [133] suggested an attack detection framework in SDN to handle the DDoS attacks using Bloom filters. Modules use Bloom filters to collect and detect abnormal flows. They designed a testbed using Mininet to validate their approach.

Aleroud et al. [134] recommended a new approach using graph theory to identify the attack in SDN network. They used an attack signature database extracted from the traditional network data sets. This approach used regular labelled flows to analyse a different sample of OpenFlow and examine whether the existing signature can be used to analyse the flow or not. They claimed that their approach yields better results as compared to [125,143], and [149]. Conti et al. [135] proposed a lightweight approach for DDoS attack detection in SDN context. They used a non-parametric approach, Cumulative Sum, to detect a DDoS attack. They used CAIDA Internet traces as well as DARPA intrusion detection evaluation data set for validation purpose. Kalkan et al. [136] proposed a statistical and packet-based defence mechanism against DDoS attacks in SDN network. In this hybrid mechanism, namely SDNScore, the switches are not only traffic-forward devices, but the devices are also collect statistical information and decide if DDoS attack is in action. After inspecting, these devices cooperate with the controller and act on attack packets. These devices does not drop all packets in the flow but instead filters out attack packets using packet-based analysis.

Bhushan et al. [28] propose a new flow table sharing approach to protect SDN networks from flow table overloading DDoS attacks. Authors proposed to use other switch flow table space with very low communication overhead. Resultantly, the increase in resistance of network against flow table overflows DDoS attacks. A queuing theory-based mathematical model approximates the used and unused flow table space. When the switch is under attack, the approach analyzes the flow table status of all other switches to identify a suitable switch. They proved their approach using network emulator Mininet.

4.2. Review of DDoS mitigation techniques in SDN

DDoS attack mitigation is also a crucial aspect to protect the network resources under attack. Researchers used many techniques such as connection migration, packet migration, limiting inflows bandwidth, adjusting time outs, and controller to controller communication protocols etc. to mitigate the DDoS attack in networks based on Software-Defined networking architecture. We have summarized all mitigation approaches in Table 9 and discussed below as.

Shin et al. [75] proposed a scheme to mitigate saturation attack by extending the capabilities of the existing OpenFlow data plane. They added two modules in Avant-Guard, connection migration module and an actuating Trigger module. Connection migration module can shift failed TCP sessions at data

Table 8
DDoS Defence solutions in SDN based on other techniques.

Sr.No.	Authors years	Scope	Algorithm/ Detection metrics	Parameters	Plane	Controller	Data set	Key points
1	Dotcenko et al. [131] (2014)	Detection	TRW-CM Rate Limit	–	Control	Beacon	Synthetic	– Statistics collected and aggregated at switch using statistic collection module. – TRW-CB network anomaly detection algorithm used to identify malicious traffic.
2	Chin et al. [132] (2015)	Detection Mitigation	Real time communicator	Source port Destination port Source IP Destination IP	Control	POX	Synthetic	– This scheme used two modules, monitor and correlator, for selective inspection of network packets. – The monitor is running a real-time communicator anomaly detection algorithm which continuously listens and notifies the associated correlator about the specific type of attack. – The correlator runs correlation algorithm which communicates with monitor and OVS to mitigate attacks.
3	Wang et al. [85] (2015)	Detection Mitigation	Graph Inference Model Chow–Liu algorithm	Dynamic feature selection	Data	Floodlight	ISCX	– Authors proposed a DDoS attack mitigation architecture named DaMask. – If query results indicate as an attack, DaMask-D module will generate an alert; otherwise, it will forwards the packet to the intended destination. – When the DaMask-M module receives alert, it will find a match for countermeasure otherwise take the default action(forward, drop, and modify).
4	Xiao et al. [133] (2016)	Detection	Bloom Filters	Packet Count Byte Count Duration	Control	Floodlight	Synthetic using Iperf	– Authors proposed an attack detection framework in SDN to handle the DDoS attacks using Bloom filters.
5	Aleroud et al. [134] (2017)	Detection	Graph prediction model Pearson Correlation	Source IP Destination IP Port number Number of packets Protocol type	Control	POX	CAIDA Synthetic using Hping3	– This approach used graph theory to identify the attack in SDN network. – k-NN classifier is used to classify the flow.
6	Conti et al. [135] (2017)	Detection	Cumulative Sum	–	Control	POX	CAIDA DARPA	– Cumulative Sum (non-parametric) approach is used to identify the anomaly. – Value of CuSum for the server is computed periodically and compared with predefined threshold if it exceeds then the attack is identified.
7	Kalkan et al. [136] (2017)	Detection Mitigation	Pre-defined threshold	Source IP Destination IP Source port Destination port Protocol Packet size	Data Control	–	Traffic traces from MAWI Group	– In this hybrid mechanism switches also collect statistical information and decide if DDoS attack is in action. – Switches cooperate with controller to act on attack packets. – The Switches drop only attack packets using packet-based analysis.
8	Wang et al. [137] 2018	Detection Mitigation	Pre-defined threshold	Packet loss rate Round trip time Available bandwidth	Control	Floodlight	Synthetic	– Link target selection module takes the information of flow entries on the switches from the controller and passes to the link congestion monitoring module to detect attack scenario. – Link congestion module uses ABW (Available Bandwidth), packet loss rate and, round trip time to monitor the congestion. – When the attack is detected, mitigating module informs the SDN controller to reroute the traffic or after detecting malicious traffic, blocks the incoming traffic ports.

(*continued on next page*)

plane before notifying to the control plane. The actuating trigger module collects the network status information and packet payload information, and it also activates some flow rule based on condition. They used NetFPGA architecture to illustrate their approach. Wang et al. [138] recommended a lightweight, efficient and protocol-independent framework, called FloodGuard, for the security of SDN networks. Proactive module dynamically derives proactive flow rules by seeing the run-time logic of the SDN controller, which helps to preserve the network policy enforcement. Packet migration module temporarily caches the packets and

Table 8 (*continued*).

Sr.No.	Authors years	Scope	Algorithm/ Detection metrics	Parameters	Plane	Controller	Data set	Key points
9	Bhushan et al. [28] (2019)	Detection Mitigation	Queuing Theory	Flow table space	Data	POX	–	– The Black List database maintains the list of IP addresses of the attack sources. – When the switch is under attack, the approach analyzes the flow table status of all other switches to identify a suitable switch. – A mathematical model, based on queuing theory, is used to calculated the unused spaces of other switches.

submits these packets to the controller using rate limit and Round Robin scheduling so that the controller could be prevented from being overwhelmed. Piedrahita et al. [139] implemented a fast and lightweight scheme, FlowFense, for DDoS attack mitigation. In this approach, the use level of interfaces of routers and SDN controller is measured to detect the congestion state. When a router detects congestion at particular interfaces, it notifies to the controller, and the controller directs to the router to limit the bandwidth of those interfaces. Wang et al. [140] suggested a secure system for access control only after authenticating entities. This scheme has three modules such as authentication & registration & policy management module, access control & communication policy module, and traceback & audit policy module. Any entity wants to communication; first, it has to register with the authentication and registration module, which provides a password for further communication. They implemented all modules on the application layer of the SDN architecture. They validated their approach by implementing a POX controller.

