Journal of Network and Computer Applications 187 (2021) 103093

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

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

Review

Emerging DDoS attack detection and mitigation strategies in software-defined networks: Taxonomy, challenges and future directions

Ismael Amezcua Valdovinos [a], Jesús Arturo Pérez-Díaz [b], Kim-Kwang Raymond Choo [c,*], Juan Felipe Botero [d]

[a] Universidad de Colima, Av. Universidad 333, Colima, 28040, Colima, Mexico
[b] Tecnologico de Monterrey, Escuela de Ingeniería y Ciencias, Epigmenio González 500, Querétaro, 76130, Qro., Mexico
[c] Department of Information Systems and Cyber Security, University of Texas at San Antonio, 1 UTSA Cir, San Antonio, 78249, TX, USA
[d] Departamento de Ingeniería Electrónica y de Telecomunicaciones, Universidad de Antioquia, Calle 67 # 53-108, Medellín, 050010, Antioquia, Colombia

ARTICLE INFO

MSC:
0000
1111

Keywords:
Software-defined network (SDN)
Denial of Service (DoS)
Cyberattacks
Machine learning
Blockchain
Network Virtual Function (NFV)
Honeynet
Network Slicing (NS)
Moving Target Defense (MTD)

ABSTRACT

Software-defined networking (SDN) is a network paradigm that decouples control and data planes from network devices and places them into separate entities. In SDN, the controller is responsible for controlling the logic of the entire network while network switches become forwarding elements that follow rules to dispatch flows. There are, however, several limitations in such a paradigm, as compared to conventional networking. For example, the controller is sensitive to a broad range of attacks, including distributed denial of service (DDoS) attacks. In this paper, we provide a systematic survey of existing DDoS detection and mitigation strategies in SDN. Based on the review of articles published between 2013 and May 2020, we provide a taxonomy of DDoS detection strategies (e.g., statistical, SDN architecture, and machine learning) and emerging approaches (e.g., network function virtualization, blockchain, honeynet, network slicing, and moving target defense). We also discuss existing challenges associated with SDN security and the implementation of security solutions, prior to identifying future research opportunities.

1. Introduction

Software-defined networking (SDN) is a relatively new networking paradigm, which can potentially alleviate the limitations of current switching networking by decoupling control and data planes, formerly implemented inside switches and routers, and enabling more flexible and manageable environments (Kreutz et al., 2015). In SDN, the control plane is located in a logically centralized controller, which simplifies policy enforcement and network configuration evolution (Kim and Feamster, 2013). Advantages associated with SDN include (i) network policies are defined by high-level languages in applications instead of low-level, vendor specific commands; (ii) application development is straightforward since the controller provides useful network abstractions such as global network view, flow management, device management, and statistics tracking; (iii) switching devices become multi-purpose devices by following flow rules provided by the control plane. These features enable the development of complex network applications that run on top of the controller to provide better management and decisions inside the network.

SDN architecture, however, presents several points of failure when compared to conventional networking. The controller being the most important entity in SDN, is also very sensitive to a broad range of attacks. For example, Swami et al. (2019a) explained that the following four features make SDN vulnerable to attacks:

1. Limited *ternary content addressable memory* (TCAM). Switches use TCAM to store flow rules. However, switches have limited space for storing large volume of entries.
2. Single point of failure. A centralized entity can potentially downgrade network performance, availability, and integrity of the network. Deploying multiple controllers can mitigate such a risk. However, secure communication between such entities is desirable, if not essential
3. Decoupling of control and data plane. Control and data planes communicate through an OpenFlow protocol. Attackers can implement saturation attacks, denial of service (DoS) attacks,

* Corresponding author.
E-mail addresses: ismaelamezcua@ucol.mx (I.A. Valdovinos), jesus.arturo.perez@tec.mx (J.A. Pérez-Díaz), raymond.choo@fulbrightmail.org (K.-K.R. Choo), juanf.botero@udea.edu.co (J.F. Botero).

https://doi.org/10.1016/j.jnca.2021.103093
Received 15 June 2020; Received in revised form 9 April 2021; Accepted 30 April 2021
Available online 11 May 2021

man-in-the-middle attacks, among others, to exhaust switch-controller bandwidth.
4. Plain switches. Switches rely on the controller to determine the actions for unknown flows. This degrades network performance if the switch constantly communicates to its controller.

Given the trend of SDN to be the next generation networking paradigm, understanding the security limitations and risks is crucial. Therefore, we investigate the different cyberattacks targeting SDN and their mitigation techniques. Specifically, we provide an in-depth review of existing research efforts on the detection and mitigation of DDoS attacks in SDN. We searched on Google Scholar using keywords such as ("software-defined network" AND cyber-attacks), ("software-defined network" AND mitigation), and ("software-defined network" AND cyber-security') to locate articles published between 2013 and 05/2020. A total of 22,800 articles were located, and after further processing, we included 38 articles in this review. For example, articles that have been previously included and discussed in existing literature reviews and surveys. As noted in Section 5, we also locate a number of existing literature reviews and surveys, and we will explain how this paper differs from existing efforts. Based on our review, we present a new taxonomy for DDoS detection and mitigation strategies in SDN. We also identify and discuss existing and emerging research challenges and opportunities.

From our literature review, we identify known strategies for DDoS detection and mitigation such as the use of statistical mechanisms to determine whether the system is being attacked by comparing the entropy of the system, the use of machine learning (ML) algorithms to provide better detection accuracy for intrusion detection systems (IDS), and the proposal of flexible architectures for ML-based IDS and intrusion prevention systems (IPS). Also, we identify new and emerging technologies for DDoS detection and mitigation, albeit these technologies are not designed for such purpose. For example, using Network Function Virtualization (NFV) to quickly deploy custom firewalls, Honeynets to gather information about attacks and distribute them for better collaboration-based IDS, and mechanisms such as Moving Target Defense (MTD) that use SDN to constantly move/update configurations, topologies, and change the dynamics of the network for attack detection, mitigation, and information gathering. Furthermore, as SDN is being used as an enabling technology for most of the previously mentioned strategies, for the upcoming 5G network environments, network slicing (NS) is being used to dynamically deploy slices segmenting the network to minimize and mitigate the impact of DDoS attacks.

The remaining of this paper is structured as follows. Section 2 provides an overview of SDN, focusing on its architectural design, its benefits, and security risks. Section 3 describes the existing attack mitigation classifications, and our proposed taxonomy which will guide the discussion of the literature review in Section 4. In Section 5, we also revisit existing literature reviews and surveys, and explain how this paper complements existing works and contributes to the literature gap. In Section 6, we discuss the security challenges associated with application, control, and data planes of SDN, as well as those relating to approaches based on blockchain, NFV, and honeypot, NS, and MTD. Finally, we draw our conclusions in Section 7.

2. SDN architecture

As our information and communication technologies advance, our society becomes more connected. There are, however, limitations in our conventional networks, for example in terms of scalability, security, operational costs, maintainability, and customization. Current network services also rely mostly on proprietary technologies and heterogeneous, diverse network devices built for specific purposes (Sherry et al., 2012; Wang et al., 2011; Walfish et al., 2004). This approach limits operation of service additions and transparent software network upgrades.

Thus, there have been interest in the research and practitioner communities to design a new (SDN) architecture that can overcome such limitations whilst providing new networking developments. SDN can be broadly considered to be the evolution of earlier concepts, such as programmable networks and interfaces, and separation of control and management planes — see also Fig. 1.

One fundamental principle in the SDN architecture, is the decoupling of the physical and logical planes in networking devices. Plane separation allows each layer to evolve almost independently, thus, enabling innovation, fast implementation and adoption of new features and services, better manageability and testing capabilities, among others. A second principle is transparency, in which devices and users should not be able to differentiate between a conventional network and an SDN. A third principle is automation and runtime deployment by logically centralizing a control plane and introducing programmable entities. This last feature enables the development of complex networking applications for better network performance, manageability, and control.

