Noname manuscript No.
(will be inserted by the editor)

A Comprehensive Survey of Security Threats and their Mitigation Techniques for next-generation SDN Controllers

Tao Han · Syed Rooh Ullah Jan · Zhiyuan Tan · Muhammad Usman · Mian Ahmad Jan* · Rahim Khan · Yongzhao Xu

Received: date / Accepted: date

Abstract Software Defined Network (SDN) and Network Virtualization (NV) are emerged paradigms that simplified the control and management of the next generation networks, most importantly, Internet of Things (IoT), Cloud Computing, and Cyber-Physical Systems. The Internet of Things (IoT) includes a diverse range of a vast collection of heterogeneous devices that require interoperable communication, scalable platforms and security provisioning. Security provisioning to an SDN based IoT network pose a real security challenge leading to various serious security threats due to the connection of various heterogeneous devices having a wide range of access protocols . Furthermore, the logical centralized controlled intelligence of the SDN architecture represents a plethora of security challenges due to its single point of failure. it may throw the entire network into chaos and thus expose it to various known and unknown security threats and attacks. security of SDN controlled IoT environment is still in infancy and thus remains the prime research agenda for both the industry and academia. This paper comprehensively reviews the current state-of-the-art security threats, vulnerabilities and issues at the control plane. Moreover, this paper contributes by presenting a detailed classification of various security attacks on the control layer. A comprehensive state-of-the-art review of the latest mitigation techniques for various security breaches is also presented. Finally, the paper presents future research directions and challenges for further investigation down the line.

Keywords Software Defined Networks · Controller · Denial of Service attacks · Spoofing attacks · Malicious injection attacks · Link Flooding attacks

* Indicates the corresponding author

Tao Han
DGUT-CNAM Institute,
Dongguan University of Technology, Dongguan 523808, Guangdong, China
E-mail: hant@dgut.edu.cn

Syed Rooh Ullah Jan
Department of Computer Science
Abdul Wali Khan University Mardan, Pakistan
E-mail: roohullahsyed@awkum.edu.pk

Zhiyuan Tan
School of Computing
Edinburgh Napier University, United Kingdom
E-mail: Z.Tan@napier.ac.uk

Muhammad Usman
Department of Computer Science and Software Engineering
Swinburne University of Technology, Australia
E-mail: musman@swin.edu.au

Mian Ahmad Jan
Department of Computer Science
Abdul Wali Khan University Mardan, Pakistan
E-mail: mianjan@awkum.edu.pk

Rahim Khan
Department of Computer Science
Abdul Wali Khan University Mardan, Pakistan
E-mail: rahimkhan@awkum.edu.pk

Yongzhao Xu
Dongguan University of Technology
Guangdong, China
E-mail: xuyz@dgut.edu.cn

1 Introduction

By 2020, it is expected that the Internet of Things (IoT) will incorporate nearly 50 billion real-world physical devices. Numerous solutions have been proposed and implemented to deal with an increased number of connected devices, however, they were not designed while keeping in mind the evolution of IoT-enabled devices [84]. The projected growth in the number of connected

devices means that the existing wired/wireless and mobile networks need to evolve to become more intelligent, secured, scalable and resource-efficient to incorporate them. The scalability of these networks is essential to manage the diverse nature of data generated by these devices. Software Defined Network (SDN) and Network Virtualization (NV) are the two promising technologies to serve as key enablers for the IoT of the near future [35]. NV allows the service providers to form separate and isolated virtual networks by enabling them to share physical resources. It offers a reduced cost by sharing the network infrastructure and improved time to market for novel applications. For future IoT networks, NV will be a crucial feature that will enable differentiated Quality-of-Service (QoS) for the diverse usage scenario and quick introduction of new applications and services.

SDN, on the other hand, is a novel programmable architecture that simplifies the control and management of next-generation networks. It has changed the way a network operates by decoupling the data plane from the control plane and manages the whole network through a centralized control intelligence [100],[114], [79], also known as SDN controller. An SDN controller is the backbone of an SDN architecture because it performs the essential operations related to the control and management of the underlying networks [78], [97], [165], [158], [178]. It is responsible for the establishment and termination of data flows at the data plane, based on various data handling policies. It implies that the network elements at the data plane are simple forwarding devices, managed and controlled by the controller. The SDN controller can assign the required resources by configuring the network policies as per requirements of an application and network hardware at the data layer. Moreover, it provides an up-to-date view of the network and topology by collecting various statistical data using open APIs. This allows network managers to apply different network-wide policies such as redirection of traffic and blocking certain packets, at the packet level without actually touching the underlying network. These attributes brought substantial managerial benefits. Although SDN has brought significant changes to the way a network operates, the single-point dependency remains a prime and challenging security issue. Compromising the security of the controller means that the safety of the whole network is at stake.

Despite all the benefits offered by the SDN , there are numerous challenging issues that need to be tackled prior to its widespread adoption. Some of these challenging issues include but are not limited to scalability, fault tolerance, communication overhead, security provisioning, and single point dependency. The centralized nature and a single point dependency of the SDN controller is its strength. However, at the same time, it is its weakness from security point of view. For instance, if the security of the controller is compromised, protection of the whole network is compromised, and the controller becomes vulnerable to a wide range of attacks. According to [74], "why take over the hosts when you can take over the whole network". It is, therefore, crucial to secure not only the controller but all the layers and interfaces while designing an SDN architecture. Security needs to be delivered as a service to protect the network resources from unauthorized access and attacks. This is because the security was not initially considered while designing an SDN architecture [8], [67]. Thus, an SDN design requires a simple, scalable, cost-effective, and in particular, an efficient and secure architecture. In an SDN architecture, the control plane is particularly vulnerable to a wide range of security attacks due to its strategic and centralized nature [76], [144], [159]. These attacks include but are not limited to DoS, DDoS, spoofing, and malicious injection attacks [8],[138], [170],[18],[106] as depicted in Fig. 1. Besides these attacks, an attacker may exploit various vulnerabilities in the Open Flow Protocol, the most commonly used protocol at the southbound interface that facilitate communication between the control and data layers of an SDN architecture [10], [161]. Securing the controller remains a key challenging area due to it single point failure for the research community in the years to come.

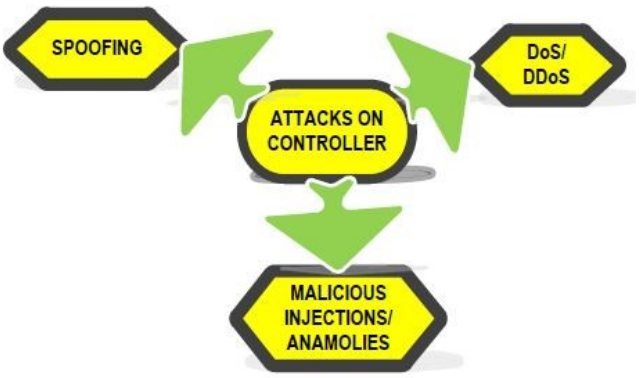

Fig. 1 Security Attacks on an SDN Controller

Based on the literature, various SDN-related survey papers are available exploring various dimensions of SDN security. For instance, SDN and its evolution [56], [25], [77], [104], [82], whilst only a few surveys focuses on the security of SDN [88], [104], [71], [166], [141], [42], [138], [56], [139], [21]. [6], [167], [138], [104]. Among them,[139] in 2013 first discussed several security challenges in SDN without an analysis model. Ad-

ditionally, they did not provide a discussion of the potential countermeasures. Furthermore, [16] conducted a survey on SDN security by means of STRODE threats model, which discuss various security issues to the overall SDN architecture. Furthermore, in [6], the authors studied various security challenges experienced by the protocols and architecture of an SDN architecture. This work further explored the existing solutions for mitigating various attacks and at the same time, classified these solutions in term of scalability, reliability, security, and performance. Another valuable work in this regards was presented in [167]. The authors surveyed various DDoS attacks targeting an SDN-based cloud architecture. An in-depth analysis of emerging trends, features and mitigating techniques for these attacks were also explained. In [138] the authors provide an overview of various security challenges, introduced at each layer of an SDN paradigm. They suggested various security enhancement techniques to address the aforementioned challenges. In [104], [19], [94], the authors investigated numerous security challenges faced by the southbound interface, i.e., OpenFlow. Moreover, various solutions were proposed to overcome such challenges. A layered taxonomy of various vulnerabilities that target each layer of an SDN architecture was also discussed. Moreover, [21], [166] considers only one type of attack, i.e., DDoS attack, data plane security and challenges [141],[42] or discusses security challenges of the whole SDN architecture where only a section is dedicate to the security of the SDN controller and are thus limited in their scope and completeness as they are not controller exclusive [25] [56] [104] [138] [171]. Unlike these surveys, This paper is first of its kind that provide an in-depth analysis of various security issues related to SDN controller along with their countermeasures. A systematic taxonomy of control plane agnostic attacks, i.e., Denial of Service (DoS), Distributed DoS, Spoofing attack and Malicious Injection attacks is also presented. Furthermore, this survey provides an insight into various solutions for the detection and mitigation of the aforementioned attacks along with their strength and weaknesses with regard to an SDN controller. Because, a secure controller implies a secure SDN network controller. Failure of the controller is failure of the whole SDN architecture. Due to this dependency, it is mandatory to know various controller agnostic attacks, so as to protect it as well as for its widespread adoption.

This survey paper will help us to answer questions like When, Why and which solution is the most appropriate to tackle a particular type of attack on the controller, due to unlimited number of mitigation techniques available in the literature. it thus help the researchers in choosing the most appropriate technique for the future on one hand, while suitable choice for practitioner on the other hand in-order to fully benefit from this technology along with making SDN a promising, trustable, dependable and secure architecture for the years to come. The main contributions of this paper are as follow.

1. This paper comprehensively reviews the existing state-of-the-art security threats, vulnerabilities and issues at the control plane.
2. This paper present an up-to-date, thematic taxonometric classification of various security attacks on the control layer of an SDN architecture.
3. A detailed analysis of these attacks along with their mitigating techniques is also provided. Moreover, design trade-off of these mitigation techniques is also provided. These mitigation techniques are summarized in a table at the end of each subsection, highlighting their main attributes, strength and weaknesses.
4. Finally, various research gaps are identified that open the gates for further exploration.

The rest of the paper is organized as follow. In Section 2, we provide an overview of an SDN architecture. In Section 3, we provide a detailed description of attacks and their countermeasures on an SDN controller. In this section, we provide a brief taxonomy of these attacks as well. Open security issues, challenges and future research directions, in the context of SDN controller, are provided in Section 4. Finally, the paper is concluded in Section 5.

2 Overview of SDN Architecture

According to Open Networking Foundation (ONF), the control and data planes in an SDN architecture are decoupled. Furthermore, the network intelligence and states are logically centralized, and the underlying network infrastructure is abstracted from the applications [157]. The architecture is vertically divided into three layers, i.e., an application layer, a control layer and a data layer [63]. These layers are separated from each other using northbound and southbound programming interfaces (APIs) [97], [124], as depicted in Fig. 2. The northbound interface facilitates the communication between the application layer and control layer. Although there exists no standardized interface, the most commonly used interface for application-to-control communication is Rest API. On the other hand, the southbound interface facilitates the communication between the control layer and data layer of an SDN architecture. OpenFlow is the most widely used API for the southbound interface. A detail discussion on these interfaces

along with the aforementioned three layers are provided in this section.

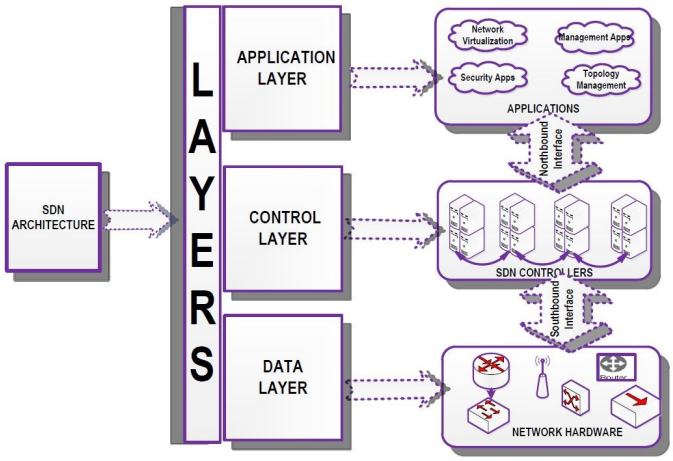

Fig. 2 An SDN Architecture

2.1 Application Layer

Application layer, also known as an application plane, provides a set of services for various applications, such as security provisioning, QoS, routing, and deep packet inspection (DPI), are few to mention here. Each application consists of an SDN Application Logic and one or more Northbound Interface (NBI) drivers. Each application supported at this layer programmatically states its requirements and desired network behavior to the controller via the northbound interface.

2.2 Northbound Interface

The Northbound Interface (NBI) facilitates application-to-control plane communication using vendor neutral open APIs. This interface is responsible for providing an abstraction of the underlying network. Furthermore, it empowers the applications by expressing the required network behavior to the controller. However, the NBI lacks a standardized interface and as such, is used on an ad-hoc basis as per SDN administrator choice.