Yuan et al. [141] implemented peers support strategy to mitigate flow table overflow DDoS attacks by combining the available unused memory of the whole SDN system. Their scheme considers all switches at a peer level. When one switch is under attack, other switches will support attacked switch by offering their unused flow table space; consequently, it contributes to mitigating DDoS attack. They used queuing theory to approximate the unused spaces of switches which are not under attack. Dridi et al. [142] suggested a novel scheme, SDN guard, to protect SDN networks against DDoS attacks by dynamically change the route of malicious traffic and adjusting flow time outs. They implemented their approach using Mininet and validate that their approach can minimize up to 32% impact on controller performance. Phan et al. [143] proposed an optimized mechanism called Idle-time Adjustment (IA), based on Support vector machine to combat the problem of flooding attacks. Firstly, the flow collector collects the information from switches, and the extractor then extracts this information. Then SVM-I process corresponding features. Next, based on the output of SVM-I, either the flow sent to the policy enforcement module or IA algorithm. If the output is normal, the IA algorithm will process the flow; otherwise, it sent to policy enforcement to implement a new policy for the same. Sahay et al. [144] introduced an approach call ArOMA, which leverages the centralized manageability and programmability features of the SDN to mitigate the DDoS attack. A controller, residing at ISP end, receive alert and generate a policy for switches to handle the DDoS attack. They implemented a Ryu controller for validating their approach. Hameed et al. [145] proposed a collaborative approach to mitigate the DDoS attacks in SDN. They designed C-to-C (Controller to Controller) protocol which helps SDN controller to communicate and transfers the attack information with each other securely. They implemented the POX controller on Mininet for validation purpose. Conti et al. [146] proposed countermeasure for route spoofing and resource exhaustion DDoS attack in SDN. The Selective Blocking module gathers the data of IP and MAC addresses and notifies the controller for further action. The Periodic Monitoring calculates the entropy of destination IP and destination port to find the information distance to determine a possible abnormal behaviour. They implemented a target scenario on Mininet. Karmakar et al. [147] used northbound application for mitigation of DDoS attacks in SDN. This system used specification and storage of the security policies to deal with DDoS attacks. They implemented the ONOS controller for the validation of their approach. Wang et al. [148] suggested a solution, namely Safe-Guard-Scheme (SGS), to protect the control plane against DDoS attacks. Anomaly detection module uses the BPNN method to find an anomaly in the network traffic. The Controller Remapping operation remaps the flows other controller and access control implement rules to block the hosts who are sending fake traffic.

5. Research challenges and research gaps

There are enormous number of security mechanisms proposed by the researchers in recent times for securing communication networks, but DDoS attacks are continuous to evolve in size, frequency, and sophistication. In this section, we discuss research challenges and research gaps that arises out of its implementation.

5.1. Research challenges

SDN promises to simplify network design and its management but still, many severe research challenges need to be addressed. This section lists such security challenges brought by SDN.

- *Acceptance of SDN Architecture:* SDN architecture needs the deployment of SDN-enable hardware, which adheres the extra cost. Further, experts are needed to design, maintain, and operate the network based on SDN architecture [17]. So the transition from traditional network to SDN based network is a challenge for the network community. Although, problems have been tackled using the integration of SDN and traditional network, however, this transition is also a challenge for the research community. This incremental deployment of the SDN network is a feasible solution [150,151], in which SDN devices are incrementally deployed in the traditional network, and the controller controls only a small portion of the network. However, traffic engineering and the optimization of resource allocation is an open challenge for the community [152].
- *Non-Availability of Production Level Controller:* Non-availability of reliable SDN controllers is another major challenge in the deployment of SDN networks [153]. There is a need of the controller which persists the performance of network similar to the traditional network. Although many open source controllers are available to use, these are not able to handle the fast-growing network [152] which is a challenge for the SDN community.

Table 9
Summary of DDoS Mitigation techniques in SDN.

Sr.No.	Authors years	Scope	Algorithm/ Detection metrics	Parameters	Plane	Controller	Data set	Key points
1	Shin et al. [75] (2013)	Mitigation	SYN cookie algorithm	Packet Count for flow rule Source IP Destination IP	Data	POX	–	– Connection migration module only forwards those traffic flows which complete the handshake at switches. There are four stages in this module, such as classification, report, migration, and relay. – Actuating trigger module collects the network status from the data plane and helps the controller to activate rule generation under predefined conditions.
2	Wang et al. [138] (2015)	Mitigation	–	Rate of packet_in messages Rate of utilization of resources	Control	POX	–	– Authors proposed FloodGuard, a lightweight, efficient and protocol-independent framework. – Proactive module dynamically derives proactive flow rules by seeing the runtime logic of the SDN controller. – Packet migration module temporarily caches the packets and submit these packets to the controller using rate limit.
3	Piedrahita et al. [139] (2015)	Mitigation	Flow transmitted bit rate	Source IP Destination IP Flow length Flow duration	Data	POX	Synthetic	– In this scheme, the use levels of interfaces of routers and SDN controller are measured to detect the congestion state. – When a router detects congestion at particular interfaces, it notifies to the controller. The controller directs to the router to limit the bandwidth of those interfaces.
4	Wang et al. [140] (2015)	Mitigation	Access Control Policies	Source MAC Destination MAC Source IP Destination IP Source Port Destination Port	Application	POX	DARPA 2000 Synthetic using TFN2K	– Authors implemented access control policies, whenever any entity wants to communicate the network, it has to register with an access control module. – After authentication, this module provides a password for further communication and updates the data paths for registered entities and flow table of switches.
5	Yuan et al. [141] (2016)	Mitigation	Queuing theory	Flow table space	Control	POX	Synthetic	– Status monitor module checks the space in the switch's table when a new rule is requested by the switch from the controller. – Traffic guiding module stores the rule in the switch table if it is not full as indicated by the status monitor module. If table memory is full, then it will store the flow rule in other switches memory.
6	Dridi et al. [142] (2016)	Mitigation	Pre-defined threshold	Bandwidth consumption Queuing delays	Control	Floodlight	Synthetic using Hping3	– This scheme dynamically change the route of malicious traffic and adjusting flow time outs. – This scheme is implemented as an SDN application on top of the controller, having the components flow management module, rule aggregation module, and monitoring module. – This scheme assigns larger time outs to flows to decrease the bandwidth utilization between switch and controller.

(*continued on next page*)

- *Scalability of Controller:* With the rapid growth of network size, scalability of the controller due to computation limit is another severe challenge [15]. A multi-controller platform [29,154,155] have been devised to enhance the reliability and scalability. In the distributed controller environment, controllers are logically centralized where one controller is root controller having the global view of the network, and others are local controllers having information of network in their domain only. However, synchronization of these controllers is another issue to be addressed.
- *Security of OpenFlow Switches:* The attackers can use various network devices to launch the DDoS attacks against OpenFlow switches [86]. The attackers could use hijacked switches to slow down the network or drop the legitimate traffic flows. These compromised switches can also overwhelm the controller by sending fake traffic packets towards the controller for rule generation. So, these switches could become a bottleneck for the whole network. Therefore, it is a severe challenge for the research community to secure the switches from compromising.

Table 9 (*continued*).