In order to fulfill the principles discussed earlier, the Open Networking Foundation (ONF) proposed a three-layer architecture for SDN. The architecture of SDN is depicted in Fig. 2.

2.1. Application plane

The application plane includes a variety of services and applications such as Deep Packet Inspector (DPI), Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS), monitoring, and load balance, which can inform decision-making in traffic engineering, quality of service (QoS) differentiation, monitoring, routing, and many other services. Applications communicate to the controller through the Northbound API, which can be implemented as a RESTful API, or through programming languages bindings such as Python and Java APIs.

2.2. Control plane

The control plane is responsible for the management of the underlying forwarding devices by using global network knowledge and information for decision making. It also interacts with the application plane to provide useful information for applications through the Northbound interface. The controller translates the requirements of networks applications running on top of it to low-level flow rules, which are shared with SDN devices through the Southbound interface for their installation (Bannour et al., 2018).

The controller provides core functionalities such as topology management, which maintains information about the interconnection of devices between each other and to end users; flow management, which is a database of the flows currently being used in the network to ensure correct synchronization between SDN devices; statistics collected from SDN devices, and device management for the discovery of end-user and network devices that comprise the infrastructure of the network.

A high-end centralized controller can handle large amounts of network flows. However, when the number of SDN devices and the traffic flows grow over time the controller can become a network bottleneck. In widely-spread networks, a single controller also introduces delay (Alsaeedi et al., 2019). Therefore, the use of multiple distributed controllers is a proposal to alleviate the problems previously described. Thus, the control plane is logically centralized but can be implemented as physically distributed systems, aiming at scalability and reliability of the whole networking system. To support such distributed architecture, the controllers use the East-Westbound interface to communicate to each other (Lin et al., 2015). As far as we know, there is still no standardization attempts for controller to controller communication and remains as a proprietary solution.

An important aspect of distributed controllers in SDN is the controller placement problem. Heller et al. (2012) address this topic by introducing two fundamental questions: how many controllers are needed

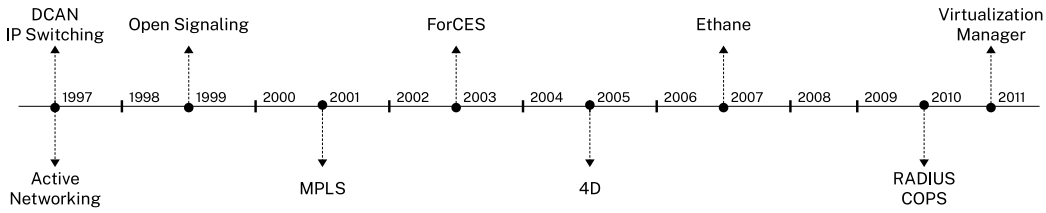

Fig. 1. Enabling technologies for the development of SDN.

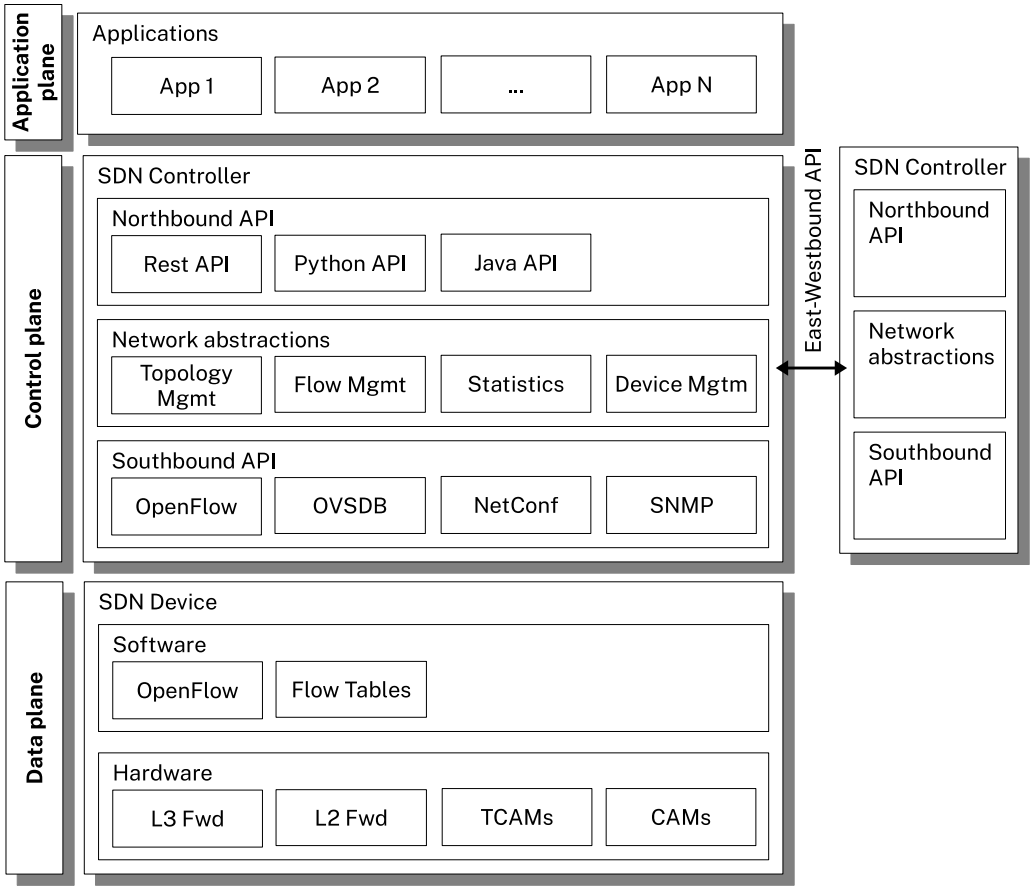

Fig. 2. SDN architectural view.

and where should they be placed in the topology. Authors argue that in Wide Area Networks (WANs), controller placement is key for the improvement of network latency. In medium-size networks, a single controller can meet response-time goals. More recent studies also suggest that single controller deployments in data centers and service network providers can be quickly overloaded whilst showing scalability implications (Yeganeh et al., 2013; Azodolmolky et al., 2013). Nowadays, the controller placement problem has received a lot of attention and has been further studied in recent literature (Das et al., 2020; Lu et al., 2019; Wang et al., 2017; Alsaeedi et al., 2019), as it is a challenge for modern data centers.

2.3. Data plane

The data plane is constituted by a set of network elements, usually called SDN devices, which are responsible for packet forwarding based on flow rules. The control-to-data plane communication is performed via an open vendor-agnostic Southbound interface. OpenFlow (McKeown et al., 2008), ForCES (Forwarding and Control Element Separation) (Doria et al., 2010), and more recently P4 (Foundation, 2020) are well-known candidates for such interface.

As depicted in Fig. 2, an SDN device can be further described by its software and hardware entities. The device requires the implementation of a well-known Southbound interface such as OpenFlow and an abstraction for the representation of the flow rules entries in Flow Tables. Its hardware counterpart, has layer 2 (L2 Fwd) and layer 3 (L3 Fwd) forwarding mechanisms implemented through field-programmable gate arrays (FPGAs) for faster forwarding. It also implements Content-Addressable Memory (CAM) and Ternary CAMs (one or both depending on the manufacturer) for storing flow rules. Flow rules, according to OpenFlow, maintain a list of flow entries and each flow is composed by a *match field* containing the header values to match, *counters* to update when packets match for certain rule, and a set of *actions* to apply when matching. Lastly, the data plane is also responsible for collecting network information and statistics (through the counter entry) to be later shared with the controller.