2.3 Control Layer

Control layer, also known as control plane, is responsible for the management and control of the overall network. This layer contains an important network component, known as the SDN controller. This component is logically centralized, however, in principle, it is physically distributed [78], [97], [165]. It is responsible for establishing and terminating data flows on various network components at the data layer, based on data handling policies. Its prime responsibility is to fine-tune the forwarding tables which reside in the forwarding plane. This tuning is based on the network topology or external service requests [68]. This layer abstracts the network complexity by maintaining an up-to-date network holistic view. There are various components of an SDN controller such as single or multiple NBI Agents, SDN Control Logic, and the Control-Data-Plane Interface (CDPI) agent. Also, the logically centralized controller can be applied to a wide variety of physical media. For instance, guided media such as Ethernet, and unguided media such as Wi-Fi, LTE and WSN. Some of the most widely used controllers are highlighted in Fig. 3.

Fig. 3 SDN Controllers

2.4 Southbound Interface

Southbound interface provides communication between the control and data layers of an SDN architecture. This interface provides event notification and statistical reports using southbound APIs. This interface is also known as a controller-switch communication interface because it facilitates communication between the controller and a switch, at the data plane. This interface enables the network managers to implement the controller decisions on the network components of a data plane. OpenFlow is the most popular and widely used protocol at the southbound interface. Other such protocols are Cisco's Open Network Environment Platform Kit (onePK), Junipers contrail [85] and Forwarding and Control Element Separation (ForCES) frame-

work [51],and protocol oblivious forwarding (POF) [149], [85].

2.5 Data Layer

The lowest layer in the SDN architecture is known as data layer/plane. This layer consists of forwarding network components such as, routers, physical and virtual switches, and access point. As a result, this layer is also known as infrastructure layer [83]. The data layer is responsible for the implementation of management functionalities such as forwarding data, fragmentation and reassembly, as instructed by the controller to the SDN-enabled switches. Furthermore, the information collected by these OpenFlow switches are forwarded to the controller, using a southbound interface [66].

3 Controller-related Threats and Attacks

In SDN, all network-related functionalities are managed, controlled and secured from a centralized controller. The single-point dependency and programmable nature of an SDN controller make it a potential choice for the attackers. If security of the controller is compromised, the whole network is vulnerable to various attacks [76], [144], [159]. An adversary can launch various attacks by exploiting vulnerabilities of a controller. These vulnerabilities can lead to castrotrophic situations, particularly in the absence of a robust and secured policy [8], [112]. For instance, an attacker can spoof the address of a controller or insert a fake controller to hijack the whole network. As a result, it is essential to design a secure mechanism to protect the network in general, and the controller in particular, from a wide range of attacks.

The southbound interface needs to be protected and secured against communication overhead. Therefore, unnecessary communication that results in congestion at this interface needs to be avoided for the smooth functioning of the network. In addition, the availability and confidentiality of the controller need to be ensured. Therefore, it is essential to secure both the controller and the southbound interface against attacks and is attracting the attention of the researchers in recent years [40], [130], [138]. Securing the controller and southbound interface may include:

1. Ensuring availability of the controller by protecting it against flooding attacks such as DoS and DDoS attacks.
2. The controller must be guaranteed with security policy enforcement, high availability and the minimum possible delay experienced during incoming packets [157].
3. Being a programmable architecture, the operating system installed on a controller must be guarded against various vulnerabilities such as exploitable patches, back and open door accounts, and open ports.
4. The controller should be protected against external and physical threats.
5. The controller needs to have an automatic alert system that needs to control the data-to-control plane communication to a minimum level, during an attack as well as informing the administrator in case of an attack.

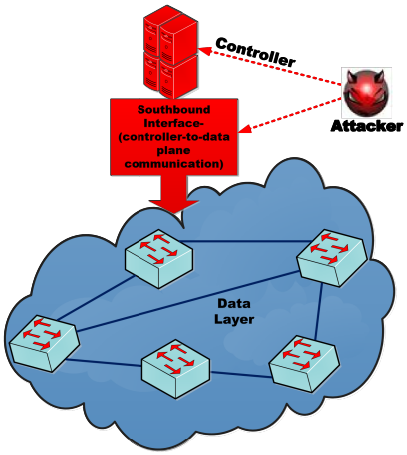

Fig. 4 Security attacks on the SDN controllers and Southbound interface

In this section, we provide a detailed discussion of the most common attacks, such as DoS/DDoS, spoofing and malicious injection, in the context of SDN controller

3.1 DoS/DDoS Attacks

The denial-of-service (DoS) and distributed DoS (DDoS) are the most common attacks launched by cyber criminals, cyber extortionists and hackers. These attacks flood the controller with spoofed packets that result in serious disruption of the provided services. These attacks compromise the controller, and as such, it is unable to respond to legitimate requests and fail to offer services due to the flooding of illicit traffic by an attacker, as shown in Fig. 5. Such a situation results in the exhaustion of network resources. Moreover, the

controller is unable to differentiate between a genuine request and an attackers request, due to the changes in the packet header that look somewhat identical for both these requests. During these attacks, it is a challenging task to analyze the huge traffic flow. Thus, the accuracy of the services provided by the controller is compromised along with lower response time. There are various reasons for launching DoS/DDoS attacks, such as financial gains, political gains, competitive edge and disruption of services [64].

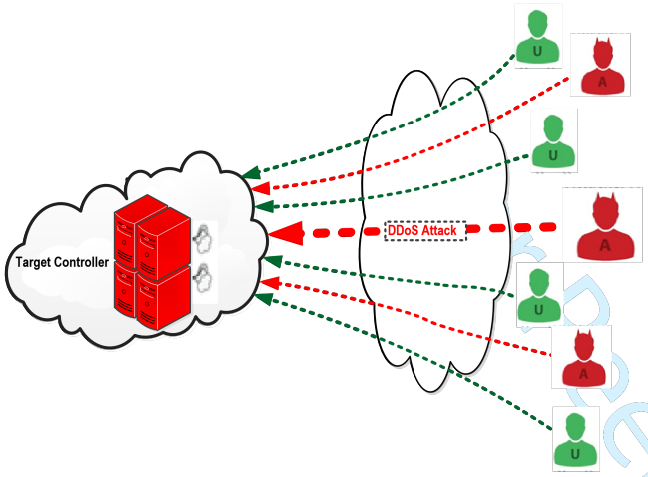

Fig. 5 DoS attacks on an SDN Controller

Keeping in view the importance of a controller in an SDN environment, there is a need for sufficient research to detect, mitigate and design preventive techniques, for the ever-increasing, novel and highly sophisticated DoS and DDoS attacks. In literature, there exist few works that study the unique relationship between DoS/DDoS attacks and an SDN controller [167]. Moreover, the available literature is not up to date. It is mainly due to the unique relationships between an SDN and DDoS attacks that are yet to be discovered. In this section, we present the latest DoS and DDoS mitigation techniques based on Entropy, Machine learning and Traffic pattern analysis from the literature [27], [49], [52], [80], [173], [87], [106], [116], [144], [145], [176], [160], [161]. Moreover, a summary of these mitigation techniques is provided in Table 1.

3.1.1 DoS and DDoS mitigation Techniques

In [27], the author proposed StateSec, that employs a stateful method to utilize in-switch processing capabilities for accurate detection and efficient mitigation of DDoS attacks in an SDN architecture. StateSec mechanism offloads the controller by reducing the communication overhead at the southbound interface. The security management of StateSec consists of three main phases, i.e., traffic/flow monitoring, anomaly detection, and mitigation/countermeasures. Traffic monitoring is performed within the switch while anomaly detection and mitigation are usually implemented at the control layer. During traffic monitoring, the proposed method monitors and matches packets against four configurable traffic features, i.e., IP addresses of source and destination and port addresses of source and destination inside a switch, using stateful programming. During the anomaly detection, an entropy-based algorithm is used at the controller for the detection of anomalies and various types of attacks, such as port scan, DoS and DDoS [99]. Finally, during mitigation, the controller alleviates these attacks by taking appropriate action, such as filtering, rate limiting and re-routing malicious traffic toward a black hole or fake server, i.e., honey pot. Extensive simulation results show that StateSec is an effective and efficient technique against DDoS attacks, that incurs a lower overhead at the controller and southbound interface. However, StateSec is unable to detect attacks with unknown patterns. Furthermore, the accuracy of the detection rate needs to be improved significantly.

In [80], the authors proposed a scheme to protect the SDN controller against DDoS attacks. During the first stage, the proposed method predicts the large volumes of incoming packets. These packets are associated with every new request for each OpenFlow switch. If the amount of incoming packets is higher than a specific threshold value, the rest of the requests are forwarded to a secured gateway to determine a DDoS attack. An algorithm is employed at the gateway by designing rules that filter out those requests that cause a dramatic decrease of entropy. Finally, the controller forwards these rules to each OpenFlow switch that inspects the incoming packets. The switch requests for irregularities, in case of irregular requests. Simulation results prove that the average of false positive and false negative is less than 2%. However, the comparison needs to be made with similar algorithms, proposed in the literature.

Security analysis and monitoring are the two core elements that are required for ensuring the security of a controller. In [144], the author proposed AVANT-GUARD, a secured framework to protect the network against DoS attacks. AVANT-GUARD addresses the two primary security challenges, i.e., reducing communication overhead at the OpenFlow protocol (southbound interface) and providing a quick response to the changing flow dynamics at the data plane. To overcome the first challenge, AVANT-GUARD adds connection migration to the data plane that avoids any further communication. To overcome the second challenge, the

Table 1 A Summary of mitigation techniques for DoS/DDoS attack on the SDN controller

Mitigation Techniques	**Issues Addressed**	**Strengths**	**Weaknesses**
StateSec [27]	DDoS and anomalies detection	Efficient against attacks, lower communication overhead	Not tested on complex networks, masking anomalies, and also unable to detect anomalies disturbing randomness
OpenFlow switch [80]	DoS	efficient and effective, Average of both false positive and false negative are less than 2%	Low accuracy, needs to be evaluated using formal methods
AVANT-GUARD [144]	DoS	Reduce data-to-control plane communication overhead, quick response to the changing flow dynamics at the data layer	Need to be installed at each network component, otherwise, unable to secure the network
Entropy-based DDoS detection and mitigation [160]	DDoS	computationally lightweight, generates fine-grained patterns, lower communication overhead at the southbound interface	Need to be integrated with others techniques to accomplish threshold determination and multi-element weight assignment
SGuard [161]	DoS	scalable, effective and easily integrates with OpenFlow	Controller exclusive, needs to be evaluated in complex scenarios to obtain better results
FL-GUARD [106]	DoS, DDoS	effective against spoofing attacks	Generic detection algorithm
Fuzzy-based DoS Detection [52]	DoS	lightweight, efficient and effective	Needs to be evaluated in complex environments with different topologies
Early detection of DDoS attacks against SDN controllers [116]	DDoS	lightweight, detect attacks within the first five hundred packets	Unable to detect attack that generate varying traffic flows

author introduced a statistic collection service on the data plane. Simulation results show that the AVANT-GUARD protects the SDN network against TCP syn-flooding attacks and network scanning attacks. However, it does not preserve the SDN against DoS attack on the application layer as well as on the Internet Control Message Protocol (ICMP) and UDP (User Datagram Protocol) layers.

In [160], the authors proposed an approach to protect SDN controllers from DDoS attacks and anomalies. The proposed method employs a distributed and lightweight entropy-based DDoS attack detection module on every edge switch at the data plane. It results in a lower communication overhead at the controller as well as at the southbound interface. The higher value of entropy indicates an increased variation in the probability distribution, whereas, a lower value indicates a decreased variation. The proposed approach uses the destination IP address at each switch for a probability distribution. As soon as a DDoS attack is detected, the alert information is forwarded to the controller for further necessary actions. The supremacy of the proposed system is that it generates fine-grained patterns with low calculation overhead in comparison to the traditional volume-based traffic analysis scheme [21]. The proposed approach has its own shortcomings. For ex-

ample, the relevant information about the distribution of the analyzed feature is lost that leads to masking of anomaly effects [58]. Similarly, the different distributions with the same amount of uncertainty cannot be distinguished by entropy values. Therefore, the proposed method is unable to detect anomalies that do not disturb randomness [86].

In [161], SGuard, a lightweight and efficient security application on top of the NOX controller, for the detection of DoS attacks was proposed. The proposed architecture consists of two modules: access control and classification. The access control module uses authorization information for tracing the genuine source of a packet by taking preventive measures, using such information. As soon as a new entity enters the network, this module gathers information related to this entity such as medium access control address (MAC), logical address, port address and switch ID. Based on this information, SGuard compares the source address, i.e., MAC/IP, against the hash table entries. This module allows normal traffic into the network while denies packets from a spoofed source. Classification module, on the other hand, employs a Self Organizing Map (SOM) [95], based on the artificial neural network to classify network traffic as normal or abnormal, using a feature vector. The classification includes three sub-parts: data collector, feature extractor and a classifier. The data collector gathers the flow entries from the flow tables of OpenFlow switches, at a particular time interval. Once the data is gathered, features are extracted and are then classified using the most relevant data as feature vector from the data flow entries. This feature vector is used for the classification between normal and malicious traffic. Moreover, the classification is enhanced further by feature ranking and selection algorithms to obtain high accuracy and efficiency. All modules of SGuard cooperate in an SDN controller to guard against DoS attacks. SGuard can easily be integrated with the OpenFlow, without making any changes to its underlying architecture. Based on extensive experimental work, it was concluded that the proposed approach is lightweight, scalable and effective against these attacks. However, the overall data training time needs to be minimized to improve the classification performance. Furthermore, SGuard needs to be evaluated for large and complex scenarios.