Sr.No.	Authors years	Scope	Algorithm/ Detection metrics	Parameters	Plane	Controller	Data set	Key points
7	Phan et al. [143] (2016)	Mitigation	SVM	Number of packets per flow Duration	Control	POX	Synthetic using BoNeSi	– Authors proposed an optimized mechanism called Idle-time Adjustment (IA), based on Support vector machine to combat the problem of flooding attacks. – Policy Enforcement Module apply with drop action for attack flows which drop all flows with large number of flows sent by attacker. – If the output is normal, the IA algorithm will process the flow; otherwise, it sent to policy enforcement to implement a new policy for the same.
8	Sahay et al. [144] (2017)	Mitigation	–	-	Control Data	Ryu	Synthetic using BoNeSi	– Flow statistics are collected periodically, and an external database is used to detect attack detection to generate attack alert. – When controller, which resides at ISP end, receives alert for mitigation, it makes policy to take mitigating actions such as redirecting traffic.
9	Hameed et al. [145] (2018)	Mitigation	Controller-to-Controller Protocol	–	Control	POX	Synthetic using Scapy	– A secure controller-to-controller protocol implemented to transfer the attack information with each other. – The mechanism enables the efficient notification along the path of an ongoing attack and effectively filter the traffic near the source of the attack.
10	Conti et al. [146] (2019)	Mitigation	Shannon Entropy	Destination IP Destination Port	Data Control	OpenDay-Light	Synthetic using Hping3	– Selective-blocking module is installed on OF-Switches to check the incoming traffic according to implemented policies. Suspected traffic messages will be forwarded to the controller for further inspection. – Periodic-monitoring module is also installed on switches to determine the anomaly using randomness of the incoming traffic.
11	Karmakar et al. [147] (2019)	Mitigation	–	-	Control	ONOS	Synthetic using Javasnoop and Metasploit	– The security application has two types of policy expressions, such as secure routing policy expressions and intrusion detection policy expressions (IDPE). – The application continuously monitors the packets, and if packet attributes match with IDPE, it drops the packets and installs new rule to block the traffic to mitigate the attack.
12	Wang et al. [148] (2019)	Mitigation	BPNN	Byte rate Symmetric flows percentage Variation rate of asymmetric flows Flows percentage with small amount packets	Data Control	RYU	Synthetic using Hping3	– This scheme protects the control plane against DDoS attacks. Anomaly detection module uses the BPNN method to find an anomaly in the network traffic. – The Controller Remapping operation remaps the flows to other controller and access control implement rules to block the hosts who are sending fake traffic

- *Security of Communication Link:* Breaches in the communication links between the switches and controllers could cause to degradation of network performance. Enhanced and protected channels can benefit to controllers and switches. More efforts are needed to develop and implement the security on these communication links. In SDN based networks, switches rely on controllers for forwarding rule for each inflow traffic, therefore for every mismatched entry, switch asks from the controller to modify or creation of new forwarding rule. So these communication links play a very crucial role to achieve the desired level of performance. There is a need of a more robust mechanism for authentication to defend the man-in-middle attacks [156]. Security of these links is also a very big challenge for the SDN community.
- *Legitimate or Illegitimate Traffic:* Many defence solutions for DDoS attack detection generate alerts by comparing the parametric values with pre-defined threshold values. Therefore, if any attack (Low-Rate DDoS attack) mimic the behaviour of normal traffic, then it will remain undetected [6]. At the same time, if the calculated value of current flow crosses the threshold value, then the mitigation module starts blocking the traffic. The mitigation module is not able to distinguish between legitimate or malicious traffic, resultantly, decreases the performance of the network. So there is a need for a valid and reliable security solution which could efficiently distinguish normal and malicious network traffic flows.
- *Unauthenticated SDN applications:* SDN architecture is divided into three different planes such as data plane, control

plane, and application plane, for efficient functionality. In the application plane, there are many applications which gain access to network resources for better management. Alongside, many other unauthorized applications gain this access using instances of authorized applications [71]. These applications may cause to compromise of network devices which lead to degradation of network performance. So there is a need for a stable application-authentication module to verify the application before permitting the access to network resources.

- *Single Point of Failure:* For improved control and management of the network, there is a centralized control plane in the SDN architecture. However, this centralization is vulnerable to network security due to lack of stable and secure controller platform [6]. In the SDN network, the controller is the only responsible for policy creation and management for the new flows coming in the network. If the centralized controller compromised by the attackers, they can have power to down the whole network. Development of a secure and robust controller is also an open challenge for the research community.
- *Discrimination of Flash Events:* When a large number of legitimate users simultaneously access the network resources, web services, or computing resources of the server, it leads to downgrade the performance of the network. This traffic scenario is known as Flash Event (FE) [157]. This scenario is very similar to DDoS attacks, and however, in FE, there are benign users instead of fake users. The situation may worst when High-Rate DDoS attack launched during the flash event. Although many authors discriminate flash events in traditional IP networks [158], yet in SDN, there are not many efforts have been done for discrimination of flash events from the DDoS attacks.

5.2. Research gaps

After the extensive review of existing DDoS defence solutions in SDN, we also found a number of research gaps. If explored, they could further help in the development and deployment of a secure solutions for SDN.

- We found that majority of the researchers such as [106–109,111,121,122] etc., have used a single controller (e.g. POX, NOX, Floodlight, and OpenDayLight etc.) in their simulation environments. However, due to the non-availability of secure and stable controller [6], this single controller could be a point of failure of the whole network. On the other hand, implementation of multiple controllers and distributed defence solution could be a better choice for future defence schemes as it could distribute overhead among various machines and load-balancing could be achieved as per requirements. In future, topologies with multiple controllers could be implemented and distributed DDoS defence mechanisms could be deployed while keeping the synchronization and communication overheads at a minimum.
- Many researchers such as [28,75,85,98,107,128,136,139,146], propose to enhance the functionality of SDN switches by embedding advanced security modules. Although the intelligence in switches reduces the computation overhead of controller and communication overhead between data plane and control plane, but it may increase the complexity of devices and incur an extra cost. Therefore, an efficient deployment of security modules in switches while minimizing the communication complexity between devices is an open issue.
- Majority of the fellow researchers, [99–101,103–105], use virtual environments to validate their defence approaches through Mininet emulator. This virtualization of network devices and links between devices affect the results drastically because modelling of Internet behaviour using simulation tools is very difficult. There is no known formula to represent Internet behaviour till date [159]. Moreover, fellow researchers used a single machine to set up their proposed experimental setups which is also a barrier in the validation of security approaches. So, involving multiple physical machines for the validation of proposed security approaches is another open issue.
- The solutions based on Information theory metrics such as [97,99,100,102–106], use predefined threshold values (depending on baseline network behaviour) for anomaly detection. Since SDN based networks are not deployed publicly yet, so to determine the correct baseline behaviour of SDN based networks is a challenge in front of the research community. Besides this, the non-availability of benchmarked data sets to represent normal and attack traffic is another open issue. However, researchers used some traffic generator tools to model normal traffic; it could not reflect today's high-speed network. These tools are not able to generate an appropriate proportion of background, normal and attack traffic [158]. So, the prediction of correct baseline network behaviour is also a research gap in the existing solutions.
- The solutions based on machine learning techniques such as [114–120], need normal traffic datasets to train the machine for DDoS detection. Non-availability of real normal traffic dataset of SDN network is also a barrier for these techniques. Fellow researchers used synthetically generated data sets to train the machines, which are biased data sets as tools are not able to generate mixed datasets as real ones. Training of machine with correct normal behaviour is also a research gap in existing work.
- Many researchers [97–99,101] have used deviations in current network behaviour and normal traffic behaviour for detection of DDoS attacks. They assumed that variations should be high enough for accurate detection of DDoS attacks, resultantly their approaches can determine only high-rate DDoS attacks. However, attackers can generate sophisticated low-rate DDoS attacks to elude such defence mechanisms. So, the detection of low-rate DDoS attacks is also a gap in the current work.
- DDoS detection and mitigation solutions based on Artificial Neural Networks (ANN) [68,125–127,129] have used multiple parameters for calculation of the current state of the network. Most of these security solutions implemented on the control plane of SDN architecture. A centralized controller is only responsible for policymaking; it can serve up to 20 000 new flow requests if it is managing 150 switches [160], so a controller has to work a lot to handle the network traffic efficiently. Still, this extra calculation overhead of multiple parameters can compromise the controller performance. So, the number of parameters used for DDoS detection could be reduced in ANN-based methods.
- Only a few authors like Jiang et al. [105] try to distinguish the flash event traffic from behaviourally similar DDoS attack traffic. Moreover, they used a small network topology i.e. 11 hosts only, which make their validation technique unrealistic.
- Wang et al. [148] used multiple controllers in master–slave architecture to validate their defence proposal. Although, they proved their approach is better in network performance, but they did not consider the overhead of synchronization of multiple controllers. Efficient synchronization is also a research gap to be considered.