2.4. Benefits of SDN

One major drive for SDN is simplification. As time passes, networking devices are becoming more complex due to the increasing number

of different network technologies to be supported. Adding hardware-based features tend to complicate implementation of such technologies. There is also an opportunity to enhance device management: rather than using tools such as SNMP and CLI, SDN enables the use of policy-based management systems (Kim and Feamster, 2013).

Increased control plane sophistication relates to an increase in the cost of networking hardware components due to the processing power required to run such advanced solutions. By using readily available and tested open source software, the development of new networking technologies may result in the reduction of costs for hardware devices. This also encourages the adoption of open environments, where protocol enhancements may be implemented by any developer or community instead of adjusting to what network vendors may offer to consumers. Thus, this allows faster growth rates in terms of new technologies and reduced costs since existing hardware can be updated to operate with new and updated protocols.

Some other SDN features relate directly to the data center architecture: automation, where SDN allows networks to be dynamically instantiated and disabled when required; scalability, where large collections of MAC address tables and VLANs become easier to manage and maintain when global network knowledge is provided; multi-path, where SDN allows to setup multiple redundant paths to provide network resiliency and fast recovery; multi-tenancy, where SDN allows for an easy management of user requirements to support different characteristics to each user; and network virtualization support, among others (Göransson et al., 2017).

2.5. Security challenges in SDN

SDN's current architecture allows the development of robust security features due to its flexible and programmable nature. Indeed, Dacier et al. (2017) provide insights on the security challenges and concerns among researchers from industry and academia exposed at the Dagstuhl research seminar, a worldwide-renowned seminar hosted on September 2016 in Germany, where pros and cons about SDN were discussed. SDN's capability of detecting attacks might become easier and more reliable, at the cost of increasing the attack surface since standards are vague about security mechanisms, such as authorization. Specifically, the introduction of networking applications that interact with the controller to define network behavior based on application demands adds complexity in terms of authentication and authorization schemes.

Abdou et al. (2018) analyze security threats related to the control plane in SDN. Authors show several spoofing and flooding attacks on different features of SDN. For example, to provide a global view of the network, the control plane has to gather information about the hosts and SDN devices connected to the network. This process is subject to spoofing attacks (MAC spoofing, IP spoofing, VLAN tag spoofing) since attackers can send fake Link-Layer Discovery Protocol (LLDP) packets, providing the controller of inaccurate data to form the network topology. Network applications that rely on this kind of data for decision making may lead to wrong computations also. Authors argue that this can also be true for loop-free forwarding, link redundancy, and device redundancy mechanisms used in SDN. If using distributed controllers for scalability, communication vulnerabilities arise, leading to synchronization failures and misconfiguration.

More recently, authors in Swami et al. (2019b) described SDN vulnerabilities from a different point of view. As discussed earlier, SDN has features that provide resiliency against malicious attacks such as having a centralized monitoring entity for malicious traffic and programmable configurations, both unique to SDN. However, its architecture also enables other kind of attacks:

- SDN devices have limited memory for flow rule storage (limited TCAM). In SDN, flow rules change frequently as a response of different flows. For example, an attacker can generate large volumes of packets using different protocols, rapidly populating the flow table and overflowing the memory capacity of the device.
- The centralized nature of the controller can become a single point of failure. DDoS attacks can target a controller, downgrading its performance, availability, and integrity of the network. The deployment of multiple distributed controllers may help in alleviating such scenarios.
- When using distributed controllers, security issues arise in terms of authentication, consistency, and scalability of the different policy rules in each domain. Secure channels are needed to provide communication between controllers. Moreover, there is no standard specification on the communication protocol and how it should be secured in order to exchange state information between controllers. If one controller is compromised, more controllers may start failing due to the issues discussed earlier. This is known as a cascading failure problem.
- In order to create, modify, and delete flow rules inside SDN devices, it is imperative to establish a communication channel between the controller and the device. Attackers can implement saturation attacks, DDoS attacks, man-in-the-middle attacks, among others, to exhaust such channel.

Table 1
SDN security risks.

Risk	Description
Unauthorized access	Concerned with access control methods. Examples include unauthorized controller access or unauthenticated application access to services and network information such as network states.
Data leakage	Concerned with the discovery of internal information required by the network in order to operate. Examples of data leakage include the discovery of flow rules on switches, forwarding policy discovery, lack of secure storage of credentials (e.g., keys and certificates).
Data modification	Concerned with the ability of an attacker to insert or modify flow rules in network devices.
Malicious Applications	Concerned with network privileges given to malicious applications or the use of poorly implemented applications.
Configuration issues	Concerned with wrong/faulty configurations or incorrect use of security features in the network.
Denial of service	Concerned with the security weakness of a central controller. An attacker could flood the controller with packets requiring new flow rules, achieving switch buffer depletion, bandwidth depletion on the controller-switch interface, or controller buffer depletion when slow attacks are used on the controller.

SDN aims to alleviate the shortcomings faced by conventional networking architectures in large-scale, dynamic networking requirements. However, we believe that the very same features that can be used to provide quick and efficient detection and mitigation techniques, can also be a target for attackers to drive the network to inadequate levels of performance. Table 1 summarizes a general overview of security risks present in current SDN architectures.

2.6. Denial of service in SDN

A Denial of Service (DoS) or a Distributed DoS (DDoS) aims at disrupting operations and at driving the network unresponsive by exhausting both network and computing resources such as bandwidth, processing power, and memory. In bandwidth exhaustion, the attacker floods the network with traffic at very high rates enough to deplete the channel's available bandwidth. After the communication channel has been exhausted, legitimate traffic cannot be served, and the service becomes unresponsive. Moreover, the attack could make use of protocol level vulnerabilities such as TCP's timeout retransmission (Kuzmanovic and Knightly, 2003; Shevtekar et al., 2005), congestion control mechanisms (Luo and Chang, 2005), and HTTP's keep alive mechanism (Adi et al., 2015), to name a few, can lead to the depletion of memory and

processing power on the server. It is important to mention that in the report of Kaspersky Q2 2020 DDoS attacks report (Kaspersky, 2020), the number of the DDoS attacks in the second quarter of 2020 increased three-fold in comparison to the same period in 2019. It implies a dramatic increment of the DDoS attacks during 2020.

In the context of SDN, apart from conventional DDoS risks, this kind of attacks can be targeted at specific points of SDN's architecture in order to bring down or degrade the performance of the network (Swami et al., 2019b). An attacker may create large amounts of malicious traffic with unknown/random protocol headers to trigger table-miss processes on the switch: if the switch does not find a match for such packets, it encapsulates and forwards them to the controller for further inspection and decision making. This process leads to several risks on the switches, the controller, and on the bandwidth of the switch-controller communication channel: (i) the switch may overflow since it only possess a limited TCAM storage for flow rules; (ii) the controller may exhaust its processing power when the number of table-miss issues is large enough; and (iii) the bandwidth of the link that interconnects the switch with the controller may exhaust its capacity caused by the amount of packets exchanged between both devices.

3. Detection and mitigation of cyberattacks

One of the contributions of our work is the classification of emerging strategies or techniques for the detection and mitigation of DDoS in SDN. Based on our literature review, we propose a taxonomy based on strategies that SDN brings for securing the network against DDoS, strategies for SDN that can be implemented as applications running on top of the controller, and strategies that use SDN as an enabler but are not particularly designed for SDN. In such context, our taxonomy spans across well-defined detection and mitigation techniques such as Statistical- and Machine Learning (ML)-based strategies, as well as new and emerging technologies such as Blockchain, Network Function Virtualization (NFV), Honeynet, Network Slicing (NS), and Moving Target Defense (MTD)-based strategies. Fig. 3 depicts how our taxonomy is organized. Some of these strategies are aimed at wider ecosystems such as 5G communications (when exploring NS strategies). SDN is considered a key technology and enabler of 5G systems because of its robust architecture and its ability to dynamically update and install protocols and services. Moreover, SDN can be further extended with new technologies that improve network operations, automation, and scalability, without disrupting the operational workflow.