In [106], the authors proposed a novel DDoS detection and prevention system, known as Floodlight-based Guard system (FL-GUARD). The architecture of FL-GUARD is based on three components: anti-spoofing module, sFlow-RT collector and a blocking module at the application plane of an SDN architecture. Initially, FL-GUARD uses the concept of dynamic IP address binding for the identification of anti-source of spoofed IP address. Next, FL-GUARD employs an improved network monitoring component, known as sFlow-RT collector, that monitors the traffic in real-time with lower delay and enhanced accuracy. Finally, DoS and DDoS attacks are detected at the source port using C-SVM, an improved version of the Support Vector Machine Algorithm (SVM). The modular design of an FL-GUARD is convenient for further modification and extension. The simulation results conclude that FL-GUARD is an efficient solution against DDoS attacks. Nonetheless, accuracy and performance of the proposed method can be enhanced further, using various other machine learning approaches such a random forest and decision tree. The FL-GUARD needs to be evaluated using performance metrics such as specificity, accuracy, precession, sensitivity, and F-Measure. Finally, execution time needs to be taken into account to understand and evaluate the behaviour of the proposed architecture fully.

In [52], the authors proposed a fuzzy-based security mechanism to guard an SDN controller against DoS attacks. The proposed approach uses a Tree-reweighted message passing (TRW) algorithm [136] along with rate limiting and fuzzy inference [34]. The inference approach is the most realistic one for resolving fuzzy inference problems as it is lightweight in term of computation and resource utilization. Simulation results reveal that the proposed method effectively detects DoS attacks in comparison to other security mechanisms. However, the proposed scheme needs to be evaluated, using complex scenarios having large volumes of traffic. Furthermore, the proposed approach does not have provisioning for mitigating strategies against attacks. In [49], the authors proposed an anomaly detection technique, for the mitigation of DDoS attacks on the controller in an SDN environment. The proposed method calculates a standard deviation of packet rate, collected for certain time intervals. The controller gathers these statistics from the OpenFlow switches at the data plane. A comparison is made between the previously calculated deviations against a real deviated value in the data set, to detect anomalies. The proposed approach employs a three-stage solution for the detection and mitigation of DDoS attack, using an RYU controller. During the first stage, irregularities are identified in the network flow. Next, the source is traced back, using packet analysis of the samples. Finally, the incoming packets from malicious sources are dropped. Although the proposed approach is efficient for the detection and mitigation of DoS attacks, its performance is dependent on the underlying dataset that are used for training purposes.

Thus, its performance needs to be tested using more than one datasets.

In [116], the authors proposed an entropy-based solution that detects DDoS attacks at the controller, using randomness of the incoming packets. The proposed method measures the probability of the occurrence of an event concerning the total number of events for early detection of an attack. The implementation of the proposed solution is based on a threshold value of entropy for the efficient detection of a DDoS attack. It implies that if the entropy value is lower than this threshold value, it needs to be considered an attack. The proposed approach is lightweight in term of the resources used and is capable of detecting DDoS within the first five hundred packets of the traffic. However, this approach is not reliable since the threshold value varies in different scenarios. Due to the programmable nature of an SDN, the network configuration may change while the network is still performing real-time monitoring. Furthermore, the proposed approach lacks any mitigation strategy.

3.2 Spoofing Attacks on the SDN controller

The controller is the backbone of an SDN network as it manages and controls the whole network. Due to its centralize nature, it is vulnerable to many types of security attacks. One such attack is spoofing attack. In spoofing attacks, an adversary launches attacks on a legal entity (server/system) by mimicking a legitimate user. The adversary forges the network information, i.e., IP address, MAC address, and ARP, intentionally by hiding its original identity, as shown in Fig. 6 [122]. These attacks violate the authentication security property of an SDN controller. Some studies show that SDN controllers, such as Floodlight, Open Daylight, Beacon, and POX are adversely affected by spoofing attacks [10]. In SDN, spoofing occurs in many forms such as IP spoofing, ARP spoofing and controller spoofing. In IP spoofing, IP address other than the attacker real IP address is used to hijack the whole network.

On the contrary to IP spoofing attacks, in Address Resolution Protocol (ARP) spoofing attacks, an association is made between the MAC address of an attacker with the IP address of a legitimate host [104]. These attacks result in the hijacking of traffic from the intended genuine users and as such these users are taken out of the network. In controller spoofing attacks, a fake controller is inserted into the network that pretends to be a legitimate controller by tricking the users. In this work, we aim to refer to all these attacks as spoofing attacks collectively. A detailed discussion on spoofing attacks can be found in [16], [70], [45].

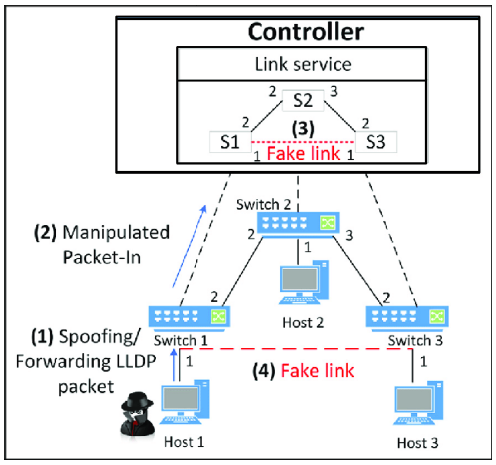

Fig. 6 IP Spoofing attack on an SDN Controller

The most challenging aspect of these attacks is tracing back the origin of an attack. This is because of poisoning network visibility, infecting topology information, misconfiguration and hijacking of services and application provided by an SDN controller. Due to these complications, sophisticated attacks such as DoS attacks, network hijacking attacks, blackhole attacks by manipulating the routing services inside the controller, man-in-the-middle attack and sometimes complete failure of the entire network takes place [7], [94], [138]. Looking at the severity of these attacks, in this section, we will discuss various solutions proposed in the literature for the detection and mitigation of an entire range of spoofing attacks that exclusively target the controller. A substantial amount of work is previously carried out in the literature to examine these attacks along with their mitigation techniques [2], [4], [9], [54], [61] [93], [107], [152]. This work is an effort to combine the previous and latest related research work. Moreover, this section also highlights strength and weaknesses as well as possible future extensions in the aforementioned research work. It will help the research community to better understand this domain and the research efforts carried out for possible future exploration. Furthermore, it is crucial for the widespread adoption of SDN-based networks. All these attacks are summarized in Table.2.

3.2.1 Spoofing mitigation techniques

In [9], the author has proposed Software Defined Security controller (SDSec), based on the Open vSwitch Controller, for the detection and mitigation of MAC and IP spoofing attacks [10]. SDSec uses an improved version of the OFDP protocol, known as Link Layer Discovery Protocol (LLDP), to identify an active link in the network. This is because the older version of

Table 2 A Summary of various mitigation techniques for spoofing attack on the SDN controller

Mitigation Techniques	**Issues Addressed**	**Strengths**	**Weaknesses**
SDSec [9]	IP and MAC spoofing prevention	efficiently detects and prevents IP and MAC spoofing attacks	does not integrated fully in SDN, and unable to offer the acceptable level of QoS as well as privacy
ARP Spoofing Mitigation [2]	ARP Reply and Request spoofing attacks	Attack detection with minimum latency and increased reliability, Performs port-level ARP packet monitoring	Lacks of an all-element threat model
Anomaly traceback [61]	Spoofing and traceback	Can easily be implemented at the SDN controller without developing dedicated routers	lacks evaluation on a benchmark data set, lack of trace route mechanism for an anomaly, and does not devise any flow statistics procedures to weighted anomalies
SDNsecured [54]	DDoS attack using IP spoofing, reply, ARP spoofing and man-in-the-middle	Employ AES for encryption and decryption while TLS for secured key exchange among the switches	lacks scalability evaluation of the controller, and cross layer controller security
Hybrid SDN [154]	ARP spoofing	A separate server is used for the collection and analysis of ARP request along with topological information of the whole network	does not study the frequency of traffic rule updates on the network performance, and non-scalable
Security-awareness SDN [55]	Network scanning, OpenFlow flooding, Switch compromise and ARP attack prevention	Security situation awareness based on features extraction with low overhead	lacks a balance between resources utilization and performance
Mitigation of DNS Amplification [93]	DNS Amplification attacks	Stores history of DNS queries as an evidence to differentiate normal packets from malicious packets	high communication overhead due to memory lookup
User Flow Validation Approach [107]	Flow table overloading, DDoS attacks and link spoofing attacks	Uses multiple Discrete-Time Finite-State Markov Chain (DTMC) model for users flow validation, unsupervised hashing for link spoofing and L1-ELM for classification	does not ensure accuracy of anomaly detection
Software-dened Mobile Networks [75]	IP-related source spoofing, DoS and DDoS	Multi-tier component-based security architecture for threats detection	non-scalable

LLDP is vulnerable to spoofing attacks [11]. The operation of SDSec is based on two tables, i.e., switch table and host table, that is added to the controller using the SQLite in-memory database. The host table holds information such as hostname, IP addresses, MAC addresses, concerned switch and its connected interfaces, authentication status and action on traffic, relevant to the hosts in the network. The switch table, on the other hand, holds information such as names, IP addresses, MAC addresses and available interfaces, about trusted switches of the network. Every time, a new switch or host join the network; it is authenticated before its communication with the SDSec controller. The latter either permits or denies the former to communicate on the network. The information of a new entity is checked and is authenticated by inspecting their data against the information contained in both of those tables. In case of a match in either of these tables, a false value is assigned to the AUTH field that indicates IP/MAC spoofing attacks. This means that the new host is using the same information similar to the one used by the genuine host in the network. Once a device leaves the network, its related information is removed from those tables to facilitate re-joining the network in the future. The proposed controller is extensively simulated in Mininet simulator using customized topology. Simulation results show that SDSec can efficiently detect and prevent IP and MAC spoofing attacks. The performance of SDSec may be enhanced further by adding more tables to the controller to cater for other types of attacks. Furthermore, the effectiveness of the proposed controller should be evaluated using other performance metrics and using various topologies and complex scenarios.

The authors in [2], proposed a technique that mitigates Address resolution protocol (ARP) request as well as reply spoofing attacks on the controller of Software Defined Networks. The notable features of this technique are that it protects the network against the said attacks with increased reliability, minimum latency and minimum communication overhead. The proposed method performs port-level ARP packet monitoring by adding an ARP module to the controller for the successful detection and mitigation of spoofing attacks on the controller. Furthermore, it can also guard a controller against communication overloading during DoS and other such attacks under various scenarios. Simulation results reveal that the proposed approach can effectively mitigate these attacks with lower overhead. However, The proposed technique is studied in a LAN network with a single controller, thus, its effectiveness should be studied in a complex environment with multiple controllers.

In SDN, tracing back the sources of an anomaly is a real challenge. Anomaly refers to an attack using a spoofed packet or misbehavior by an attacker. The authors in [61] proposed a method that passively identifies switches that are on the network path of an anomaly. Because SDN technologies tend to be deployed in the next generation networks including data centers, the proposed method can easily be implemented without developing dedicated routers like usual IP traceback techniques. The concept of the proposed method is based on forwarding rules with different parameters. The two most crucial forwarding rules are *matching* and *instruction*. The matching rule works like a filter where each packet's header is checked and matched with that of the switch tables entry to confirm whether it belongs to the flow or not. As a result, the actions such as forwarding/ modification are taken for each packet. On the other hand, instruction refers to a set of actions. OUTPUT is the primary instruction that forwards the packet to a particular port of a switch. Other such actions are *modification* of the header of a packet, i.e., MAC address, TTL, and various counters for monitoring purposes, i.e., number of bytes, number of packets and duration. The proposed method is evaluated using various topologies and various attacks such as distributed attacks with many hosts to study its effectiveness. It is concluded that the proposed approach fulfill its design objectives. However, this method may be enhanced further by using stochastic analysis to find out the route that has been taken by an anomaly.

IP spoofing, reply attacks, ARP spoofing, DDoS along with man-in-the-middle attack are series of threats to an SDN architecture in general and the controller in particular. In [54], the author proposed a solution that guards the SDN controller against the attacks above. The authors has employed Advanced Encryption Standard (AES) [26] along with Transport Layer Security (TLS) [48]. AES was used for encryption and decryption purposes while TLS was employed for secured key exchange among the switches. Moreover, IPsec is employed for tunneling between the hosts and gateways. Thus, IPsec preserves confidentiality and authenticity of data packets. Simulation results show that the proposed approach can effectively mitigate DDoS attack using IP spoofing, reply, ARP spoofing and man-in-the-middle attack. However, the proposed methods should be evaluated and tested in a real-time environment to study its effectiveness.