- Most of the security solutions such as [68,125,126,131,139] use network traffic features for detecting DDoS attacks in SDN. These methods use native OpenFlow statistics collection techniques for the collection of network features from the data plane. However, the collection of network statistics using OpenFlow induces much processing overhead for centralized control plane in large scale networks [117]. Some authors [97,161] used the sFlow technique, which separates the network statistics collection process from forwarding logic of switches, to overcome the problem of overhead. However, sFlow gathers partial information which affects the accuracy of the defence solution. So, the collection of network statistics with minimum overhead is also a research gap in existing work.
- Some authors [103,104,111,120] used generalized information Entropy measures such as Renyi's Entropy and ϕ-Entropy for detecting DDoS attacks in SDN. These generalized information theory metrics produce better results as compared to traditional Shannon Entropy measure but the optimal selection of entropic index parameter α value is very challenging.
- Many authors [75,139,140,142,145–147] have used a single SDN controller to validate their mitigation approaches. But under DDoS attacks, the central controller itself become vulnerable. However, there is a need to use multiple SDN controllers for implementing realistic distributed network topologies to simulate real world scenario.

6. Conclusion and future directions

The rapid growth of Internet-based services and applications lead to an increase in the number of threats to its existence. Many practical solutions have been devised in the past to secure network infrastructure. SDN is one of the most reliable and robust solutions in combating network-wide issues of flexibility, management, and adaptability for sustainable growth of the Internet and its services. SDN separates the functionality of a network into three layers which simplify the network management and brings the innovation in the network security domain. However, this new paradigm is also vulnerable to DDoS attacks which make its deployment very challenging.

This paper provides a comprehensive overview of SDN layered architecture along with its strengths to combat the problem of DDoS attacks and its vulnerabilities which lead to new DDoS attacks instead of conventional DDoS attacks. This paper has reviewed around 70 of such prominent research articles. We found that around 47% researchers have used information theory-based methods, about 42% have used machine learning-based methods, and approximately 20% have used Artificial Neural network-based methods to detect DDoS attacks in SDN.

Besides this, we have listed out technical aspects of various security mechanisms that will help the fellow researchers to understand state-of-the-art. The controller remains the primary target for the attackers because the controller is a most responsible component of the SDN enable network. The research challenges such as acceptance of SDN architecture, non-availability of production level controller, scalability, security of SDN switches and communication link, dependency on central controller, etc. continue to co-exist which makes the implementation of a secure SDN an open issue.

As a part of future work, we are motivated to develop a distributed DDoS detection and mitigation framework using generalized information theory metric to reduce the overall overhead of a single controller in SDN. Further, characterization of flash events from similar-looking high-rate DDoS attacks is also going to be a challenging area of research.

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.

Acknowledgements

Jagdeep Singh is a Research Scholar in Computer Science and Engineering department at Shaheed Bhagat Singh State Technical Campus, Ferozepur, Punjab, India. The authors acknowledges the support and facilities provided by the department of Computer Science and Engineering at Shaheed Bhagat Singh State Technical Campus, India, Ferozepur and I.K.G. Punjab Technical University, India, Kapurthala.

References

[1] Internet growth usage statistics, 2019, https://www.clickz.com/internet-growth-usage-stats-2019-time-online-devices-users/235102/.
[2] DoS attack report, 2020, https://www.britannica.com/technology/denial-of-service-attack.
[3] M. Feily, A. Shahrestani, S. Ramadass, A survey of botnet and botnet detection, in: 2009 Third International Conference on Emerging Security Information, Systems and Technologies, IEEE, 2009, pp. 268–273.
[4] M. Abu Rajab, J. Zarfoss, F. Monrose, A. Terzis, A multifaceted approach to understanding the botnet phenomenon, in: Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement, 2006, pp. 41–52.
[5] B. Saha, A. Gairola, Botnet: an overview, CERT-In White Paper, CIWP-2005-05, Vol. 240, 2005.
[6] N.Z. Bawany, J.A. Shamsi, K. Salah, DDoS attack detection and mitigation using SDN: methods, practices, and solutions, Arab. J. Sci. Eng. 42 (2) (2017) 425–441.
[7] M.M. Joëlle, Y.-H. Park, Strategies for detecting and mitigating DDoS attacks in SDN: A survey, J. Intell. Fuzzy Systems 35 (6) (2018) 5913–5925.
[8] S. Dong, K. Abbas, R. Jain, A survey on distributed denial of service (DDoS) attacks in SDN and cloud computing environments, IEEE Access 7 (2019) 80813–80828.
[9] A.P. Fajar, T.W. Purboyo, A survey paper of distributed denial-of-service attack in software defined networking (sdn), Int. J. Appl. Eng. Res. 13 (1) (2018) 476–482.
[10] X. Xu, H. Yu, K. Yang, DDoS attack in software defined networks: a survey, ZTE Commun. 15 (3) (2017).
[11] K. Kalkan, G. Gur, F. Alagoz, Defense mechanisms against DDoS attacks in SDN environment, IEEE Commun. Mag. 55 (9) (2017) 175–179.
[12] M.P. Singh, A. Bhandari, New-flow based DDoS attacks in SDN: Taxonomy, rationales, and research challenges, Comput. Commun. (2020).
[13] C. Douligeris, D.N. Serpanos, Network Security: Current Status and Future Directions, John Wiley & Sons, 2007.
[14] B. Mukherjee, L.T. Heberlein, K.N. Levitt, Network intrusion detection, IEEE Netw. 8 (3) (1994) 26–41.
[15] D. Kreutz, F.M. Ramos, P.E. Verissimo, C.E. Rothenberg, S. Azodolmolky, S. Uhlig, Software-defined networking: A comprehensive survey, Proc. IEEE 103 (1) (2014) 14–76.
[16] T. Benson, A. Akella, D.A. Maltz, Unraveling the complexity of network management, in: NSDI, 2009, pp. 335–348.
[17] W. Xia, Y. Wen, C.H. Foh, D. Niyato, H. Xie, A survey on software-defined networking, IEEE Commun. Surv. Tutor. 17 (1) (2014) 27–51.
[18] J. Pan, S. Paul, R. Jain, A survey of the research on future internet architectures, IEEE Commun. Mag. 49 (7) (2011) 26–36.
[19] L. Popa, A. Ghodsi, I. Stoica, HTTP as the narrow waist of the future Internet, in: Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks, 2010, pp. 1–6.
[20] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J.D. Thornton, D.K. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos, et al., Named data networking (ndn) project, in: Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC, Vol. 157, Citeseer, 2010, p. 158.
[21] A.T. Campbell, H.G. De Meer, M.E. Kounavis, K. Miki, J.B. Vicente, D. Villela, A survey of programmable networks, ACM SIGCOMM Comput. Commun. Rev. 29 (2) (1999) 7–23.
[22] O.N. Fundation, Software-defined networking: The new norm for networks, ONF White Paper, Vol. 2, pp. 2–6.
[23] I. Ahmad, S. Namal, M. Ylianttila, A. Gurtov, Security in software defined networks: A survey, IEEE Commun. Surv. Tutor. 17 (4) (2015) 2317–2346.