This section provides a description of each category in our taxonomy. It is worth noting that strategies such as Statistical, ML, and architectural SDN improvements are well-studied techniques for DDoS detection and mitigation. Therefore, this survey is not meant to provide a comprehensive study of such strategies. We believe there is a significant number of surveys that provide complete views on those approaches. However, we do describe and analyze current and representative works on each well-studied category for the sake of completeness. Instead, this survey is aimed at providing the reader with emerging detection and mitigation strategies for DDoS cyberattacks in scenarios where SDN is used to enforce security issues, also where SDN is further extended with strategies for detection and mitigation of DDoS, and finally where SDN is used as an enabler for emerging strategies.

3.1. Strategies enhancing SDN security capabilities

As mentioned earlier, one of the strengths of SDN is the separation of responsibilities by using a layered architecture. However, all layers are vulnerable to different kinds of security threats, being the controller and control plane the most sensitive to such threats. Common DDoS detection and mitigation strategies in this category include channel flooding between the controller and the switches using specially crafted packets introduced by malicious hosts, as well as constantly sending unknown packets to the switches and consequently overflowing their flow rule buffers.

We can further categorize architectural enhancements for SDN with regards of cyberattack detection and mitigation by planes. In the application plane, where network applications such as routing, traffic monitoring, and virtualization are deployed, mitigation techniques attempt to avoid unauthorized access and provide better security mechanisms to avoid the injection of malicious entities to the controller through the northbound API. In the control plane, besides securing the controller, security enhancements aim at providing a more robust communication channel between network applications and the controller (northbound API), and the communication channel between controllers in multi-controller architectures (east–westbound API). In the data or forwarding plane, mitigation techniques aim at avoiding switch buffer overflows, placement of fake or malicious rules on the forwarding table, and to secure the communication channel between the switches and controller (southbound API).

3.2. Strategies that provide SDN security capabilities

We define the strategies that provide SDN security capabilities as those approaches that add detection and mitigation capabilities through the use and implementation of modules and applications on top of SDN. Such approaches comprise SDN applications, standalone applications that communicate to the SDN controller, and extra modules proposed to act as security providers. We further categorize the strategies in Statistical, ML, and Blockchain strategies. DDoS mitigation strategies in this category span from the detection and mitigation of low and high rate DoS attacks using statistical and ML algorithms to securing and sharing information about malicious nodes in inter and intra-domain collaboration frameworks using blockchain as a secure and verifiable alternative.

3.2.1. Description of statistical strategies

In Statistical-based attack mitigation techniques, a statistical model for common traffic is computed and later used in deduction tests to determine if traffic flows follow the computed model. Flows not obeying such statistical model are classified as malicious and are later used to gain knowledge about the attack and to determine the best mitigation mechanism.

According to Khalaf et al. (2019), statistical-based DDoS detection and mitigation strategies for SDN include the analysis of correlation between multiple features in order to classify malicious and legitimate traffic. Statistical moments such as the mean and standard deviation are also widely used for observing patterns in flows to match anomalous flows. Such techniques are the basis of anomaly-based intrusion detection systems. Regression analysis is also another popular technique which allows to predict through a mathematical model how flows should behave. From our literature review, we observe that statistical approaches are still a potential solution to the detection and mitigation of malicious flows in SDN.

3.2.2. Description of machine learning strategies

Machine Learning (ML) in the context of SDN is primarily used to develop systems that can learn from the data flows, extract hidden patterns, and perform decisions automatically (Sultana et al., 2019). Such techniques are proven to be efficient for DDoS detection and mitigation that improve detection rate, reduce false alarm rates, and decrease computation and communication cost.

ML techniques can be further be categorized in supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning, and deep learning (Sultana et al., 2019; Xie et al., 2019; Zhao et al., 2019; Mohammed et al., 2019). In supervised learning, a labeled training dataset is used to build and train the system model, which is then used to perform decisions according to a set of rules. Unsupervised learning on the other hand, uses unlabeled datasets to

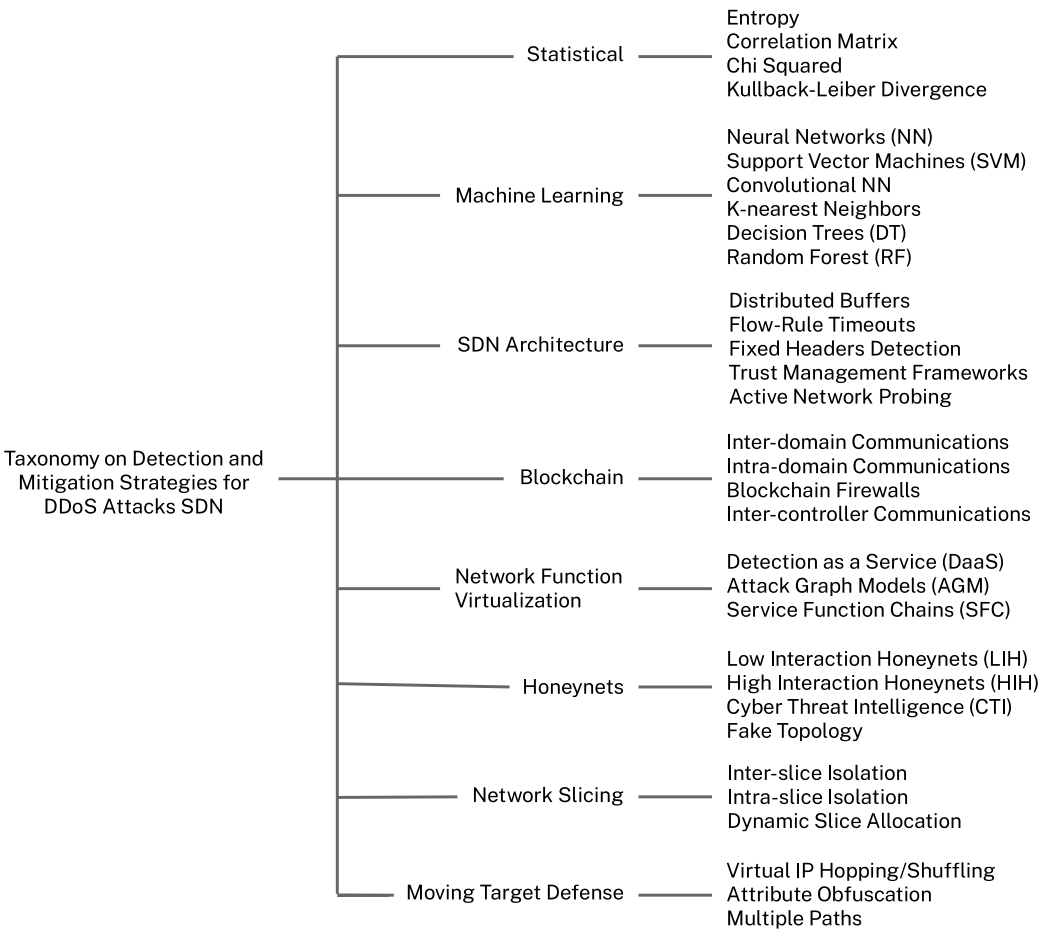

Fig. 3. Taxonomy on detection and mitigation strategies for DDoS Attacks in SDN.

find patterns, structures, or knowledge by clustering similar data into a predefined number of groups. Semi-supervised learning trains its model with labeled and unlabeled data in order to discover and find hidden patterns. In Reinforcement learning an agent learns in a trial-and-error manner by using an action space and a state space. The goal is to dynamically adjust parameters to achieve the maximum reinforcement signal, which is a long-term reward based on immediate and future rewards. In the context of SDN, the controller acts as an agent which monitors the network status to make decisions usually to the data forwarding plane. Deep learning uses multiple layered neural networks interconnected as neuron-like nodes. Pattern recognition is performed at a hidden layer via activation functions contained in each node. Each layer takes the input of the previous layer as input to apply non-linear transformations for feature extraction and classification.