An automatic ARP spoofing detection and mitigation mechanism for hybrid SDN were proposed [154]. Hybrid SDN refers to the partial installation of SDN-enabled devices in a traditional network. The key benefits of such an architecture are to achieve all the benefits

of SDN from a traditional environment with lower deployment cost. The proposed method achieved this by installing a separate server that collects all the ARP requests. Furthermore, the controller and southbound interface are protected from the unnecessary processing of malicious data from the attackers by diverting them towards that particular server. In addition to that, topological information of the whole network is also gathered at the aforementioned server. The authors have employed a graph-based traversal mechanism that represents the network topology in the form of a graph. It can aids in the accurate detection of the attacker's location by verifying legitimate users. Next, the flow rules related to forwarding of ARP packets from the source to that particular server are installed on the switches for further analysis and accurate detection of ARP spoofing attacks. Simulation results demonstrate that the proposed method can effectively detect and mitigate threats and attacks related to ARP spoofing but lacks real-time evaluation and results.

Another beneficial work that focuses on the detection of four types of attacks, i.e., network scanning attacks, OpenFlow flooding attacks, switch compromised attacks and ARP attacks, targeting both the data plane and control plane of the SDN controller is presented in [55]. The author proposed a security situation-awareness approach based on flow features extraction. The author considered a total of twelve features for these four different types of attacks. Furthermore, multiple observations-based hidden Markov model (HMM) [129] was employed for the situation assessment purposes and building a quantification model in the assessor. The quantification model calculates the situation value and predicts the SDN situation status. Higher the situation status value, higher is the risk of attack. Moreover, The Baum-Welch algorithm [163] is employed to calculate probabilities and model training while Viterbi algorithm [73] was employed for predicting the status of the network. As an initial step towards security situation awareness in SDN, simulation results show its effectiveness. However, the proposed approach may be further extended to consider other type of attacks. Furthermore, accuracy and efficiency of the proposed method may be further enhanced to adapt it for a real-time environment.

Another significant contribution in this area is presented in [93] that proposes a novel security framework to protect the network against DNS amplification attacks. The proposed framework stores the history of DNS queries and uses it as an evidence to differentiate between normal and malicious packets. The proposed framework consists of two main components: a switch and an SDN controller. The responsibilities of the switch are that it stores mapping (strict one-to-one mapping), and validate query records related to DNS request, e.g., the source IP addresses, destination IP addresses contained in a DNS request message. Initially, a switch checks whether the received packet is a DNS request. If true, the information is stored locally at the memory of the switch or forwarded to the controller in case of unavailability of memory in the switch. Next, the switch checks the validity of the request with the DNS response available in its memory. Upon matching, the request is then forwarded and becomes part of the DNS requests which is later on used for validation purposes. Simulation results show that the proposed system can effectively tackle these attacks by removing the possibility of false positive packets. However, communication delay that occurs when a switch communicates with a controller due to unavailability of memory space and for the DNS request validation, should be minimized. Moreover, further experiments need to be carried out in a real-world environment to study its effectiveness.

Another security architecture that protects the SDN controller from three types of security attacks, i.e., flow table overloading, DDoS attacks and link spoofing attacks, is presented in [107]. The proposed architecture uses multiple controllers in a star topology, instead of a single controller used in previous studies, to validate the user flows. A star topology is mainly chosen to mitigate the effects of flow table overloading attacks that occur due to anomalies. For validation of users flow and flow table occupancy, the proposed architecture employs Discrete-Time Finite-State Markov Chain (DTMC) model [36] at all switches in the network. This model provides updated information from time to time on the state, i.e., idle, busy/transmitting, of these switches. Moreover, the proposed architecture tackles the issue of link spoofing attack that occurs between the switches at the data layer by verifying these links using an unsupervised hashing method at the controller [121] Finally, a hybrid classifier is employed at the controller by combining fuzzy logic classifier with extreme learning machine (L1-ELM) running on a neural network. This hybrid approach is used because it proved to be an efficient classifier that classifies malicious packets from regular data packets. The proposed approach initially blocks and detects anomalies at the switches, while the remaining anomalies that escape those switches are identified and mitigated by the controller. Moreover, switches are informed by the controller about those escaped anomalies. Results from the simulation reveal that the proposed approach can effectively defend SDN networks against the aforementioned attacks. However, the network flow should be validated in large-scale networks.

Another vital contribution for SDN controller is presented in [75]. The authors have proposed a novel multi-tier component-based security architecture. The proposed architecture aims to protect Software-defined Mobile Networks (SDMNs) against IP-related attacks such as DoS, DDoS and IP spoofing attacks. The proposed architecture consists of five components, i.e., secure communication, policy-based communication, security information and event management, security-defined monitoring and deep packet inspection component. The role of the secure communication component is to protect the data-to-control plane channel using Host Identity Protocol (HIP) with IPSec tunneling. The policy-based communication component protects the network, associated channels and devices against DoS and source address spoofing, based on pre-defined policy. In the security management and monitoring component, Deep Packet Inspection (DPI) is carried out for the detection of vulnerabilities and security threats as well as checking the security mechanisms used in the underlying network. [107]. The responsibility of security-defined monitoring is to coordinate the monitoring activities. Finally, Deep Packet Inspection (DPI), is used to improve security threat detection. Simulation results prove that the proposed architecture can protect the network against IP-related attacks on SDMNs. However, its feasibility needs to be studied in real-world settings. Furthermore, the requirements and guidelines are not clear on how to integrate it with the current system.

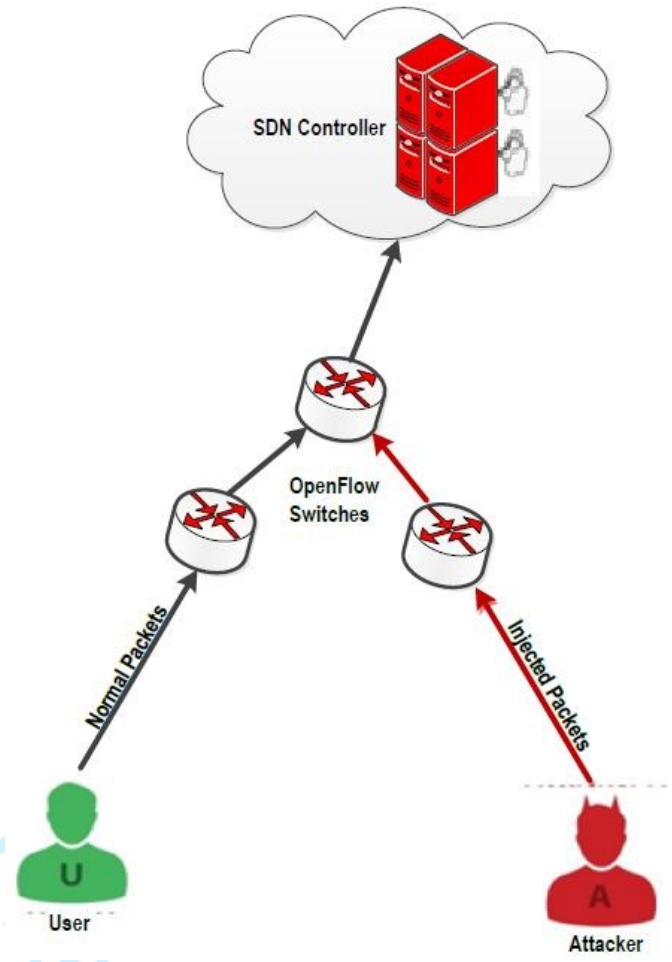

Fig. 7 Malicious injection attacks on the SDN Controller

3.3 Anomalies and Malicious Injection attacks on the SDN controller

Malicious injection attack is yet another type of attacks and remains an intruder's preferred choice to exploit various vulnerabilities in SDN-based networks, as shown in Fig. 7 [47]. During malicious injection attacks, an adversary seizes a single or multiple hosts for launching malicious packets. The only thing that an attacker requires is the same privileges as a normal user. On the other hand, during SQL injection attacks, a perpetrator modify the anticipated effect of an SQL query by injecting new SQL keywords or operators into the query. A detailed discussion on SQL injection attacks and their countermeasures can be found in [69] Recently, some research works have been carried out on the detection of anomalies as well as malicious injection attacks on an SDN controller [18], [72], [101], [103], [137], [148], [153], [168]. However, numerous challenges remain unaddressed due to the unique characteristics of SDN controller along with the varying nature of SDN traffic. In the following subsection, we present the aforementioned mitigation techniques and provide an overview of them in Table 3.

3.3.1 Mitigation Techniques for Anomalies and Malicious Injection attacks

A novel topology discovery protocol, Secure and Efficient OpenFlow Discovery Protocol (sOFTDP) was proposed in [18]. The design objectives of the sOFTDP were to overcome various operational and security limitations of the OpenFlow Discovery Protocol (OFDP) [111]. The proposed protocol is lightweight, dynamic and suggests minimal changes to the Open Flow switch design. Minimal changes to the open flow switch imply that sOFTDP shifts a part of the topology discovery procedure from the controller to a switch. As such, the switch alone detects the link events and notifies the controller, whenever necessary. A controller contains the necessary mechanism to deal with switch notifications and dynamically changes its topology map for making routing decisions without prior knowledge of the events that cause topological changes. The two main

Table 3 A Summary of mitigation techniques for Malicious Injection attacks on the SDN controller

Mitigation Techniques	**Issues Addressed**	**Strengths**	**Weaknesses**
sOFTDP [18]	Link Injection and Fabrication attacks	Secured, Improved performance and topology discovery time	Ineffective against link fabrication attacks in relay manner, and not evaluated on larger testbeds
EUNOIA [148]	IDS	Machine learning-based effective solution for intrusion detection system	Needs to be tested on multiple classifiers
RAD [92]	Anomalies detection	Enhances performance by detecting attacks with improved agility and efficiency, even during link failure and burst Traffic	Needs to be evaluated for complex traffic generation and in different attack scenarios
Synaptic [137]	Anomalies and vulnerabilities	Guard a security chain against anomalies, intrusion and vulnerabilities	Can be enhanced further using formal methods
ML-based IDS [5]	Detect and mitigate DoS, probe, U2R, and R2L attacks with an improved accuracy	A flow-based anomaly detection using machine learning techniques to overcome the limitation of signature-based IDS.	Low accuracy, i.e., high-false rate
Byzantine FL [103]	Protect data and control planes, and southbound interface against unauthorized access using multiple controllers	A cost effective controller assignment algorithm for a given set of switches	Lacks optimal controller assignment algorithm
Athena [101]	Detects well-known network anomalies in an efficient manner	Scalable anomaly detection framework requiring minimal programming effort and no specialized software	Controller exclusive, and overload the devices at the data plane during heavy traffic
Sampled-DP [72]	Anomalies detection	Cluster centers and outlier points extraction to eliminate redundancy	Low performance, and slow response towards intrusion
Dynamic Access Control system (DAC) [153]	API abuse attacks	Secured API requests with minimum latency	Limited in its scope because it is controller specific
Scalable Traffic Sampling [168]	Anomalies and malicious packet detection in large-scaled networks	Enhances monitoring and performance by selecting the most feasible switches for scalable traffic sampling	Does not consider complex topologies and attacks scenarios

events that cause topological changes are: adding new links to the network and removal of existing links from the network. The proposed protocol was implemented as a topology discovery protocol module in the floodlight controller for evaluation purposes. Simulation results showed that sOFTDP outperforms OFDP regarding security, performance and topology discovery time, respectively. Despite all these advantages, sOFTDP is unable to protect the network against the link fabrication attacks on relay nodes. Moreover, effectiveness of the proposed protocol should be tested in large and complex scenarios.

Another significant contribution is EUNOIA [148], a threat-aware system based on machine learning. The proposed system detects and mitigates network intrusion in four stages: data processing for feature selection, predictive data modelling for machine learning and anomaly detection, decision making for intrusion detection, and response system. During the first stage, redundant data is filtered out from a large volume of data that include raw data containing both historical archival traffic and real-time incoming traffic data. This stage reduces ambiguity from voluminous traffic gathered previously for the predictive data modelling subsystem with increased reliability. In the predictive data modelling stage, an attack model for intrusion detection is developed using decision-tree machine learning algorithms [146]. The classification algorithm is used to train a classifier that can label or predict any new unknown audit data, related to either relevant or irrelevant class. Once a classification model is developed, malicious data is identified in real-time with minimal overhead. In decision making for intrusion detection stage, redundancy and uncertainty in the previous stages are further enhanced using random forest machine learning algorithm [30] along with active learning technique [37]. This stage protects the controller against network intrusion while detecting anomalies with improved accuracy. In the final stage, i.e., the response system, EUNOIA employs a reactive routing and novel cost function. Simulation results show that the proposed system is lightweight and an efficient solution against the network intrusion detection. The drawback of EUONIA is that the cost function employed at the response system is straightforward and needs to be improved. Furthermore, EUNOIA exclusively considers a large volume of data. However, the effectiveness of the proposed methods should be studied and tested with varying sizes of data.