[24] S. Shin, G. Gu, Attacking software-defined networks: A first feasibility study, in: Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, 2013, pp. 165–166.
[25] P. Fonseca, R. Bennesby, E. Mota, A. Passito, A replication component for resilient OpenFlow-based networking, in: 2012 IEEE Network Operations and Management Symposium, IEEE, 2012, pp. 933–939.
[26] S. Scott-Hayward, S. Natarajan, S. Sezer, A survey of security in software defined networks, IEEE Commun. Surv. Tutor. 18 (1) (2015) 623–654.
[27] S.T. Ali, V. Sivaraman, A. Radford, S. Jha, A survey of securing networks using software defined networking, IEEE Trans. Reliab. 64 (3) (2015) 1086–1097.
[28] K. Bhushan, B.B. Gupta, Distributed denial of service (DDoS) attack mitigation in software defined network (SDN)-based cloud computing environment, J. Ambient Intell. Humaniz. Comput. 10 (5) (2019) 1985–1997.
[29] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama, et al., Onix: A distributed control platform for large-scale production networks, in: OSDI, Vol. 10, 2010, pp. 1–6.
[30] OpenFlow switch, 2020, https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf. (Accessed on 11 March 2020).
[31] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, OpenFlow: enabling innovation in campus networks, ACM SIGCOMM Comput. Commun. Rev. 38 (2) (2008) 69–74.
[32] Open networking foundation, 2020, https://www.opennetworking.org.
[33] A. Lara, A. Kolasani, B. Ramamurthy, Network innovation using openflow: A survey, IEEE Commun. Surv. Tutor. 16 (1) (2013) 493–512.
[34] B.A.A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, T. Turletti, A survey of software-defined networking: Past, present, and future of programmable networks, IEEE Commun. Surv. Tutor. 16 (3) (2014) 1617–1634.
[35] Y. Jarraya, T. Madi, M. Debbabi, A survey and a layered taxonomy of software-defined networking, IEEE Commun. Surv. Tutor. 16 (4) (2014) 1955–1980.
[36] R. Presuhn, J. Case, K. McCloghrie, M. Rose, S. Waldbusser, Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP), Tech. Rep., 2002, STD 62, RFC 3416, December.
[37] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, J. Wilcox, Intelligent design enables architectural evolution, in: Proceedings of the 10th ACM Workshop on Hot Topics in Networks, 2011, pp. 1–6.
[38] B. Raghavan, M. Casado, T. Koponen, S. Ratnasamy, A. Ghodsi, S. Shenker, Software-defined internet architecture: decoupling architecture from infrastructure, in: Proceedings of the 11th ACM Workshop on Hot Topics in Networks, 2012, pp. 43–48.
[39] H. Kim, N. Feamster, Improving network management with software defined networking, IEEE Commun. Mag. 51 (2) (2013) 114–119.
[40] J. Sherry, S. Ratnasamy, J.S. At, A Survey of Enterprise Middlebox Deployments, Technical Report No. UCB/EECS-2012-24, Citeseer, 2012.
[41] Technical Report on SDN, 2019, http://www2.technologyreview.com/article/412194/tr10-software-defined-networking/.
[42] H. Jamjoom, D. Williams, U. Sharma, Don't call them middleboxes, call them middlepipes, in: Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, 2014, pp. 19–24.
[43] S. Shenker, M. Casado, T. Koponen, N. McKeown, et al., The future of networking, and the past of protocols, Open Networking Summit, Vol. 20, 2011, pp. 1–30.
[44] H. Alkhatib, P. Faraboschi, E. Frachtenberg, H. Kasahara, D. Lange, P. Laplante, A. Merchant, D. Milojicic, K. Schwan, IEEE CS 2022 Report (Draft), Tech. Rep., IEEE Computer Society, 2014.
[45] S. Scott-Hayward, G. O'Callaghan, S. Sezer, SDN security: A survey, in: 2013 IEEE SDN for Future Networks and Services, SDN4FNS, IEEE, 2013, pp. 1–7.
[46] A. Doria, J.H. Salim, R. Haas, H.M. Khosravi, W. Wang, L. Dong, R. Gopal, J.M. Halpern, Forwarding and control element separation (ForCES) protocol specification, RFC 5810 (2010) 1–124.
[47] A. Tewari, B. Gupta, Security, privacy and trust of different layers in Internet-of-Things (IoTs) framework, Future Gener. Comput. Syst. (2018).
[48] H. Song, Protocol-oblivious forwarding: Unleash the power of SDN through a future-proof forwarding plane, in: Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, 2013, pp. 127–132.
[49] T. Ubale, A.K. Jain, Survey on DDoS attack techniques and solutions in software-defined network, in: Handbook of Computer Networks and Cyber Security, Springer, 2020, pp. 389–419.
[50] Nox controller, 2020, https://github.com/noxrepo/nox. (Accessed on 11 March 2020).
[51] Pox controller, 2020, https://github.com/noxrepo/pox. (Accessed on 11 March 2020).
[52] Project floodlight, 2020, http://www.projectfloodlight.org/floodlight/. (Accessed on 11 March 2020).