3.2.3. *Description of blockchain strategies*

Blockchain is a decentralized platform that enables secure, shared, and distributed recording and tracking of resources. It maintains records in distributed, fault-tolerant, and append-only blocks, similar to database entries. Blocks are accessible to all participants in the blockchain but cannot be deleted nor altered. Each block points to the previous block through a reference hash value of its parent block. A block is composed by a header, which holds the block version, the parent block hash, the Merkle tree root hash (the hash value for all transactions in the block), a timestamp, a current hash target, and a nonce that increases for every hash calculation; the body is composed of a transaction counter and transactions (Zheng et al., 2018). Fig. 4 depicts how a blockchain is composed.

The blockchain technology enables the communicating parties to interact without the need of a trusted third-party entity (Salman et al., 2019). The characteristics of blockchain, including decentralization, persistence, and auditability, propose a promising technology for its implementation in a plethora of sectors such as healthcare, finance, smart contracts, IoT, and SDN (Kosba et al., 2016; Fernández-Caramés and Fraga-Lamas, 2018).

There are three types of blockchains in terms of data management and availability: public, private and consortium. In public blockchains, any node can participate and interact with any other node. In private blockchains, network access is restricted. Hence, any node wanting to participate in the blockchain must obtain approval from the blockchain owner. When applied to SDN, the controller becomes the owner of the blockchain, only giving permissions to trusted devices on its network. In consortium blockchains, selected nodes can validate blocks, but any node can perform transactions. From our literature review, private and consortium blockchains are used in the context of SDN since the controller plays the role of the blockchain owner. Blockchain in SDN allows transactions between collaborative SDN controllers in order to disseminate and execute proper actions when an attack is detected. Each transaction performed is confirmed by members in order to be appended to the blockchain. Only authorized nodes inside the SDN are able to confirm and append transactions. This allows SDN on multi-domain scenarios to securely collaborate to prevent and mitigate cyberattacks. By assigning to the controller the role of central authority and owner of a trusted blockchain it is expected to intrinsically achieve reliability, safety, traceability and controller synchronization (Alharbi, 2020).

Security risks such as unauthorized access, data leakage, DoS, and data modification can be prevented by using blockchain technology. Therefore, we believe it is important to provide an overview of existing SDN/Blockchain-based cyberattack mitigation techniques.

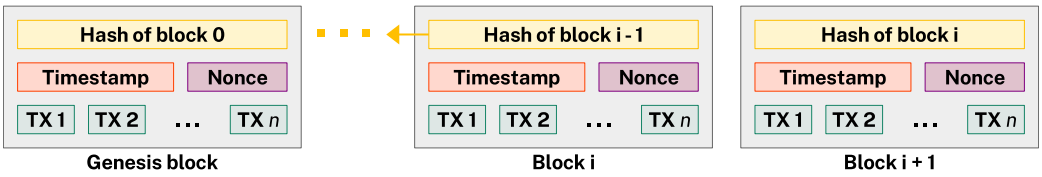

Fig. 4. An example of a blockchain. TX are the transactions.

3.3. Strategies that use SDN as an enabler

We define the strategies that use SDN as an enabler as those strategies that are not specifically designed for SDN but use SDN as an enabling technology for their deployment. For example, 5G technology uses SDN as a key technology for its development. The implementation of 5G on conventional networking hardware becomes a difficult task due to its dynamic configurations and service deployments. Therefore, SDN is being used as a network paradigm that allows the deployment of dynamically-configured application and virtualization services. Such services can be used to efficiently deploy firewalls, IDSs, and IPSs close to the location of DDoS attacks to facilitate detection and mitigation. Also, strategies such as the deployment of honeypots to divert malicious traffic and to learn from malicious nodes are approaches that have been proposed by researchers.

3.3.1. Description of network function virtualization strategies

NFV allows the use of virtualization to provide network functions such as software-based firewalls and gateways, where each VM performs different network operations (Bonfim et al., 2019). It aims to reduce equipment costs, to improve operation performance and efficiency, to optimize network configuration and resource allocation, to provide flexible network function deployment and dynamic operation, and to reduce energy consumption. NFV and SDN are complementary technologies aimed at providing complete and efficient networking solutions. On one hand, NFV provides different networking functions through VM instances. On the other hand, SDN provides the connectivity and programmability.

Benefits of translating network functions into VMs include the flexibility to allocate different network functions in vendor-agnostic hardware, quick implementation and deployment of on-demand networking functions, multi-tenancy support, and better management and automation of operational processes (ETSI, 2012).

In a cyberattack scenario, a SDN controller detecting a DDoS attack can launch a command to wake up or set up a VM with Network Functions (NFs) with specific configurations to mitigate the attack. The parameters on CPU and memory of the VM can be set up accordingly to the degree of security risk the attack is requiring. Furthermore, virtual Intrusion Detection Systems/Intrusion Prevention Systems and firewalls can be deployed dynamically to provide better security.

3.3.2. Description of honeynet strategies

Honeynets are deliberately vulnerable network segments carefully tuned to act as a target for attackers. Honeynets are composed by sets of honeypots, often isolated from real operating systems, services, and network functions. Data collected from honeypots provide warnings of new attacks and exploitation, allowing operators to establish mitigation plans. Legacy honeynet architectures suffer from proper data control mechanisms and data capture capabilities (Kyung et al., 2017). SDN can provide such mechanisms for honeynets to be properly implemented in next-generation networks. In this hybrid architecture, honeypots can act as vulnerable hosts of an SDN to provide early detection of cyberattacks and overall statistics about the attacks, whilst maintaining the network operational.

Honeynet-based techniques and mechanisms can be broadly classified as per interaction activity. In Low-Interaction Honepots (LIHs), the honeypot simulates limited network features in order to reduce maintaining costs. On the other hand, High-Interaction Honeypots (HIHs) can provide whole OS functions by hosting a variety of honeypots and services. HIHs can capture not only network behavior, but also system activity in order to provide better and more complete information about the attack. In Hybrid Honeypots (HHs), several frontend LIHs are deployed to simulate hosts and several backend HIHs gather information and interact with the attacker.

3.3.3. Description of network slicing strategies

5G networks aim at the provision of several user QoS requirements for different applications in terms of data transmission rates and latencies. The design principles for its architectural model include multi-tenancy to allow multiple service providers to operate through a shared infrastructure which consists of heterogeneous hardware and software resources. In order to manage such heterogeneous environments, 5G must implement efficient control frameworks and the fragmentation of administrative domains (Sayadi et al., 2016).

SDN is one of the key technologies that enables 5G deployment since it provides robust software control, hardware infrastructure, and the communication between them in a seamless manner (Zhang et al., 2017), whilst providing high reliability and high-speed for network data and nodes (Khan et al., 2020a).

Network Slicing (NS) refers to a form of network virtualization used in 5G networks that aim at deploying multiple logical networks running on top of a single shared physical network infrastructure. Rost et al. (2017) define NS as a logical end-to-end construct that is self-contained, has customized functions, and uses network function chains to provide desired QoS services to groups of devices.

Although NS are self-contained, they tend to exchange state information with each other for slice management and service optimization by using an entity called NS manager. Threats regarding NS include weak or non existent secure communication between network slices and their respective manager, impersonation attacks against a physical host and the NS manager and its instances, DDoS attacks against slices can have disruptive effects on other slices, and the use of different policies and protocols on each slice can lead to different levels of authentication and service level agreements for users.