Robust and Agile Defense (RAD) system [92] is a reactive mechanism that guards the SDN controller against spoofing attacks while ensuring high availability and reliability of the underlying network. The design of RAD is based on three modules of a controller, i.e., a traffic analyzer, a traffic engineer, and a rule manager. Each of these modules has its role and responsibilities. The role of the traffic analyzer is to monitor the bursty and bandwidth-starving traffic flows using sampled flow real-time (sFlowRT) [140], a real-time monitoring tool. A signature-based intrusion detection system, i.e., snort IDS [147], is used to recognize anomalies and attacks, based on attack signatures. Traffic engineering module, on the other hand, monitor network utilization and delay for each link. The link incurring high cost regarding these two metrics are excluded from the route generation. The authors aim for the creation of a suitable route for multi-dimensional and load balancing data with improved efficiency. The cost function of the traffic engineering module controls the preference weight of these two sub-modules and is responsible for the normalization of various ranges of metric values. Finally, the rule manager module creates the flow rules for the data layer and selection of the best routes for regulating traffic using both reactive and proactive approaches with increased scalability. Simulation result shows that RAD can easily be integrated with the SDN controller and enhance its performance. However, the proposed system fails to detect anomalies. Furthermore, RAD should be evaluated using various topologies, attack scenarios using routing metrics other than the one used by the authors. The cost function used in the traffic engineering sub-module can be enhanced and evaluated for further studies.

Synaptic [137] is an automated method that performs automatic verification of security chains deployed at the control and data planes of an SDN architecture . Security chains consist of various security functions such as firewalls and intrusion detection systems and are responsible for the prevention of data leakage and any security violations. Due to the dynamic and complex nature of these security chains, it is essential to guard it against attacks, anomalies and possible intrusion. Synaptic built formal verification models from security chains using a frenetic family of the SDN programming language, particularly using pyretic language along with an extension called Kinetic [59], [60], [91]. The role of the pyretic language is to specify network configuration in Python, which is later on compiled into low-level rules. On the other hand, the Kinetic extension is used to define policies for the control layer. All these functions are combined to generate formal models based on security chain specification before its implementation. Translation specification algorithms, e.g. Satisfiability Modulo Theories (SMT) [29] is used at the data plane, while, Kinetic is used at the control plane to translate specification of security chains into

formal methods that can then be verified automatically. A prototype of the proposed approach is designed, and its performance is evaluated in term of response time and memory consumption with varying sizes of security chains using various validity checkers such as CVC4 [20], veriT [29], and nuXmv [33]. Simulation results show that the proposed system can be enhanced further by using various translation algorithms that support more complex rules, related to the said security functions. Furthermore, formal models that are generated by checked properties may be enhanced.

There are two types of Intrusion Detection System (IDS) [117], i.e., a Host intrusion detection system (HIDS) and a network intrusion detection system (NIDS). In [5], the authors proposed a two-stage Network Intrusion Detection System (NDIS). It is based on pattern recognition used for a neural network with machine learning approaches for the detection of signature and non-signature based attacks on an SDN controller. During the first stage, a virtual testbed is designed and developed to simulate the processes of a real network environment, using a star topology. Hosts with the server and vice versa are connected to the OpenFlow switch for the detection of signature-based attacks. In the second stage, non-signature based or unknown attacks are detected and are integrated with the signature-based architecture, designed for the previous stage. This hybrid approach can effectively detect both signature and non-signature based attacks. Based on the simulation results, it was concluded that the proposed architecture achieves its objectives with 97% detection accuracy. However, this detection accuracy may be enhanced further using other neural network techniques.

The Byzantine fault-tolerant mechanism [103] is used to secure the control plane, data plane and control-to-data plane interface against unauthorized access. The proposed mechanism manages each device at the data plane using multiple controllers [32]. Byzantine architecture ensures accurate updates of flow tables despite issuing false instructions by numerous compromised controllers. The authors have designed a cost-effective controller assignment algorithm, based on a heuristic, also known as Capacity First Allocation (CFA). The CFA ensures that an optimal number of controllers required for a given set of switches are maintained while satisfying their security requirements. The assignment algorithm serves two purposes. Firstly, the varying number of controllers needed for each switch and secondly, the number of switches served by a single controller. Thus the proposed algorithm ensure to employ an optimal number of controllers needed for a switch, based on the minimum residual capacity of the controllers, while, keeping in view the cost of deploying these controllers. Furthermore, Byzantine mechanism exchanges various messages between a set of controllers connected with a switch. Its performance is hugely dependent on the link latency among these controllers. The proposed algorithm is extensively simulated using various scenarios. Based on the simulations results, it was concluded that the proposed mechanism efficiently assigns the controllers based on the requirements for a given set of switches. The performance of the proposed architecture may be evaluated further using algorithms other than the one used in this paper. Furthermore, the proposed architecture may be studied further in more complex scenarios under different network conditions.

Another significant contribution is Athena [101], an integrated, scalable and distributed framework to support sophisticated anomaly detection across the control and data planes of an SDN controller. Athenas API offers the developers an abstraction from a complex data extraction service, with minimal programming effort while implementing and adding new and third party anomaly detection services to the SDN stack. This is because the proposed architecture does not require any specialized software except OpenFlow support. Compared with the previously available solutions, the proposed architecture include a variety of network features and detection algorithms for use in simplifying the design and deployment of general-purpose data plane-based anomaly detection framework in large-scale SDN networks. In term of scalability, the network feature collection and data management of the proposed architecture uses a distributed database, a computing cluster and a distributed controller. Network features are generated and collected above the controller instances in a distributed manner, and the same is published to the database. To speed up the runtime detection, Athena integrates a machine learning library and an anomaly detection algorithms, which is installed in the form of jobs at the computing cluster. Simulation results of the proposed architecture reveal that it can efficiently support well-known network anomaly detection services. However, the performance of the proposed method can be enhanced further using high performance distributed databases such as Cassandra, instead of MongoDB used by Athena.

Sampled-DP [72] is yet another essential anomaly detection framework. It is the combination of two algorithms, namely density peak-based clustering algorithm with sampling adaptation and an unsupervised cluster-based feature selection mechanism. The density peak-based clustering algorithm with sampling adaptation algorithm automatically extracts the cluster centers and outlier points with increased memory and time efficiency, as opposed to the other such clustering meth-

ods in the literature. On the other hand, the second algorithm groups together attributes having maximum redundancy and remove them for feature selection purposes. The performance of sampled-DP is evaluated using KDDCup99 dataset [90]. It was concluded that the proposed framework outperforms the existing algorithms in term of runtime, adjusted mutual information, homogeneity score and detection accuracy. However, the sampled-DP needs to be improved further in such a way that it performs real-time packet clustering to protect the network much quicker against any possible intrusion. Furthermore, the performance of sampled-DP is dataset-dependent. Therefore, its behavior needs to be studied using complex datasets having a variety of features.

In [153], the authors proposed a controller-specific Dynamic Access Control system (DAC). According to the authors, the static permission does not protect the controller against API abuse. Therefore, DAC employs four dynamic permission controls: read, add, update and remove, for authorizing an Open Flow app to access an SDN controller. The DAC consists of three components, i.e., a high-level policy engine, a northbound security extension between the control layer and application layer, and a controller-specific IDS. The high-level policy engine pre-defines and validates every request for each Open Flow application, according to the aforementioned permission set. The northbound security extension is responsible for authenticating and authorizing every request from Open Flow app as well as validating every request, using accounting records provided by the controller-specific IDS. This extension employs a token-based and password-based authentication. The controller-specific IDS, on the other hand, is responsible for the collection of information related to permission choices and accounting records from the database and high-level policy engine. Controller DAC is used on the top of four open source SDN controllers including OpenDaylight, ONOS, Floodlight and Ryu. Simulation results show that the proposed methods can effectively and efficiently protect an SDN controller from API abuse vulnerabilities with a minor latency of less than 0.5%, using dynamic access control, as opposed to static control.

One of the challenging issues is how to select a set of switches for data sampling in an SDN with increased reliability and scalability. The authors in [168] tackle this issue by selecting switches with the relatively higher importance that improve the monitoring and performance of the IDS as well as reduces the chances of congestion in the network. The author has used a packet sampling technique by capturing only selected packets from the traffic flow at the selected set of switches for monitoring purposes. To overcome this issue, The authors employ a centrality measure, based on graph theory, as a packet sampling technique for the selection of packet sampling points among the switches in the network. It selects the sampling points based on the number of shortest paths that pass through a switch for all the node pairs [62]. Once a decision regarding data sampling points is confirmed, a decision sampling rate is made. Packets are sampled at the selected switches at a specific rate. The sampled data are then forwarded to an IDS for further analysis of malicious traffic and anomalies. Simulation results demonstrate that the proposed method enhances the performance of an IDS by efficiently capturing anomalies in a large-scale network. However, further experiments need to be carried out using sampling point techniques, other than the one employed in this paper.

4 Open Research Issues, Challenges and Directions

The centralized and programmable nature of an SDN makes it a promising, innovative and adaptable technology. It has simplified many issues related to the control and management of the next generation networks [12], [131]. However, among others, security provisioning remains one of the major issue that hinder its widespread adoption [23] [135]. This is particularly due to the centralized controller, which is responsible for the control and managemet of the overall network [177]. According to Robert Hinden [74], "why take over the hosts when you can take over the whole network?". It is therefore imperative to consider security as a service while designing an SDN network in general, and the controller in particular [8]. To ensure security of the SDN controller, novel and innovative mechanisms need to be developed that are computationally lightweight and can efficiently and effectively identify malicious traffic [96], [150]. Moreover, further studies should be carried out to study the characteristics of southbound interface (SBI) which facilitates communication between a controller and various network components at the data layer, i.e., flow rules installation [38], [24], network configuration [143], traffic statistics [46], and optimal path selection [13]. These challenges should be addressed to protect the controller against malicious traffic that later on results in various security attacks such as DoS/DDos, spoofing and malicious injection attacks [14], [110], [23]. Furthermore, such malicious traffic can disrupt various network operations required for the smooth functioning of a network. There is no built-in security mechanism at the southbound interface. This provides an ideal platform for adversaries to launch various attacks on the

SDN controllers [23], [46]. It is thus imperative to develop novel link scanning mechanisms, similar to [115], to identify whether congestion on the southbound interface is caused by normal traffic or malicious traffic from an attacker.

The SDN controller is highly vulnerable to malicious traffic, particular during flooding attacks. These attacks disrupt various functions and services offered by the network in general and controller in particular to the legitimate users. In the literature, a threshold value is used to detect and mitigate such attacks and proved to be very effective [43], [162], [126]. Once the traffic at the network crosses that specific threshold value, flooding attack on the network is detected, and an alarm is raised. However, attacks such as flash crowd-based attacks [21] bypass these threshold-based mechanisms and are thus remain undetected. This is because the attackers craft their traffic to look legitimate and may use spoofed addresses to launch such attacks by making it very difficult to combat. Therefore, more research is needed that not only detect attacks on the basis of threshold value but also on novice IP address filtering techniques [41], [102], and deep packet inspection [142], [169], [28] that detect malicious packets with lower delay and increased accuracy [113], [44]. The researchers should strive for novice and computationally lightweight traffic flow analysis techniques that analyze the characteristics of Open Flow traffic stored in switches at the data layer to identify the malicious traffic. Therefore, early detection of DDoS becomes an important area for further exploration [110].

In SDN, real-time monitoring needs to be performed for the identification of malicious traffic with increased accuracy, primarily during large-scale DoS/DDoS attacks. Thus, accuracy is a major concern of various DDoS mitigation techniques [3], [120], [57]. It can help us to determine how accurately the system can detect the occurrence or absence of an attack. For instance, an inaccurate technique can generate a large number of false alarms by making the traffic flow through the system unnoticed. It is thus important to quickly react to an attack by detecting it with high precision, particularly, during heavy traffic from the perpetrator in DDoS attacks. In addition, most of the existing DoS mitigation techniques incur higher communication and processing overhead during the detection stage in which the data from network components at the data layer are collected and transferred to the controller for decision making [132] [105] [123] [31]. It is thus necessary to design novel monitoring and response mechanisms with lower computational overhead that alert the network component, detecting the attacker traffic, and safeguarding the network from such attacks [134], [39]. In this regard, some of the response mechanisms that need to be looked further in detail includes high enforced policy module with predefined actions [109], identity management [15], certification from multiple authorities [133], threat isolation and public key encryption of data [23]. Establishing an automatic trust management [175], [151] and mutual authentication [155], among the communicating entities should be supported to protect the network from various known and unknown attacks [23]. Scalability is another important issue that needs to enhance in particular during the DDoS attack [164], [150], [125]. With the increasing rate, volume and number of bots used in these attacks, defense mechanisms should be able to perform effectively in the presence of a large attack. In these situations, automatic network slicing is an effective technique for securing controllers against DDoS in distributed environments [50]. However, issues such as inconsistent configuration / synchronization [169], [128], [23] is a hidden security threat and need further research.