[53] Ryu, 2020, https://osrg.github.io/ryu/. (Accessed on 11 March 2020).
[54] S. Khan, A. Gani, A.W.A. Wahab, A. Abdelaziz, M.A. Bagiwa, FML: A novel forensics management layer for software defined networks, in: 2016 6th International Conference-Cloud System and Big Data Engineering, Confluence, IEEE, 2016, pp. 619–623.
[55] A. Voellmy, H. Kim, N. Feamster, Procera: a language for high-level reactive network control, in: Proceedings of the First Workshop on Hot Topics in Software Defined Networks, 2012, pp. 43–48.
[56] C.J. Anderson, N. Foster, A. Guha, J.-B. Jeannin, D. Kozen, C. Schlesinger, D. Walker, NetKAT: Semantic foundations for networks, ACM SIGPLAN Not. 49 (1) (2014) 113–126.
[57] N. Foster, R. Harrison, M.J. Freedman, C. Monsanto, J. Rexford, A. Story, D. Walker, Frenetic: A network programming language, ACM SIGPLAN Not. 46 (9) (2011) 279–291.
[58] A. Tootoonchian, Y. Ganjali, Hyperflow: A distributed control plane for openflow, in: Proceedings of the 2010 Internet Network Management Conference on Research on Enterprise Networking, Vol. 3, 2010.
[59] Opendaylight user guide, 2020, https://docs.opendaylight.org/en/stable-fluorine/user-guide/alto-user-guide.html. (Accessed on 11 March 2020).
[60] H. Uppal, D. Brandon, OpenFlow Based Load Balancing, CSE561: Networking Project Report, University of Washington, Citeseer, 2010.
[61] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, S. Shenker, NOX: towards an operating system for networks, ACM SIGCOMM Comput. Commun. Rev. 38 (3) (2008) 105–110.
[62] K. Dhamecha, B. Trivedi, Sdn issues-a survey, Int. J. Comput. Appl. 73 (18) (2013).
[63] A. Voellmy, P. Hudak, Nettle: Taking the sting out of programming network routers, in: International Symposium on Practical Aspects of Declarative Languages, Springer, 2011, pp. 235–249.
[64] W. Stallings, Software-defined networks and openflow, Internet Protocol J. 16 (1) (2013) 2–14.
[65] F. Hu, Q. Hao, K. Bao, A survey on software-defined network and openflow: From concept to implementation, IEEE Commun. Surv. Tutor. 16 (4) (2014) 2181–2206.
[66] P. Manso, J. Moura, C. Serrão, SDN-based intrusion detection system for early detection and mitigation of DDoS attacks, Information 10 (3) (2019) 106.
[67] J. Zheng, Q. Li, G. Gu, J. Cao, D.K. Yau, J. Wu, Realtime DDoS defense using COTS SDN switches via adaptive correlation analysis, IEEE Trans. Inf. Forensics Secur. 13 (7) (2018) 1838–1853.
[68] Y. Xu, Y. Liu, DDoS attack detection under SDN context, in: IEEE INFOCOM 2016-the 35th Annual IEEE International Conference on Computer Communications, IEEE, 2016, pp. 1–9.
[69] Z. Liu, R.H. Campbell, M. Mickunas, Active security support for active networks, IEEE Trans. Syst. Man Cybern. C Appl. Rev. 33 (4) (2003) 432–445.
[70] S.W. Shin, P. Porras, V. Yegneswaran, G. Gu, A framework for integrating security services into software-defined networks, in: Open Networking Summit, Open Networking Summit, 2013.
[71] X. Wen, Y. Chen, C. Hu, C. Shi, Y. Wang, Towards a secure controller platform for openflow applications, in: Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, 2013, pp. 171–172.
[72] S. Hartman, M. Wasserman, D. Zhang, Security requirements in the software defined networking model, 2013, Internet Engineering Task Force, Internet-Draft draft-hartman-sdnsec-requirements-01.
[73] H. Xie, T. Tsou, D. Lopez, H. Yin, V. Gurbani, Use cases for ALTO with software defined networks, 2012, Working Draft, IETF Secretariat, Internet-Draft draft-xie-alto-sdn-extension-use-cases-01. txt.
[74] J. Naous, D. Erickson, G.A. Covington, G. Appenzeller, N. McKeown, Implementing an OpenFlow switch on the NetFPGA platform, in: Proceedings of the 4th ACM/IEEE Symposium on Architectures for Networking and Communications Systems, 2008, pp. 1–9.
[75] S. Shin, V. Yegneswaran, P. Porras, G. Gu, Avant-guard: Scalable and vigilant switch flow management in software-defined networks, in: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, 2013, pp. 413–424.
[76] G. Yao, J. Bi, L. Guo, On the cascading failures of multi-controllers in software defined networks, in: 2013 21st IEEE International Conference on Network Protocols, ICNP, IEEE, 2013, pp. 1–2.
[77] Crippling cyber-attacks, 1998, https://www.bbc.com/news/technology-35376327. 25/3/2019, (Accessed on 13 February 2020).
[78] N.Z. Bawany, J.A. Shamsi, Application layer DDoS attack defense framework for smart city using SDN, in: The Third International Conference on Computer Science, Computer Engineering, and Social Media, CSCESM2016, 2016, p. 1.
[79] S. Jajodia, K. Kant, P. Samarati, A. Singhal, V. Swarup, C. Wang, Secure Cloud Computing, Springer, 2014.