There are recommendations to alleviate the security issues described earlier. For example, it is imperative to provide secure communications between NS and their manager in order to avoid Man-in-the-Middle attacks, mutual authentication to avoid impersonation attacks, and a centralized policy management to accurately manage different policies and authentication methods on each NS. However, NS can also be used to mitigate attacks due to their dynamic instances and configuration.

3.3.4. Description of moving target defense strategies

Moving Target Defense (MTD) emerged as an alternative to traditional security approaches such as Intrusion Detection Systems (IDS) to provide proactive defense against adaptive attacks (Sengupta et al., 2020). Nowadays, cyberattacks are based on complex, well-planned, adaptive attacks in which attackers spend time doing reconnaissance research on the target system in order to launch powerful attacks. MTD's premise is, if network configuration, software, and topologies are constantly moving and/or changing, the information gathered from the reconnaissance phase becomes obsolete and cannot be used for future attacks. Thus, the attacker must spend more time on gathering

resources to plan the attack. The goal is to constantly move configurations such as network ports, software, VMs, among others, in order to confuse and increase the attacker's uncertainty.

MTDs can be modeled as a three variable system $\langle M, T, C \rangle$ where M represents the movement strategy (what to move), T defines a timing function that represents when the movement will take place (when to move), and C represents the set of configurations present when the movement takes place (where to move) (Sengupta et al., 2019). For MTDs to be effective, the when to move variable must include randomness into its inputs. If the system configuration is moving at a constant rate, the attacker can easily overcome the system's defense mechanism.

Due to the nature of MTDs, it is difficult to provide a framework for traditional networks for its implementation. SDN and NFV are able to provide such framework since the prior allows administrators to configure and implement optimal movement strategies and NFV can help in bringing instances up of network functions dynamically.

4. Emerging mitigation strategies

This section provides an in-depth description of related works categorized by how they interact with SDN and by their detection and mitigation strategy. We defined the interaction with SDN as follows: strategies for detection and mitigation of cyberattacks by enhancing SDN, strategies that further extend the behavior of SDN by implementing modules and applications on the controller in order to provide security capabilities to SDN, and strategies that use SDN as an enabler technology that can be used to prevent or detect DDoS attacks. It also provides a summary with critical evaluations and comparison tables on each category.

4.1. Strategies that enhance SDN security capabilities

This section discusses proposals to provide defense mechanisms for DDoS by enhancing the inner workflow of SDN or by fine tuning parameters in order to avoid resource exhaustion at different layers.

Control plane attacks in SDN seek to exhaust its bandwidth by flooding the network with carefully crafted packets which are sent to the controller by the switch. Kandoi and Antikainen (2015) state that when the input buffer of a switch is full and it keeps receiving large volumes of new packets within a short time period, it leads to heavy control plane bandwidth consumption since the switch needs to forward complete packets to the controller. Thus, an increase in delay is experienced when installing new flow table entries. Due to the limited output queue size and high latency, the switch might not be able to forward packets in this time period. When memory on the switch is occupied and the switch receives an instruction to install a new flow rule, it then sends an `OFPT_ERROR` message to the controller with `OFPFMFC_TABLE_FULL` as error code and drops the packet. According to Kandoi and Antikainen (2015), having large timeout values prevents the creation of new flow rules. On the other hand, having low timeout values incurs in large overheads as the switch will repeatedly request rules for previously known flows. Hence, a low timeout value could possibly aid an attack on the switch's flow table. Based on this result, authors state that low timeout values help during DoS attacks on SDN. An experiment was conducted using Mininet with CpQD's OpenFlow 1.3 Software Switch and a NOX controller. Each packet is configured with a unique source port and traffic is sent from source (*h1*) to sink (*h2*). In each experiment, 50,000 UDP packets are sent from source *h1* at 1000 packets per second. The authors found that when an increase in idle timeout is met, an increase in the number of packet drops is also experienced.

Similarly, the result in the second experiment shows that an increase in the control plane bandwidth decreases the number of dropped packets as the switch cannot store the packets in its output queue. Network configuration needs to be fine-tuned. A mitigation strategy located at the control plane can be achieved by rate-limiting the number of packets sent to the controller as this allows the controller to handle large amounts of messages by instructing the switches to send their messages at lower rates.

Lin et al. (2017) take a different approach in preventing buffer overflow on switches. The defense mechanism, called PBUF, detects when a buffer in a switch is about to overflow and finds idle buffers on switches to store flow rules for new packets. The Ternary Content Addressable Memory (TCAM) is a data structure used for allocating flow rules. However, their buffer capacity is limited. For example, the 540ZL switch has a TCAM that stores up to 1500 flow rule entries. To avoid buffer overflow when a DoS attack is occurring, PBUF gathers statistics of incoming packets and estimates the switch buffer size by means of network calculus, a mathematical study of queues that allows to find the lower bounds of the system. The buffer size is then used to define a system threshold which triggers the finding of idle buffers in other switches to store flow rules when occupation of the switch is approaching the specific threshold. Results show that PBUF improves the capacity for defending against DoS attacks in SDNs. However, there is no mechanism to differentiate between legitimate and malicious network traffic.

Durner et al. (2017) propose a detection and mitigation against DoS attacks in data plane based on their observation that header fields in attacking flows remain the same during the attack. Identifying fixed headers can help the detection of such attack. A DoS attack against the Data plane can be detected by restricting the fixed header fields of the attacking flow as these cause abnormal traffic patterns. The authors use a table of counters with different header fields as columns. These tables are inspected regularly for statistical data. The header fields are hashed with a uniformly dispersed function with fixed output size to normalize the table size. Two detection algorithms are provided: the first algorithm provides the method to unwrap and handle the incoming encapsulated `PacketIn` packets directed to the controller used to provide the switch with new rules; the second algorithm provides methods to execute detection routines at regular intervals and evaluates the counter value. Once an attack is detected, it can be stopped by installing a low-priority drop rule in the switch as it will handle further attacks without the aid of the controller. The algorithm is validated by abstracted simulation results. The study is limited for attackers with small botnets since large botnets can change all header fields simultaneously.

Trust management network frameworks provide each user trust level values to allow a certain type of traffic or rate from a host. During DDoS attacks, trust value of legitimate users may rapidly decline to the point that they get classified as attackers, resulting in legitimate users getting blocked and blacklisted even after the attack ends. Wei and Fung (2015) and Sarwar et al. (2019) investigate the introduction of trust management frameworks for SDN and SDN-based IoT environments respectively. Authors argue that better trusts management and provisioning enables the minimization of legitimate requests rejection.

An important aspect of network security is the localization and mitigation of failing or compromised elements. Chao et al. (2016) actively probe switches with a set of generated test packets coming from the controllers to ensure that all flow rules behave as expected. The controller also inserts test flow rules and collects statistics to identify inconsistencies used to locate the root cause of defect in packet handling and unintended flow rules. To mitigate a compromised switch, an application-level packet obfuscation is used by means of encryption to prevent the adversary from recovering the packet content without having the secret key. The K-shortest path algorithm is used to compute routing paths for the topology. Results show that the proposed scheme can reduce the number of test packets and can also cover every network rule. However, this scheme is limited to attacks against switches and does not address eavesdropping on packet content by compromised switches.

Summary and critical analysis. Architectural enhancements to SDN can prevent and mitigate several attacks to both control and data planes, normally the main targets for attackers is the controller, but also are the switches on SDNs because they tend to be limited on storage for flow rules. We analyzed and discussed proposals to avoid buffer overflow that span from adjusting control plane parameters such as fine-tuning timeout values for error messages to seek for idle buffers on switches whenever a switch is about to overflow storing flow rules. The case of adjusting timeout values has the limitation of being deployment-dependent since characteristics such as the physical technology used for interconnecting the network, the kind of network, and the processing power of the devices, namely the controller and switches, lead to different timeout values.