In addition, security of the existing mechanisms such as forensic remediation [89], [127], verification framework[17], proactive and reactive recovery mechanisms [108], [22] [1], resilient control planes [65], [108], [53], [156], [141], SDN security framework [171], [98], state machine replication [119], [171], [118], dynamic, automated and assured device association [172] should be enhanced to cater for the known and unknown attacks in the future. On the other hand, malicious packets can affect the network view that results in flow rule conflicts and policy violations. It is thus necessary to propose novel techniques that enforce the security by tackling the issue of policy violation in SDNs [81], [125], [162]. Moreover, for the effective mitigation of malware injection attacks, defense mechanisms such as N-version protection [174] should be investigated further.

5 Conclusions

Despite the hype surrounding SDN, its security-related issues have not been explored in depth. It is because security was not considered initially, despite its core value in an SDN design. It is due to various aforementioned security issue among many other security dimensions that hinder its widespread adoption. It is thus imperative to tackle various security dimensions to make it a promising and secure technology, otherwise, its benefits might be quickly overcome by its security concerns. It is therefore, utmost important to consider security as essential part while designing an SDN. In this paper, we have provided a comprehensive and critical discussion on the latest security attacks that exclusively target SDN controller along with a detailed discussion

on their mitigation techniques. Unlike, previously related surveys that are either specific to a particular type of attack or discuss attacks on the whole SDN architecture. Furthermore, This paper provides guidance to the readers on the suitability of a particular mitigation technique for a particular scenario. Moreover, future directions and suggestions for securing SDN architecture in general and the control layer in particular are provided. From this survey, We have concluded that conducive efforts are needed to explore innovative methods for ensuring security of the SDN controller due to single point dependency. It is expected that more research efforts should be carried out to validate elements address, establish trust mechanisms between all elements, ensuring strong encryption, authentication and access control among devices. Research should focus on designing highly accurate attack detection model, novice monitoring methods that report the sudden increase in packet-in events in advance.Unfortunately, cyber attacks may be detected, but after the damage is done. Therefore, developing a cyber system that can survive an attack is a challenge. However,Our effort is to complement existing surveys and stimulate more research studies in this area in order to make SDN, a secure, trustful and dependable architecture in the years to come.

6 Acknowledgements

The author Syed Rooh Ullah Jan is the equal first co-author of this paper. This work was supported in part by the International Scientific and Technological Cooperation Project of Dongguan (2016508102011) and in part by Science and Technology Planning Project of Guangdong Province (2016A020210142).

References

1. Sarah Abdallah, Imad H Elhajj, Ali Chehab, and Ayman Kayssi. A network management framework for sdn. In *New Technologies, Mobility and Security (NTMS), 2018 9th IFIP International Conference on*, pages 1–4. IEEE, 2018.
2. Ahmed M AbdelSalam, Ashraf B El-Sisi, and Vamshi Reddy. Mitigating arp spoofing attacks in software-defined networks.
3. Ihsan Abdulqadder, Deqing Zou, Israa Aziz, Bin Yuan, and Weiqi Dai. Deployment of robust security scheme in sdn based 5g network over nfv enabled cloud environment. *IEEE Transactions on Emerging Topics in Computing*, 2018.
4. Ihsan H Abdulqadder, Deqing Zou, Israa T Aziz, and Bin Yuan. Validating user flows to protect software defined network environments.
5. Atiku Abubakar and Bernardi Pranggono. Machine learning based intrusion detection system for software defined networks. In *Emerging Security Technologies (EST), 2017 Seventh International Conference on*, pages 138–143. IEEE, 2017.
6. Ijaz Ahmad, Suneth Namal, Mika Ylianttila, and Andrei Gurtov. Security in software defined networks: A survey. *IEEE Communications Surveys & Tutorials*, 17(4):2317–2346, 2015.
7. Adnan Akhunzada, Ejaz Ahmed, Abdullah Gani, Muhammad Khurram Khan, Muhammad Imran, and Sghaier Guizani. Securing software defined networks: taxonomy, requirements, and open issues. *IEEE Communications Magazine*, 53(4):36–44, 2015.
8. Adnan Akhunzada, Abdullah Gani, Nor Badrul Anuar, Ahmed Abdelaziz, Muhammad Khurram Khan, Amir Hayat, and Samee U Khan. Secure and dependable software defined networks. *Journal of Network and Computer Applications*, 61:199–221, 2016.
9. Mahmoud Al-Ayyoub, Yaser Jararweh, Elhadj Benkhelifa, Mladen Vouk, Andy Rindos, et al. Sdsecurity: A software defined security experimental framework. In *Communication Workshop (ICCW), 2015 IEEE International Conference on*, pages 1871–1876. IEEE, 2015.
10. Malek Al-Zewairi, Dima Suleiman, and Sufyan Almajali. An experimental software defined security controller for software defined network. In *Software Defined Systems (SDS), 2017 Fourth International Conference on*, pages 32–36.
11. Talal Alharbi, Marius Portmann, and Farzaneh Pakzad. The (in) security of topology discovery in software defined networks. In *Local Computer Networks (LCN), 2015 IEEE 40th Conference on*, pages 502–505. IEEE, 2015.
12. Syed Riffat Ali. Software defined networking (sdn). In *Next Generation and Advanced Network Reliability Analysis*, pages 105–130. Springer, 2019.
13. Lylia Alouache, Nga Nguyen, Makhlouf Aliouat, and Rachid Chelouah. Survey on iov routing protocols: Security and network architecture. *International Journal of Communication Systems*, 32(2):e3849, 2019.
14. Abdullah Soliman Alshraa and Jochen Seitz. Using inspector device to stop packet injection attack in sdn. *IEEE Communications Letters*, 2019.
15. Izzat Alsmadi. Identity management. In *The NICE Cyber Security Framework*, pages 313–329. Springer, 2019.
16. Izzat Alsmadi and Dianxiang Xu. Security of software defined networks: A survey. *computers & security*, 53:79–108, 2015.
17. Rashid Amin, Martin Reisslein, and Nadir Shah. Hybrid sdn networks: A survey of existing approaches. *IEEE Communications Surveys & Tutorials*, 2018.
18. Abdelhadi Azzouni, Raouf Boutaba, Nguyen Thi Mai Trang, and Guy Pujolle. softdp: Secure and efficient topology discovery protocol for sdn. *arXiv preprint arXiv:1705.04527*, 2017.
19. Christian Banse and Sathyanarayanan Rangarajan. A secure northbound interface for sdn applications. In *Trustcom/BigDataSE/ISPA, 2015 IEEE*, volume 1, pages 834–839. IEEE, 2015.
20. Clark W Barrett, Roberto Sebastiani, Sanjit A Seshia, Cesare Tinelli, et al. Satisfiability modulo theories. *Handbook of satisfiability*, 185:825–885, 2009.
21. Narmeen Zakaria Bawany, Jawwad A Shamsi, and Khaled Salah. Ddos attack detection and mitigation using sdn: Methods, practices, and solutions. *Arabian*

Journal for Science and Engineering, 42(2):425–441, 2017.

22. Jaouad Benabbou, Khalid Elbaamrani, and Noureddine Idboufker. Security in openflow-based sdn, opportunities and challenges. *Photonic Network Communications*, pages 1–23, 2018.
23. Jaouad Benabbou, Khalid Elbaamrani, and Noureddine Idboufker. Security in openflow-based sdn, opportunities and challenges. *Photonic Network Communications*, 37(1):1–23, 2019.
24. Samaresh Bera, Sudip Misra, and Abbas Jamalipour. Flowstat: Adaptive flow-rule placement for per-flow statistics in sdn. *IEEE Journal on Selected Areas in Communications*, 2019.
25. Lionel Bertaux, Akram Hakiri, Samir Medjiah, Pascal Berthou, and Slim Abdellatif. A dds/sdn based communication system for efficient support of dynamic distributed real-time applications. In *Distributed Simulation and Real Time Applications (DS-RT), 2014 IEEE/ACM 18th International Symposium on*, pages 77–84. IEEE, 2014.
26. Uri Blumenthal, Fabio Maino, and Keith McCloghrie. The advanced encryption standard (aes) cipher algorithm in the snmp user-based security model. Technical report, 2004.
27. Julien Boite, Pierre-Alexis Nardin, Filippo Rebecchi, Mathieu Bouet, and Vania Conan. Statesec: Stateful monitoring for ddos protection in software defined networks. In *Network Softwarization (NetSoft), 2017 IEEE Conference on*, pages 1–9. IEEE, 2017.
28. Michel S Bonfim, Kelvin L Dias, and Stenio FL Fernandes. Integrated nfv/sdn architectures: A systematic literature review. *ACM Computing Surveys (CSUR)*, 51(6):114, 2019.
29. Thomas Bouton, Diego Caminha B de Oliveira, David Déharbe, and Pascal Fontaine. verit: an open, trustable and efficient smt-solver. In *International Conference on Automated Deduction*, pages 151–156. Springer, 2009.
30. L Breiman. Random forests machine learning. 45: 5–32. *View Article PubMed/NCBI Google Scholar*, 2001.
31. Krzysztof Cabaj, Marcin Gregorczyk, Wojciech Mazurczyk, Piotr Nowakowski, and Piotr Żórawski. Network threats mitigation using software-defined networking for the 5g internet of radio light system. *Security and Communication Networks*, 2019, 2019.
32. Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems (TOCS)*, 20(4):398–461, 2002.
33. Roberto Cavada, Alessandro Cimatti, Michele Dorigatti, Alberto Griggio, Alessandro Mariotti, Andrea Micheli, Sergio Mover, Marco Roveri, and Stefano Tonetta. The nuxmv symbolic model checker. In *International Conference on Computer Aided Verification*, pages 334–342. Springer, 2014.
34. Yuanyuan Chai, Limin Jia, and Zundong Zhang. Mamdani model based adaptive neural fuzzy inference system and its application. *International Journal of Computational Intelligence*, 5(1):22–29, 2009.
35. Xiangle Cheng, Yulei Wu, Geyong Min, and Albert Y Zomaya. Network function virtualization in dynamic networks: A stochastic perspective. *IEEE Journal on Selected Areas in Communications*, 2018.
36. Gianfranco Ciardo. Discrete-time markovian stochastic petri nets. In *Computations with Markov Chains*, pages 339–358. Springer, 1995.
37. David A Cohn, Zoubin Ghahramani, and Michael I Jordan. Active learning with statistical models. *Journal of artificial intelligence research*, 1996.
38. Douglas Comer and Adib Rastegarnia. Towards disaggregating the sdn control plane. *arXiv preprint arXiv:1902.00581*, 2019.
39. Jie Cui, Qinghe Lu, Hong Zhong, Miaomiao Tian, and Lu Liu. A load-balancing mechanism for distributed sdn control plane using response time. *IEEE Transactions on Network and Service Management*, 15(4):1197–1206, 2018.
40. Marc C Dacier, Sven Dietrich, Frank Kargl, and Hartmut König. Overview of talks 4.1 network monitoring & sdn. *Network Attack Detection and Defense-Security Challenges and Opportunities of Software-Defined Networking*, page 7, 2017.
41. Nhu-Ngoc Dao, Junho Park, Minho Park, and Sungrae Cho. A feasible method to combat against ddos attack in sdn network. In *2015 International Conference on Information Networking (ICOIN)*, pages 309–311. IEEE, 2015.
42. Tooska Dargahi, Alberto Caponi, Moreno Ambrosin, Giuseppe Bianchi, and Mauro Conti. A survey on the security of stateful sdn data planes. *IEEE Communications Surveys & Tutorials*, 19(3):1701–1725, 2017.
43. Jisa David and Ciza Thomas. Efficient ddos flood attack detection using dynamic thresholding on flow-based network traffic. *Computers & Security*, 2019.
44. Hubert DCruze, Ping Wang, Raed Omar Sbeit, and Andrew Ray. A software-defined networking (sdn) approach to mitigating ddos attacks. In *Information Technology-New Generations*, pages 141–145. Springer, 2018.
45. Marco De Vivo, Gabriela O de Vivo, and Germinal Isern. Internet security attacks at the basic levels. *ACM SIGOPS operating systems review*, 32(2):4–15, 1998.
46. Ahmed Demirpolat, Doğanalp Ergenç, Esref Ozturk, Yusuf Ayar, and Ertan Onur. Software-defined network security. In *Enabling Technologies and Architectures for Next-Generation Networking Capabilities*, pages 232–253. IGI Global, 2019.
47. Shuhua Deng, Xing Gao, Zebin Lu, and Xieping Gao. Packet injection attack and its defense in software-defined networks. *IEEE Transactions on Information Forensics and Security*, 13(3):695–705, 2018.
48. Tim Dierks. The transport layer security (tls) protocol version 1.2. 2008.
49. C Dillon and Michael Berkelaar. Openflow (d) dos mitigation, 2014.
50. Sinh Do, Luong Vy Le, Bao Shuh Paul Lin, and Li-Ping Tung. Sdn/nfv based internet of things for multi-tenant networks. *Transactions on Networks and Communications*, 6(6):40, 2019.
51. Avri Doria, J Hadi Salim, Robert Haas, Horzmud Khosravi, Weiming Wang, Ligang Dong, Ram Gopal, and Joel Halpern. Forwarding and control element separation (forces) protocol specification. Technical report, 2010.
52. Sergei Dotcenko, Andrei Vladyko, and Ivan Letenko. A fuzzy logic-based information security management for software-defined networks. In *Advanced Communication Technology (ICACT), 2014 16th International Conference on*, pages 167–171. IEEE, 2014.
53. Amit Dvir, Yoram Haddad, and Aviram Zilberman. Wireless controller placement problem. In *Consumer Communications & Networking Conference (CCNC), 2018 15th IEEE Annual*, pages 1–4. IEEE, 2018.