[80] S. Bu, F.R. Yu, X.P. Liu, H. Tang, Structural results for combined continuous user authentication and intrusion detection in high security mobile ad-hoc networks, IEEE Trans. Wireless Commun. 10 (9) (2011) 3064–3073.
[81] S. Sezer, S. Scott-Hayward, P.K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, N. Rao, Are we ready for SDN? Implementation challenges for software-defined networks, IEEE Commun. Mag. 51 (7) (2013) 36–43.
[82] A. Wang, Y. Guo, F. Hao, T. Lakshman, S. Chen, Scotch: Elastically scaling up sdn control-plane using vswitch based overlay, in: Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies, 2014, pp. 403–414.
[83] Q. Yan, F. Yu, Q. Gong, J. Li, Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges, IEEE Commun. Surv. Tutor. 18 (1) (2015) 602–622.
[84] T. Ubale, A.K. Jain, Taxonomy of DDoS attacks in software-defined networking environment, in: International Conference on Futuristic Trends in Network and Communication Technologies, Springer, 2018, pp. 278–291.
[85] B. Wang, Y. Zheng, W. Lou, Y. Hou, DDoS attack protection in the era of cloud computing and software-defined networking, Comput. Netw. 81 (2015) 308–319.
[86] D. Kreutz, F.M. Ramos, P. Verissimo, Towards secure and dependable software-defined networks, in: Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, 2013, pp. 55–60.
[87] L. Schehlmann, S. Abt, H. Baier, Blessing or curse? Revisiting security aspects of software-defined networking, in: 10th International Conference on Network and Service Management (CNSM) and Workshop, IEEE, 2014, pp. 382–387.
[88] Open Networking Specifications 1.5.1, Vol. 3, Open Networking Foundation, 2015.
[89] E. Spitznagel, D. Taylor, J. Turner, Packet classification using extended TCAMs, in: 11th IEEE International Conference on Network Protocols, 2003. Proceedings, IEEE, 2003, pp. 120–131.
[90] M. Parashar, A. Poonia, K. Satish, A survey of attacks and their mitigations in software defined networks, in: 2019 10th International Conference on Computing, Communication and Networking Technologies, ICCCNT, IEEE, 2019, pp. 1–8.
[91] A. Akhunzada, E. Ahmed, A. Gani, M.K. Khan, M. Imran, S. Guizani, Securing software defined networks: taxonomy, requirements, and open issues, IEEE Commun. Mag. 53 (4) (2015) 36–44.
[92] J.M. Dover, A Denial of Service Attack Against the Open Floodlight SDN Controller, Tech. Rep., Dover Networks, 2013.
[93] R. Kandoi, M. Antikainen, Denial-of-service attacks in OpenFlow SDN networks, in: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), IEEE, 2015, pp. 1322–1326.
[94] P. Zhang, H. Wang, C. Hu, C. Lin, On denial of service attacks in software defined networks, IEEE Netw. 30 (6) (2016) 28–33.
[95] C.E. Shannon, A mathematical theory of communication, Bell Syst. Tech. J. 27 (1948) 379–423.
[96] C.H. Bennett, P. Gács, M. Li, P.M. Vitányi, W.H. Zurek, Information distance, IEEE Trans. Inform. Theory 44 (4) (1998) 1407–1423.
[97] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, V. Maglaris, Combining openFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments, Comput. Netw. 62 (2014) 122–136.
[98] R. Wang, Z. Jia, L. Ju, An entropy-based distributed DDoS detection mechanism in software-defined networking, in: 2015 IEEE Trustcom/BigDataSE/ISPA, Vol. 1, IEEE, 2015, pp. 310–317.
[99] S.M. Mousavi, M. St-Hilaire, Early detection of DDoS attacks against SDN controllers, in: 2015 International Conference on Computing, Networking and Communications, ICNC, IEEE, 2015, pp. 77–81.
[100] J. Boite, P.-A. Nardin, F. Rebecchi, M. Bouet, V. Conan, Statesec: Stateful monitoring for DDoS protection in software defined networks, in: 2017 IEEE Conference on Network Softwarization, NetSoft, IEEE, 2017, pp. 1–9.
[101] S.-C. Tsai, I.-H. Liu, C.-T. Lu, C.-H. Chang, J.-S. Li, Defending cloud computing environment against the challenge of DDoS attacks based on software defined network, in: Advances in Intelligent Information Hiding and Multimedia Signal Processing, Springer, 2017, pp. 285–292.
[102] K. Kalkan, L. Altay, G. Gür, F. Alagöz, JESS: Joint entropy-based DDoS defense scheme in SDN, IEEE J. Sel. Areas Commun. 36 (10) (2018) 2358–2372.
[103] K.S. Sahoo, D. Puthal, M. Tiwary, J.J. Rodrigues, B. Sahoo, R. Dash, An early detection of low rate DDoS attack to SDN based data center networks using information distance metrics, Future Gener. Comput. Syst. 89 (2018) 685–697.
[104] K.S. Sahoo, M. Tiwary, B. Sahoo, Detection of high rate DDoS attack from flash events using information metrics in software defined networks, in: 2018 10th International Conference on Communication Systems & Networks, COMSNETS, IEEE, 2018, pp. 421–424.
[105] Y. Jiang, X. Zhang, Q. Zhou, Z. Cheng, An entropy-based DDoS defense mechanism in software defined networks, in: International Conference on Communicatins and Networking in China, Springer, 2016, pp. 169–178.
[106] G.-C. Hong, C.-N. Lee, M.-F. Lee, Dynamic threshold for DDoS mitigation in SDN environment, in: 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC, IEEE, 2019, pp. 1–7.
[107] N.Z. Bawany, J.A. Shamsi, SEAL: SDN based secure and agile framework for protecting smart city applications from DDoS attacks, J. Netw. Comput. Appl. 145 (2019) 102381.
[108] A. Ahalawat, S.S. Dash, A. Panda, K.S. Babu, Entropy based DDoS detection and mitigation in openflow enabled SDN, in: 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking, ViTECoN, IEEE, 2019, pp. 1–5.
[109] M. Xuanyuan, V. Ramsurrun, A. Seeam, Detection and mitigation of DDoS attacks using conditional entropy in software-defined networking.
[110] J. Cui, M. Wang, Y. Luo, H. Zhong, DDoS detection and defense mechanism based on cognitive-inspired computing in SDN, Future Gener. Comput. Syst. 97 (2019) 275–283.
[111] R. Li, B. Wu, Early detection of DDoS based on phi-entropy in SDN networks, in: 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference, Vol. 1, ITNEC, IEEE, 2020, pp. 731–735.
[112] N. Pitropakis, E. Panaousis, T. Giannetsos, E. Anastasiadis, G. Loukas, A taxonomy and survey of attacks against machine learning, Comp. Sci. Rev. 34 (2019) 100199.
[113] N. Bindra, M. Sood, Detecting DDoS attacks using machine learning techniques and contemporary intrusion detection dataset, Autom. Control Comput. Sci. 53 (5) (2019) 419–428.
[114] Q. Niyaz, W. Sun, A.Y. Javaid, A deep learning based DDoS detection system in software-defined networking (SDN), 2016, arXiv preprint arXiv:1611.07400.
[115] T. Hurley, J.E. Perdomo, A. Perez-Pons, HMM-based intrusion detection system for software defined networking, in: 2016 15th IEEE International Conference on Machine Learning and Applications, ICMLA, IEEE, 2016, pp. 617–621.
[116] A. Alshamrani, A. Chowdhary, S. Pisharody, D. Lu, D. Huang, A defense system for defeating DDoS attacks in SDN based networks, in: Proceedings of the 15th ACM International Symposium on Mobility Management and Wireless Access, 2017, pp. 83–92.
[117] D. Hu, P. Hong, Y. Chen, FADM: DDoS flooding attack detection and mitigation system in software-defined networking, in: GLOBECOM 2017-2017 IEEE Global Communications Conference, IEEE, 2017, pp. 1–7.
[118] A.B. Dehkordi, M. Soltanaghaie, F.Z. Boroujeni, A New DDoS Detection Method in Software Defined Network.
[119] J. Li, Z. Zhao, R. Li, H. Zhang, Ai-based two-stage intrusion detection for software defined iot networks, IEEE Internet Things J. 6 (2) (2018) 2093–2102.
[120] S. Guozi, W. JIANG, G. Yu, R. Danni, L. Huakang, DDoS attacks and flash event detection based on flow characteristics in SDN, in: 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS, IEEE, 2018, pp. 1–6.
[121] V. Deepa, K. Sudar, P. Deepalakshmi, Design of ensemble learning methods for DDoS detection in SDN environment, in: 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking, ViTECoN, IEEE, 2019, pp. 1–6.
[122] T.V. Phan, M. Park, Efficient distributed denial-of-service attack defense in SDN-based cloud, IEEE Access 7 (2019) 18701–18714.
[123] M. Myint Oo, S. Kamolphiwong, T. Kamolphiwong, S. Vasupongayya, Advanced support vector machine-(ASVM-) based detection for distributed denial of service (DDoS) attack on software defined networking (SDN), J. Comput. Netw. Commun. 2019 (2019).
[124] J. Li, Y. Liu, L. Gu, DDoS attack detection based on neural network, in: 2010 2nd International Symposium on Aware Computing, IEEE, 2010, pp. 196–199.
[125] R. Braga, E. Mota, A. Passito, Lightweight DDoS flooding attack detection using NOX/OpenFlow, in: IEEE Local Computer Network Conference, IEEE, 2010, pp. 408–415.
[126] Y. Cui, L. Yan, S. Li, H. Xing, W. Pan, J. Zhu, X. Zheng, SD-Anti-DDoS: Fast and efficient DDoS defense in software-defined networks, J. Netw. Comput. Appl. 68 (2016) 65–79.
[127] J. Cui, J. He, Y. Xu, H. Zhong, TDDAD: Time-based detection and defense scheme against DDoS attack on SDN controller, in: Australasian Conference on Information Security and Privacy, Springer, 2018, pp. 649–665.
[128] C. Li, Y. Wu, X. Yuan, Z. Sun, W. Wang, X. Li, L. Gong, Detection and defense of DDoS attack–based on deep learning in OpenFlow-based SDN, Int. J. Commun. Syst. 31 (5) (2018) e3497.