We also discussed how switches, with extra computational power, can detect fixed headers and install short-lived flow rules to quickly overcome DDoS attacks. Giving more general computing power to switches can lead to new security breaches if not designed properly. Finally, trust management and fault localization and mitigation are topics related to conventional networks but can be easily implemented in SDN because of its programmable nature and global network view information. Both strategies incur in heavy signaling between switches and controller. Tables 2 and 3 summarize related work on the strategies discussed earlier.

4.2. Strategies that provide SDN security capabilities

As mentioned in Section 3.2, this section discussed strategies that provide detection and mitigation of DDoS capabilities to SDN. We further categorize the strategies into Statistical, ML, and Blockchain whilst providing critical analysis on each subcategory.

4.2.1. Statistical strategies

A popular technique for statistical cyberattack detection is the use of entropy as a metric to measure the amount of uncertainty of randomness associated with incoming data flows. The more random the data is, the more entropy the system has. Comparing the entropy of two or more different classes of flows allows to detect changes in randomness. Kalkan et al. (2018) propose the Joint Entropy Based scoring system (JESS) that aims at detecting and mitigation DDoS attacks. It does so in three different stages: (i) in the *Nominal stage*, the SDN devices or switches forward all incoming packet headers to the controller which calculates the join entropy of each pair in attack-free periods; (ii) in the *Preparatory stage*, an attack is detected when the join entropy historically computed by the controller and the join entropy computed on the incoming pair exceeds a threshold; and finally, (iii) the *Active Migration stage* the switch creates a suspicious pairs (a pair with the maximum difference from the joint entropy) to send them to the controller in order to forward appropriate flow rules for mitigation. Performance evaluation of JESS is carried out by using the MAWI Working Group Traffic Archive dataset[1] to create a nominal profile. Mininet and an SDN simulator are also used for creating a realistic network topology and traffic flow. Several known attack types such as TCP SYN flood, DNS amplification, and other generic attacks are used for the evaluation. Experimental results show 80% success rate for unfamiliar attacks and 70% for mixed attacks. Authors argue that JESS performs efficiently even with low storage and processing devices. The switch-controller channel is usually secured by encryption protocols and is not meant to be used for constant traffic between devices. JESS, however, generates a constant flow of information between the switch and the controller. Thus, all information must be secured and encrypted/decrypted at each endpoint. Moreover, as data is flowing constantly, the channel can be exhausted, leading to a DoS.

Authors in Bavani et al. (2020) also take an entropy-based approach for the detection of malicious flows in SDN. The switch processes incoming packets by comparing header fields with the installed flow rules. If a packet does not match any rules, known as `table-miss` packets, it is forwarded to the controller for further inspection. An attacker can send large amounts of `table-miss` packets to exhaust resources on the controller. To overcome such attack, authors use an entropy-based algorithm that evaluates the amount of incoming packets for normal and attack flows. If the entropy of incoming packets is higher than the entropy of normal packets, the controller drops the packets coming from that IP. Experimental setup includes POX as SDN controller, Mininet and OVS for network topology and switching respectively and Scapy[2] for traffic generation. Authors claim a 97% attack detection with 25% of the total flows being malicious in a single controller environment. As a downside, only one metric (rate of incoming traffic) is used in this approach. During peak hours, incoming traffic increases, thus leading to false positives on legitimate traffic flows.

In order to overcome single metric entropy measurements, Duy et al. (2018) propose an architecture for the detection of DDoS attacks that is based on four components: (i) the *Packet collector* gathers incoming packets sent from switches to the controller and stores information about destination addresses; (ii) the *Entropy-based Sensor Engine* uses the collected data from the Packet collector to detect DDoS attacks. This component computes the entropy of flows based on IP addresses and a weight based on the role of the destination host (Web servers, storage servers, a proxy, user PC, etc.); (iii) the *Attack Confirmation Engine* uses a threshold to determine if certain flow is malicious; and (iv) the *Attack Mitigation Engine* for installing rules on switches to mitigate the attack. To evaluate performance of the proposed architecture, POX acts as a controller, Mininet provides the network topology, and Scapy generates traffic. Authors report that the system can detect and mitigate at early stages of the attack (4.07 s). We find that the assignment of weights depending on the role of the destination plays an important role in accurately measuring entropy since different targets expect different traffic patterns and rates. Moreover, the parameters such as the threshold, the calculation of the entropy and their corresponding weights should be topics of interest for future research.

Switches on SDN are plain devices whose sole purpose is to forward incoming packets according to their flow table entries. Shin et al. (2013b) propose the introduction of intelligence into SDN switches with the goal of keeping flows in the data plane as much as possible. Wang et al. (2015) and Kalkan et al. (2017) further extend this approach by implementing statistical algorithms in SDN switches for the detection and mitigation of DDoS attacks. During attack-free periods, nominal profiles are computed and sent to the controller for storing purposes only. If the bandwidth occupied at any switch exceeds a threshold, the switch asks for the profile to the controller, comparing the entropy of the incoming flow with the calculated entropy of the nominal profile. It is worth notice that, most of the calculations are executed in the switch. Experimental evaluation is carried out by simulations using the MAWI Working Group Traffic Archive. Authors claim a precision of 98% for the detection of TCP-SYN flood attacks, 100% for SQL Slammer Worm attacks, and 99% for DNS and NTP attacks. By providing some sort of intelligence to the switches, the latter can perform complex processing tasks such as extended packet inspection and decision making. This however, goes against the principle of plane separation proposed by the ONF for SDN. Moreover, the limited processing capacity of the switches makes them easier targets for DDoS attacks compared to a high-end controller computer. Moreover, we believe that the limited processing capacity from the switches makes them easy targets for DDoS attacks when compared to a high-end controller computer.

[1] https://mawi.wide.ad.jp/mawi/.

[2] https://pypi.org/project/ScapyTrafficGenerator/.

Table 2
SDN architecture-based detection and mitigation strategies. DoS: Denial of Service.

Authors (Year)	Focus of the paper	Type	Domain	Strategy	Shortcomings
Kandoi and Antikainen (2015)	To provide mechanisms to mitigate flooding attacks targeted the control plane bandwidth and the switch's flow table.	DoS	SDN	Improve configuration parameters to alleviate the effects of DoS attacks for dynamically changing the rate limit of the number of packets sent to the controller to avoid congestion.	Fine tuning of network parameters can be hard for large-scale networks.
Lin et al. (2017)	A defense mechanism to prevent buffer overflow in flooding attacks and controller overload.	DoS	DNS	PBUF detects if a buffer in a switch is about to overflow using network calculus to then find idle buffers on other switches to store flow rules for new packets.	Large-scale network deployments may incur in intensive processing times in the controller.
Durner et al. (2017)	Attacks on data plane that target the limited buffer size on switches	DoS	SDN	Hash the header of incoming packets to detect repeated entries and block the sender.	Large botnets possess enough computational power to change most header fields, overcoming the fixed headers approach for detection.
Wei and Fung (2015)	A buffer prioritizing algorithm for controller to handle routing requests based on trust values.	DoS	SDN	A ranking algorithm identifies regular users based on past requests and rank their flows with high priority. New attackers have low priority and will not forward messages before regular users.	Does not mitigate DoS attacks, but rather lets regular users to keep using the service.

Table 3
(Cont.) SDN architecture-based detection and mitigation strategies. DoS: Denial of Service. CI: Configuration Issues.