54. Levent Ertaul and Krishnakumar Venkatachalam. Security of software defined networks (sdn).
55. Zhijie Fan, Ya Xiao, Amiya Nayak, and Chengxiang Tan. An improved network security situation assessment approach in software defined networks. *Peer-to-Peer Networking and Applications*, pages 1–15, 2017.
56. Hamid Farhady, HyunYong Lee, and Akihiro Nakao. Software-defined networking: A survey. *Computer Networks*, 81:79–95, 2015.
57. Lyndon Fawcett, Sandra Scott-Hayward, Matthew Broadbent, Andrew Wright, and Nicholas Race. Tennison: A distributed sdn framework for scalable network security. *IEEE Journal on Selected Areas in Communications*, 36(12):2805–2818, 2018.
58. Pierdomenico Fiadino, Alessandro D'Alconzo, Mirko Schiavone, and Pedro Casas. Challenging entropy-based anomaly detection and diagnosis in cellular networks. In *ACM SIGCOMM Computer Communication Review*, volume 45, pages 87–88. ACM, 2015.
59. Nate Foster, Arjun Guha, Mark Reitblatt, Alec Story, Michael J Freedman, Naga Praveen Katta, Christopher Monsanto, Joshua Reich, Jennifer Rexford, Cole Schlesinger, et al. Languages for software-defined networks. *IEEE Communications Magazine*, 51(2):128–134, 2013.
60. Nate Foster, Rob Harrison, Michael J Freedman, Christopher Monsanto, Jennifer Rexford, Alec Story, and David Walker. Frenetic: A network programming language. *ACM Sigplan Notices*, 46(9):279–291, 2011.
61. Jerome Francois and Olivier Festor. Anomaly traceback using software defined networking. In *Information Forensics and Security (WIFS), 2014 IEEE International Workshop on*, pages 203–208. IEEE, 2014.
62. Linton C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.
63. Open Networking Fundation. Software-defined networking: The new norm for networks. *ONF White Paper*, 2:2–6, 2012.
64. Robin Gandhi, Anup Sharma, William Mahoney, William Sousan, Qiuming Zhu, and Phillip Laplante. Dimensions of cyber-attacks: Cultural, social, economic, and political. *IEEE Technology and Society Magazine*, 30(1):28–38, 2011.
65. Fida Gillani, Ehab Al-Shaer, and Qi Duan. In-design resilient sdn control plane and elastic forwarding against aggressive ddos attacks. In *Proceedings of the 5th ACM Workshop on Moving Target Defense*, pages 80–89. ACM, 2018.
66. Kannan Govindarajan, Kong Chee Meng, and Hong Ong. A literature review on software-defined networking (sdn) research topics, challenges and solutions. In *Advanced Computing (ICoAC), 2013 Fifth International Conference on*, pages 293–299. IEEE, 2013.
67. Akram Hakiri, Aniruddha Gokhale, Pascal Berthou, Douglas C Schmidt, and Thierry Gayraud. Software-defined networking: Challenges and research opportunities for future internet. *Computer Networks*, 75:453–471, 2014.
68. Evangelos Haleplidis, Kostas Pentikousis, Spyros Denazis, J Hadi Salim, David Meyer, and Odysseas Koufopavlou. Software-defined networking (sdn): Layers and architecture terminology. Technical report, 2015.
69. William G Halfond, Jeremy Viegas, Alessandro Orso, et al. A classification of sql-injection attacks and countermeasures. In *Proceedings of the IEEE International Symposium on Secure Software Engineering*, volume 1, pages 13–15. IEEE, 2006.
70. R. Halili. "network security and spoofing attacks". Febuary 2018. www.pecb.com.
71. Muhammad Reazul Haque, Saw C Tan, Zulfadzli Yusoff, Ching K Lee, and Rizaludin Kaspin. Ddos attack monitoring using smart controller placement in software defined networking architecture. In *Computational Science and Technology*, pages 195–203. Springer, 2019.
72. Daojing He, Sammy Chan, Xiejun Ni, and Mohsen Guizani. Software-defined-networking-enabled traffic anomaly detection and mitigation. *IEEE Internet of Things Journal*, 4(6):1890–1898, 2017.
73. Henry Hendrix. Viterbi decoding techniques in the tms320c54x family. *Texas Instruments, June*, 1996.
74. Robert M Hinden. Why take over the hosts when you can take over the network. In *RSA Conference*, pages 1–41, 2014.
75. HIP. "the open hip project", February 2018. www.openhip.org.
76. Sungmin Hong, Lei Xu, Haopei Wang, and Guofei Gu. Poisoning network visibility in software-defined networks: New attacks and countermeasures. In *NDSS*, volume 15, pages 8–11, 2015.
77. Fei Hu. *Network Innovation through OpenFlow and SDN: Principles and Design*. CRC Press, 2014.
78. Fei Hu, Qi Hao, and Ke Bao. A survey on software-defined network and openflow: From concept to implementation. *IEEE Communications Surveys & Tutorials*, 16(4):2181–2206, 2014.
79. Haojun Huang, Hao Yin, Geyong Min, Hongbo Jiang, Junbao Zhang, and Yulei Wu. Data-driven information plane in software-defined networking. *IEEE Communications Magazine*, 55(6):218–224, 2017.
80. Xueli Huang, Xiaojiang Du, and Bin Song. An effective ddos defense scheme for sdn. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017.
81. Amani Abu Jabal, Maryam Davari, Elisa Bertino, Christian Makaya, Seraphin Calo, Dinesh Verma, Alessandra Russo, and Christopher Williams. Methods and tools for policy analysis. *ACM Computing Surveys (CSUR)*, 51(6):121, 2019.
82. Nachikethas A Jagadeesan and Bhaskar Krishnamachari. Software-defined networking paradigms in wireless networks: A survey. *ACM Computing Surveys (CSUR)*, 47(2):27, 2015.
83. Raj Jain and Subharthi Paul. Network virtualization and software defined networking for cloud computing: a survey. *IEEE Communications Magazine*, 51(11):24–31, 2013.
84. Mian Ahmad Jan, Muhammad Usman, Xiangjian He, and Ateeq Ur Rehman. Sams: A seamless and authorized multimedia streaming framework for wmsn-based iomt. *IEEE Internet of Things Journal*, 2018.
85. Yosr Jarraya, Taous Madi, and Mourad Debbabi. A survey and a layered taxonomy of software-defined networking. *IEEE Communications Surveys & Tutorials*, 16(4):1955–1980, 2014.
86. Mobin Javed, Ayesha Binte Ashfaq, M Zubair Shafiq, and Syed Ali Khayam. On the inefficient use of entropy for anomaly detection. In *RAID*, pages 369–370. Springer, 2009.
87. Jack L Johnson. Design of experiments and progressively sequenced regression are combined to achieve minimum data sample size. *International Journal of Hydromechatronics*, 1(3):308–331, 2018.
88. Kallol Krishna Karmakar, Vijay Varadharajan, and Udaya Tupakula. Mitigating attacks in software defined

network (sdn). In *Software Defined Systems (SDS), 2017 Fourth International Conference on*, pages 112–117. IEEE, 2017.

89. Thejaswini Kasaraneni. A survey on software-defined wireless sensor networks: Challenges and design requirements.
90. KDDCup. "kddcup99 dataset)", January 1999. www.kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
91. Hyojoon Kim, Joshua Reich, Arpit Gupta, Muhammad Shahbaz, Nick Feamster, and Russell J Clark. Kinetic: Verifiable dynamic network control. In *NSDI*, pages 59–72, 2015.
92. Mihui Kim, Younghee Park, and Rohit Kotalwar. Robust and agile system against fault and anomaly traffic in software defined networks. *Applied Sciences*, 7(3):266, 2017.
93. Soyoung Kim, Sora Lee, Geumhwan Cho, Muhammad Ejaz Ahmed, Jaehoon Paul Jeong, and Hyoungshick Kim. Preventing dns amplification attacks using the history of dns queries with sdn. In *European Symposium on Research in Computer Security*, pages 135–152. Springer, 2017.
94. Rowan Kloti, Vasileios Kotronis, and Paul Smith. Openflow: A security analysis. In *Network Protocols (ICNP), 2013 21st IEEE International Conference on*, pages 1–6. IEEE, 2013.
95. Teuvo Kohonen. The self-organizing map. *Neurocomputing*, 21(1-3):1–6, 1998.
96. R Koning, B de Graaff, G Polevoy, R Meijer, C de Laat, and P Grosso. Measuring the efficiency of sdn mitigations against attacks on computer infrastructures. *Future Generation Computer Systems*, 91:144–156, 2019.
97. Diego Kreutz, Fernando MV Ramos, Paulo Esteves Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76, 2015.
98. Diego Kreutz, Jiangshan Yu, Paulo Esteves-Veríssimo, Cátia Magalhães, and Fernando MV Ramos. The kiss principle in software-defined networking: a framework for secure communications. *IEEE Security & Privacy*, 16(5):60–70, 2018.
99. Anukool Lakhina, Mark Crovella, and Christophe Diot. Mining anomalies using traffic feature distributions. In *ACM SIGCOMM Computer Communication Review*, volume 35, pages 217–228. ACM, 2005.
100. Adrian Lara, Anisha Kolasani, and Byrav Ramamurthy. Network innovation using openflow: A survey. *IEEE communications surveys & tutorials*, 16(1):493–512, 2014.
101. Seunghyeon Lee, Jinwoo Kim, Seungwon Shin, Phillip Porras, and Vinod Yegneswaran. Athena: A framework for scalable anomaly detection in software-defined networks. In *Dependable Systems and Networks (DSN), 2017 47th Annual IEEE/IFIP International Conference on*, pages 249–260. IEEE, 2017.
102. Cheng Li, Zhengrui Qin, Ed Novak, and Qun Li. Securing sdn infrastructure of iot–fog networks from mitm attacks. *IEEE Internet of Things Journal*, 4(5):1156–1164, 2017.
103. He Li, Peng Li, Song Guo, and Amiya Nayak. Byzantine-resilient secure software-defined networks with multiple controllers in cloud. *IEEE Transactions on Cloud Computing*, 2(4):436–447, 2014.
104. Wenjuan Li, Weizhi Meng, et al. A survey on openflow-based software defined networks: Security challenges and countermeasures. *Journal of Network and Computer Applications*, 68:126–139, 2016.
105. Gang Liu, Wei Quan, Nan Cheng, Hongke Zhang, and Shui Yu. Efficient ddos attacks mitigation for stateful forwarding in internet of things. *Journal of Network and Computer Applications*, 2019.
106. Jing Liu, Yingxu Lai, and Shixuan Zhang. Fl-guard: A detection and defense system for ddos attack in sdn. In *Proceedings of the 2017 International Conference on Cryptography, Security and Privacy*, pages 107–111. ACM, 2017.
107. Madhusanka Liyanage, Ijaz Ahmed, Jude Okwuibe, Mika Ylianttila, Hammad Kabir, Jesus Llorente Santos, Raimo Kantola, Oscar López Perez, Mikel Uriarte Itzazelaia, and Edgardo Montes De Oca. Enhancing security of software defined mobile networks. *IEEE Access*, 5:9422–9438, 2017.
108. Carmen Mas Machuca, Petra Vizarreta, Raphael Durner, Dorabella Santos, and Amaro de Sousa. Design problems towards reliable sdn networks. In *Photonic Networks and Devices*, pages NeM2F–1. Optical Society of America, 2018.
109. Hellen Maziku, Sachin Shetty, and David M Nicol. Security risk assessment for sdn-enabled smart grids. *Computer Communications*, 133:1–11, 2019.
110. David McGrew and Kenneth S Beck. Inspection of traffic via sdn, February 12 2019. US Patent App. 10/205,641.
111. Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, 2008.
112. Jim Metzler. Understanding software-defined networks. *InformationWeek Reports*, pages 1–25, 2012.
113. Wang Miao, Geyong Min, Yulei Wu, Haojun Huang, Zhiwei Zhao, Haozhe Wang, and Chunbo Luo. Stochastic performance analysis of network function virtualisation in future internet. *IEEE Journal on Selected Areas in Communications*, 2019.
114. Wang Miao, Geyong Min, Yulei Wu, Haozhe Wang, and Jia Hu. Performance modelling and analysis of software-defined networking under bursty multimedia traffic. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 12(5s):77, 2016.
115. Jelena Mirkovic and Peter Reiher. A taxonomy of ddos attack and ddos defense mechanisms. *ACM SIGCOMM Computer Communication Review*, 34(2):39–53, 2004.
116. Seyed Mohammad Mousavi and Marc St-Hilaire. Early detection of ddos attacks against sdn controllers. In *Computing, Networking and Communications (ICNC), 2015 International Conference on*, pages 77–81. IEEE, 2015.
117. Srinivas Mukkamala, Guadalupe Janoski, and Andrew Sung. Intrusion detection using neural networks and support vector machines. In *Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on*, volume 2, pages 1702–1707. IEEE, 2002.
118. Amit Nayyer, Aman Kumar Sharma, and Lalit K Awasthi. Issues in software-defined networking. In *Proceedings of 2nd International Conference on Communication, Computing and Networking*, pages 989–997. Springer, 2019.
119. Binh Nguyen, Tian Zhang, Bozidar Radunovic, Ryan Stutsman, Thomas Karagiannis, Jakub Kocur, and Jacobus Van der Merwe. A reliable distributed cellular

core network for hyper-scale public clouds. Technical report, Technical Report. https://www. microsoft. com/en-us/research/uploads/prod/2018/02/ECHO-TR. pdf, 2018.