[129] T.M. Nam, P.H. Phong, T.D. Khoa, T.T. Huong, P.N. Nam, N.H. Thanh, L.X. Thang, P.A. Tuan, V.D. Loi, et al., Self-organizing map-based approaches in DDoS flooding detection using SDN, in: 2018 International Conference on Information Networking, ICOIN, IEEE, 2018, pp. 249–254.
[130] M.P. Novaes, L.F. Carvalho, J. Lloret, M.L. Proença, Long short-term memory and fuzzy logic for anomaly detection and mitigation in software-defined network environment, IEEE Access 8 (2020) 83765–83781.
[131] S. Dotcenko, A. Vladyko, I. Letenko, A fuzzy logic-based information security management for software-defined networks, in: 16th International Conference on Advanced Communication Technology, IEEE, 2014, pp. 167–171.
[132] T. Chin, X. Mountrouidou, X. Li, K. Xiong, Selective packet inspection to detect DoS flooding using software defined networking (SDN), in: 2015 IEEE 35th International Conference on Distributed Computing Systems Workshops, IEEE, 2015, pp. 95–99.
[133] P. Xiao, Z. Li, H. Qi, W. Qu, H. Yu, An efficient DDoS detection with bloom filter in SDN, in: 2016 IEEE Trustcom/BigDataSE/ISPA, IEEE, 2016, pp. 1–6.
[134] A. AlEroud, I. Alsmadi, Identifying cyber-attacks on software defined networks: An inference-based intrusion detection approach, J. Netw. Comput. Appl. 80 (2017) 152–164.
[135] M. Conti, A. Gangwal, M.S. Gaur, A comprehensive and effective mechanism for DDoS detection in SDN, in: 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob, IEEE, 2017, pp. 1–8.
[136] K. Kalkan, G. Gür, F. Alagöz, Sdnscore: A statistical defense mechanism against DDoS attacks in sdn environment, in: 2017 IEEE Symposium on Computers and Communications, ISCC, IEEE, 2017, pp. 669–675.
[137] J. Wang, R. Wen, J. Li, F. Yan, B. Zhao, F. Yu, Detecting and mitigating target link-flooding attacks using sdn, IEEE Trans. Dependable Secure Comput. 16 (6) (2018) 944–956.
[138] H. Wang, L. Xu, G. Gu, Floodguard: A dos attack prevention extension in software-defined networks, in: 2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, IEEE, 2015, pp. 239–250.
[139] A.F.M. Piedrahita, S. Rueda, D.M. Mattos, O.C.M. Duarte, Flowfence: a denial of service defense system for software defined networking, in: 2015 Global Information Infrastructure and Networking Symposium, GIIS, IEEE, 2015, pp. 1–6.
[140] X. Wang, M. Chen, C. Xing, SDSNM: a software-defined security networking mechanism to defend against DDoS attacks, in: 2015 Ninth International Conference on Frontier of Computer Science and Technology, IEEE, 2015, pp. 115–121.
[141] B. Yuan, D. Zou, S. Yu, H. Jin, W. Qiang, J. Shen, Defending against flow table overloading attack in software-defined networks, IEEE Trans. Serv. Comput. 12 (2) (2016) 231–246.
[142] L. Dridi, M.F. Zhani, SDN-guard: DoS attacks mitigation in SDN networks, in: 2016 5th IEEE International Conference on Cloud Networking, Cloudnet, IEEE, 2016, pp. 212–217.
[143] T.V. Phan, T. Van Toan, D. Van Tuyen, T.T. Huong, N.H. Thanh, OpenFlowSIA: An optimized protection scheme for software-defined networks from flooding attacks, in: 2016 IEEE Sixth International Conference on Communications and Electronics, ICCE, IEEE, 2016, pp. 13–18.
[144] R. Sahay, G. Blanc, Z. Zhang, H. Debar, ArOMA: An SDN based autonomic DDoS mitigation framework, Comput. Secur. 70 (2017) 482–499.
[145] S. Hameed, H. Ahmed Khan, SDN based collaborative scheme for mitigation of DDoS attacks, Future Internet 10 (3) (2018) 23.
[146] M. Conti, C. Lal, R. Mohammadi, U. Rawat, Lightweight solutions to counter DDoS attacks in software defined networking, Wirel. Netw. 25 (5) (2019) 2751–2768.
[147] K.K. Karmakar, V. Varadharajan, U. Tupakula, Mitigating attacks in software defined networks, Cluster Comput. 22 (4) (2019) 1143–1157.
[148] Y. Wang, T. Hu, G. Tang, J. Xie, J. Lu, SGS: Safe-guard scheme for protecting control plane against DDoS attacks in software-defined networking, IEEE Access 7 (2019) 34699–34710.
[149] A.S. Da Silva, C.C. Machado, R.V. Bisol, L.Z. Granville, A. Schaeffer-Filho, Identification and selection of flow features for accurate traffic classification in SDN, in: 2015 IEEE 14th International Symposium on Network Computing and Applications, IEEE, 2015, pp. 134–141.
[150] S. Agarwal, M. Kodialam, T. Lakshman, Traffic engineering in software defined networks, in: 2013 Proceedings IEEE INFOCOM, IEEE, 2013, pp. 2211–2219.
[151] C.E. Rothenberg, M.R. Nascimento, M.R. Salvador, C.N.A. Corrêa, S. Cunha de Lucena, R. Raszuk, Revisiting routing control platforms with the eyes and muscles of software-defined networking, in: Proceedings of the First Workshop on Hot Topics in Software Defined Networks, 2012, pp. 13–18.
[152] J. Xie, F.R. Yu, T. Huang, R. Xie, J. Liu, C. Wang, Y. Liu, A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges, IEEE Commun. Surv. Tutor. 21 (1) (2018) 393–430.
[153] S. Vissicchio, L. Vanbever, O. Bonaventure, Opportunities and research challenges of hybrid software defined networks, ACM SIGCOMM Comput. Commun. Rev. 44 (2) (2014) 70–75.
[154] J. McCauley, A. Panda, M. Casado, T. Koponen, S. Shenker, Extending SDN to large-scale networks, Open Networking Summit, 2013, pp. 1–2.
[155] S. Hassas Yeganeh, Y. Ganjali, Kandoo: a framework for efficient and scalable offloading of control applications, in: Proceedings of the First Workshop on Hot Topics in Software Defined Networks, 2012, pp. 19–24.
[156] W. Li, W. Meng, L.F. Kwok, A survey on openFlow-based software defined networks: Security challenges and countermeasures, J. Netw. Comput. Appl. 68 (2016) 126–139.
[157] S. Bhatia, G. Mohay, A. Tickle, E. Ahmed, Parametric differences between a real-world distributed denial-of-service attack and a flash event, in: 2011 Sixth International Conference on Availability, Reliability and Security, IEEE, 2011, pp. 210–217.
[158] S. Behal, K. Kumar, M. Sachdeva, Characterizing DDoS attacks and flash events: Review, research gaps and future directions, Comp. Sci. Rev. 25 (2017) 101–114.
[159] S. Floyd, V. Paxson, Difficulties in simulating the Internet, IEEE/ACM Trans. Netw. 9 (4) (2001) 392–403.
[160] L. Yao, P. Hong, W. Zhou, Evaluating the controller capacity in software defined networking, in: 2014 23rd International Conference on Computer Communication and Networks, ICCCN, IEEE, 2014, pp. 1–6.
[161] P. Wang, K.-M. Chao, H.-C. Lin, W.-H. Lin, C.-C. Lo, An efficient flow control approach for SDN-based network threat detection and migration using support vector machine, in: 2016 IEEE 13th International Conference on E-Business Engineering, ICEBE, IEEE, 2016, pp. 56–63.