120. Tam N Nguyen. The challenges in sdn/ml based network security: A survey. *arXiv preprint arXiv:1804.03539*, 2018.
121. Tri-Hai Nguyen and Myungsik Yoo. Analysis of link discovery service attacks in sdn controller. In *Information Networking (ICOIN), 2017 International Conference on*, pages 259–261. IEEE, 2017.
122. Tri-Hai Nguyen and Myungsik Yoo. A hybrid prevention method for eavesdropping attack by link spoofing in software-defined internet of things controllers. *International Journal of Distributed Sensor Networks*, 13(11):1550147717739157, 2017.
123. Jianbing Ni, Xiaodong Lin, et al. Towards edge-assisted internet of things: From security and efficiency perspectives. *arXiv preprint arXiv:1902.07094*, 2019.
124. Bruno Astuto A Nunes, Marc Mendonca, Xuan-Nam Nguyen, Katia Obraczka, and Thierry Turletti. A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys & Tutorials*, 16(3):1617–1634, 2014.
125. Nicolae Paladi and Christian Gehrmann. Sdn access control for the masses. *Computers & Security*, 80:155–172, 2019.
126. Arvind R Bhagat Patil and Nileshsingh V Thakur. Mitigation against denial-of-service flooding and malformed packet attacks. In *Third International Congress on Information and Communication Technology*, pages 335–342. Springer, 2019.
127. Pradnya Patil, Larry Hash, Joshua White, Ali Tekeoglu, et al. *SECURITY CHALLENGES IN SDN IMPLEMENTATION*. PhD thesis, 2018.
128. Isravel Deva Priya and Salaja Silas. A survey on research challenges and applications in empowering the sdn-based internet of things. In *Advances in Big Data and Cloud Computing*, pages 457–467. Springer, 2019.
129. Lawrence Rabiner and B Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986.
130. Fernando MV Ramos, Diego Kreutz, and Paulo Verissimo. Software-defined networks: On the road to the softwarization of networking. *Cutter IT journal*, 2015.
131. Deepak Singh Rana, Shiv Ashish Dhondiyal, and Sushil Kumar Chamoli. Software defined networking (sdn) challenges, issues and solution. 2019.
132. Filippo Rebecchi, Julien Boite, Pierre-Alexis Nardin, Mathieu Bouet, and Vania Conan. Ddos protection with stateful software-defined networking. *International Journal of Network Management*, 29(1):e2042, 2019.
133. Tirumaleswar Reddy, Daniel Wing, and Prashanth Patil. Short term certificate management during distributed denial of service attacks, January 10 2019. US Patent App. 16/110,102.
134. Francisco Javier Ros and Pedro Miguel Ruiz. Five nines of southbound reliability in software-defined networks. In *Proceedings of the third workshop on Hot topics in software defined networking*, pages 31–36. ACM, 2014.
135. Kshira Sagar Sahoo, Sanjaya Kumar Panda, Sampa Sahoo, Bibhudatta Sahoo, and Ratnakar Dash. Toward secure software-defined networks against distributed denial of service attack. *The Journal of Supercomputing*, pages 1–46, 2019.
136. Stuart E Schechter, Jaeyeon Jung, and Arthur W Berger. Fast detection of scanning worm infections. In *International Workshop on Recent Advances in Intrusion Detection*, pages 59–81. Springer, 2004.
137. Nicolas Schnepf, Rémi Badonnel, Abdelkader Lahmadi, and Stephan Merz. Automated verification of security chains in software-defined networks with synaptic. In *Network Softwarization (NetSoft), 2017 IEEE Conference on*, pages 1–9. IEEE, 2017.
138. Sandra Scott-Hayward, Sriram Natarajan, and Sakir Sezer. A survey of security in software defined networks. *IEEE Communications Surveys & Tutorials*, 18(1):623–654, 2016.
139. Sandra Scott-Hayward, Gemma O'Callaghan, and Sakir Sezer. Sdn security: A survey. In *Future Networks and Services (SDN4FNS), 2013 IEEE SDN For*, pages 1–7. IEEE, 2013.
140. sFlow RT. "a real-time monitoring tool (sampled flow real-time)", January 2018. www.inmon.com/products/sFlow-RT.php.
141. Arash Shaghaghi, Mohamed Ali Kaafar, Rajkumar Buyya, and Sanjay Jha. Software-defined network (sdn) data plane security: Issues, solutions and future directions. *arXiv preprint arXiv:1804.00262*, 2018.
142. Pramod Shanbhag, Lakshmi Narayana DRONADULA, and R Abhilash. Reducing multicast service traffic for matching and streaming in sdn (software defined networking enabled networks, January 29 2019. US Patent App. 10/193,763.
143. Toru Shimanaka, Ryusuke Masuoka, and Brian Hay. Cyber deception architecture: Covert attack reconnaissance using a safe sdn approach. In *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019.
144. Seungwon Shin, Vinod Yegneswaran, Phillip Porras, and Guofei 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*, pages 413–424. ACM, 2013.
145. Palaiahnakote Shivakumara, Dongqi Tang, Maryam Asadzadehkaljahi, Tong Lu, Umapada Pal, and Mohammad Hossein Anisi. Cnn-rnn based method for license plate recognition. *Caai Transactions on Intelligence Technology*, 3(3):169–175, 2018.
146. Chris Sinclair, Lyn Pierce, and Sara Matzner. An application of machine learning to network intrusion detection. In *Computer Security Applications Conference, 1999.(ACSAC'99) Proceedings. 15th Annual*, pages 371–377. IEEE, 1999.
147. snort. "a signature based intrusion detection system (snort ids)", January 2018. www.en.wikipedia.org/wiki/Snort.
148. Chungsik Song, Younghee Park, Keyur Golani, Youngsoo Kim, Kalgi Bhatt, and Kunal Goswami. Machine-learning based threat-aware system in software defined networks. In *Computer Communication and Networks (ICCCN), 2017 26th International Conference on*, pages 1–9. IEEE, 2017.
149. Haoyu 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*, pages 127–132. ACM, 2013.
150. Nasrin Sultana, Naveen Chilamkurti, Wei Peng, and Rabei Alhadad. Survey on sdn based network intrusion detection system using machine learning approaches. *Peer-to-Peer Networking and Applications*, 12(2):493–501, 2019.

151. Said Talbi and Abdelmadjid Bouabdallah. Interest-based trust management scheme for social internet of things. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–12, 2019.
152. Huan-rong Tang, Chao Xu, Xin-gao Luo, and Jian-quan OuYang. Traceback-based bloomfilter ips in defending syn flooding attack. In *Wireless Communications, Networking and Mobile Computing, 2009. WiCom'09. 5th International Conference on*, pages 1–6. IEEE, 2009.
153. Yuchia Tseng, Montida Pattaranantakul, Ruan He, Zonghua Zhang, and Farid Naït-Abdesselam. Controller dac: Securing sdn controller with dynamic access control. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017.
154. Fahad Ubaid, Rashid Amin, Faisal Bin Ubaid, and Muhammad Muwar Iqbal. Mitigating address spoofing attacks in hybrid sdn. *INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS*, 8(4):562–570, 2017.
155. Tariq Umer, Mubashir Husain Rehmani, Ahmed E Kamal, and Lyudmila Mihaylova. Information and resource management systems for internet of things: Energy management, communication protocols and future applications, 2019.
156. Jonathan Vestin. *SDN-Enabled Resiliency in Computer Networks*. PhD thesis, Karlstads universitet, 2018.
157. Stefano Vissicchio, Laurent Vanbever, and Olivier Bonaventure. Opportunities and research challenges of hybrid software defined networks. *ACM SIGCOMM Computer Communication Review*, 44(2):70–75, 2014.
158. Guodong Wang, Yanxiao Zhao, Jun Huang, and Yulei Wu. An effective approach to controller placement in software defined wide area networks. *IEEE Transactions on Network and Service Management*, 15(1):344–355, 2018.
159. Haopei Wang, Lei Xu, and Guofei Gu. Floodguard: A dos attack prevention extension in software-defined networks. In *Dependable Systems and Networks (DSN), 2015 45th Annual IEEE/IFIP International Conference on*, pages 239–250. IEEE, 2015.
160. Rui Wang, Zhiping Jia, and Lei Ju. An entropy-based distributed ddos detection mechanism in software-defined networking. In *Trustcom/BigDataSE/ISPA, 2015 IEEE*, volume 1, pages 310–317. IEEE, 2015.
161. Tao Wang and Hongchang Chen. Sguard: A lightweight sdn safe-guard architecture for dos attacks. *China Communications*, 14(6):113–125, 2017.
162. Yang Wang, Tao Hu, Guangming Tang, Jichao Xie, and Jie Lu. Sgs: Safe-guard scheme for protecting control plane against ddos attacks in software-defined networking. *IEEE Access*, 2019.
163. Lloyd R Welch. Hidden markov models and the baum-welch algorithm. *IEEE Information Theory Society Newsletter*, 53(4):10–13, 2003.
164. Jia-Si Weng, Jian Weng, Yue Zhang, Weiqi Luo, and Weiming Lan. Benbi: Scalable and dynamic access control on the northbound interface of sdn-based vanet. *IEEE Transactions on Vehicular Technology*, 68(1):822–831, 2019.
165. Wenfeng Xia, Yonggang Wen, Chuan Heng Foh, Dusit Niyato, and Haiyong Xie. A survey on software-defined networking. *IEEE Communications Surveys & Tutorials*, 17(1):27–51, 2015.
166. Rui Xiao, Hui Zhu, Chao Song, Ximeng Liu, Jian Dong, and Hui Li. Attacking network isolation in software-defined networks: New attacks and countermeasures. In *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–9. IEEE, 2018.
167. Qiao Yan, F Richard Yu, Qingxiang Gong, and Jianqiang Li. Software-defined networking (sdn) and distributed denial of service (ddos) attacks in cloud computing environments: A survey, some research issues, and challenges. *IEEE Communications Surveys & Tutorials*, 18(1):602–622, 2016.
168. Seunghyun Yoon, Taejin Ha, Sunghwan Kim, and Hyuk Lim. Scalable traffic sampling using centrality measure on software-defined networks. *IEEE Communications Magazine*, 55(7):43–49, 2017.
169. Changhe Yu, Julong Lan, Zehua Guo, Yuxiang Hu, and Thar Baker. An adaptive and lightweight update mechanism for sdn. *IEEE Access*, 2019.
170. Adel Zaalouk, Rahamatullah Khondoker, Ronald Marx, and Kpatcha Bayarou. Orchsec: An orchestrator-based architecture for enhancing network-security using network monitoring and sdn control functions. In *Network Operations and Management Symposium (NOMS), 2014 IEEE*, pages 1–9. IEEE, 2014.
171. Heng Zhang, Zhiping Cai, Qiang Liu, Qingjun Xiao, Yangyang Li, and Chak Fone Cheang. A survey on security-aware measurement in sdn. *Security and Communication Networks*, 2018, 2018.
172. Shaobo Zhang, Xiong Li, Zhiyuan Tan, Tao Peng, and Guojun Wang. A caching and spatial k-anonymity driven privacy enhancement scheme in continuous location-based services. *Future Generation Computer Systems*, 94:40–50, 2019.
173. Shuce Zhang, Hiromu Iwashita, and Kazushi Sanada. Thermal performance difference of ideal gas model and van der waals gas model in gas-loaded accumulator. *International Journal of Hydromechatronics*, 1(3):293–307, 2018.
174. Jianjun Zheng and Akbar Siami Namin. A survey on the moving target defense strategies: An architectural perspective. *Journal of Computer Science and Technology*, 34(1):207–233, 2019.
175. Hong Zhong, Jianqiao Sheng, Yan Xu, and Jie Cui. Scplbs: a smart cooperative platform for load balancing and security on sdn distributed controllers. *Peer-to-Peer Networking and Applications*, 12(2):440–451, 2019.
176. Ying Zhou, Quansen Sun, and Jixin Liu. Robust optimisation algorithm for the measurement matrix in compressed sensing. *CAAI Transactions on Intelligence Technology*, 3(3):133–139, 2018.
177. Liehuang Zhu, Md Monjurul Karim, Kashif Sharif, Fan Li, Xiaojiang Du, and Mohsen Guizani. Sdn controllers: Benchmarking & performance evaluation. *arXiv preprint arXiv:1902.04491*, 2019.
178. Yuan Zuo, Yulei Wu, Geyong Min, and Laizhong Cui. Learning-based network path planning for traffic engineering. *Future Generation Computer Systems*, 92:59–67, 2019.