Authors (Year)	Focus of the paper	Type	Domain	Strategy	Shortcomings
Sarwar et al. (2019)	A trust-based request prioritizing algorithm for maximum utilization of trusted users.	DoS	IoT	Assigns high buffer priority for trusted users to maintain certain service level. Non-trusted users can be classified as malicious and be blacklisted.	If new flows arrive at the controller without a prior-known trusted value, the system cannot determine if they are trusted or not.
Chao et al. (2016)	A data plane system for localization and mitigation of comprised switches disobeying installed flow rules.	DoS, DL, DM	SDN	For fault localization, the controller generates packets probes to verify if switches follow installed flow rules. Also, flow rules designed to match expected behavior are also installed. Obfuscation techniques are used on packet headers to hide information from attackers for mitigation.	Careful design and generation of packet probes depend on the state of the network. Thus, a great number of packet probing is expected on large-scale deployments.

As we have described, the effectiveness of statistical strategies relies heavily on the definition of proper parameters. To provide more accurate computation of such parameters, Oo and Htein Maw (2019) propose the use of the well-known Exponential Weighted Moving Average (EWMA) technique to compute thresholds for their Modified Adaptive Threshold Algorithm (MATA). The latter enhances accuracy from 94.3% to 99.47% when compared to its predecessor the Adaptive Threshold Algorithm (ATA) (Siris and Papagalou, 2004). Experimental evaluation is carried out by using an ONOS controller, Mininet for the network topology, sFlow-RT to collect and analyze traffic flows, and two physical hosts connected to the network.

A feature of SDN that is heavily exploited to enhance security is the programmable nature of the controller. Rinaldi et al. (2019) propose the use of agents installed on top of the controller for anomaly detection. A logical element, called Traffic Agent Controller, is placed into the controller to collect statistical data of traffic flows, process the data, and to populate a centralized database. It uses the Kullback–Leibler Divergence (KLD) entropy (Kullback and Leibler, 1951). Morales et al. (2015) implemented a collection, detection, and mitigation module to cope with TCP-SYN flooding attacks by performing three phases: (i) the *collection phase* extracts data from TCP packets; (ii) the *detection phase* uses three detection algorithms, namely the Correlation Matrix, Shannon Entropy, and Chi Squared; and (iii) the *mitigation phase* a method is used to block the traffic from compromised hosts and new flow rules are installed in the switches. Performance evaluation uses Mininet for network virtualization. Results show that reaction time is limited by the time window required to detect statistical changes. Moreover, the maximum load this module can handle is limited to 40,000 packets per second.

Data plane is being overloaded with the deployment of multiple repetitive tasks of collecting, filtering, processing, calculating, and classifying the set of features per flow or connection to detect DDoS attacks using ML-based technique. This last issue was identified by Lapolli et al. (2019), where the proposed solution consisted of implementing at a programmable data plane, a DDoS detection system that analyzes the entropy of the IP addresses using a threshold to detect benign or anomalous traffic. Although this method shows excellent results, it only analyzes a single anomaly and, in a production scenario, high false positives could arise when the legitimate behavior of service users is highly random.

Summary and critical analysis. The use of entropy as a metric for differentiating between legitimate and malicious flows is feasible and widely used for anomaly-based attack detection. Khalaf et al. (2019) state that statistical approaches to DDoS can be further categorized into spectral analysis, statistical moments, and Markov hidden models. However, from our literature research we could only retrieve most relevant statistical approaches based on entropy. We also observe that, in most related work the entropy is computed solely based on traffic rates, without making distinctions of the type of host inside the SDN. By introducing weights to the calculation of entropy for each role, the accuracy of statistical-based detection strategies is enhanced. We believe there is still an area of opportunity in the study of computation of thresholds, entropies, and their corresponding weights for different scenarios in SDN. As mentioned before, the effectiveness of statistical strategies for attack detection and mitigation depend on the proper definition of the parameters discussed previously.

The introduction of some sort of intelligence for SDN switches goes against the principle of separation defined by the ONF, however it seems that is a tendency in order to improve some SDN weaknesses. SDN devices or switches are generally implemented in hardware as field-programmable gate arrays (FPGA) and firmware supporting OpenFlow's southbound protocols to deliver high-performance forwarding. Currently, there is a trend for the introduction of more CPU power into switches for better manageability and programmability to support newer protocol specifications (Foundation, 2020; Yazdinejad et al., 2019; Kaljic et al., 2019; Jiang et al., 2019). Researchers are proposing the use of such computational power from the switches to implement statistical strategies for the detection and mitigation of different attacks (Shin et al., 2013b; Wang et al., 2015; Kalkan et al., 2017). We believe this introduces some benefits like improving the intelligence

Table 4
Statistical-based detection and mitigation strategies. DoS: Denial of Service.

Authors (Year)	Focus of the paper	Type	Domain	Strategy	Shortcomings
Kalkan et al. (2018)	A joint entropy scoring system to detect and mitigate DDoS attacks.	DoS	SDN	Match the entropy of incoming flows and a nominal entropy. If a threshold is exceeded, the controller installs appropriate flows for mitigation.	Generates a constant flow between switches and controller.
Bavani et al. (2020)	Entropy-based algorithm to alleviate exhaustion of the switch-controller channel.	DoS	SDN	Match entropy of incoming flows with a nominal entropy. The controller observes the percentage of dropped packets to determine if traffic is malicious.	When traffic increases it can lead to false positives, blocking legitimate traffic flows.
Duy et al. (2018)	Entropy-based architecture for DDoS detection and mitigation based on roles.	DoS	SDN	Calculates the entropy of flows based on IP addresses and a weight based on the role of the destination. This allows different types of services (Web servers, storage servers, etc.) without blocking legitimate traffic.	Proper weight calculation for differentiated services.
Wang et al. (2015)	DDoS detection using entropy for anomaly detection on SDN switches.	DoS	SDN	Given some intelligence to switches, an entropy-based algorithm is used to detect anomalies of flows on switches.	Limited processing and storage capabilities on switches can lead to unresponsive devices.
Kalkan et al. (2017)	DDoS detection using entropy for anomaly detection on SDN switches.	DoS	SDN	Further extends (Wang et al., 2015) by enabling attribute selection for entropy and controller-side storage.	Limited processing capabilities on switches can lead to unresponsive devices.

Table 5
(Cont.) Statistical-based detection and mitigation strategies. DoS: Denial of Service. CI: Configuration Issues.

Authors (Year)	Focus of the paper	Type	Domain	Strategy	Shortcomings
Oo and Htein Maw (2019)	Enhancing threshold calculation on statistical-based approaches.	DoS	SDN	Enhances the Adaptive Threshold Algorithm (ATA) described in Siris and Papagalou (2004) with EWMA for better threshold value computation.	Limited size testbed with only two hosts in the network.
Rinaldi et al. (2019)	Agent-based IDS in SDN with Kullback–Leibler Divergence (KLD) entropy.	DoS and CI	SDN	An agent placed in the controller uses KLD entropy to classify legitimate and malicious flows.	Requires a good amount of processing power from the controller.
Morales et al. (2015)	A DDoS detection and mitigation module for Floodlight based on statistical classification.	DoS	SDN	A DCM (Data collection, Detection, and Mitigation) module is placed in the controller for traffic classification based on correlation matrix, entropy, and Chi-square statistical methods.	The reaction time of the system for classification is bounded by the time required to detect statistical changes (≈37.8 s).
Lapolli et al. (2019)	Detection of DoS attacks at the Programmable Data Plane	DoS	SDN	Calculation, at the programmable switch, of the entropy based on the IP addresses of the packets to detect, based on a threshold, of possible DoS attacks	High false positive rate in traffic with random behavior.

or performance of the switches in detecting DDoS attacks, but also it can introduce some security vulnerabilities since switches become easy targets for resource exhaustion attacks. So there are still some challenges concerning this issue. Tables 4 and 5 show a summary of statistical-based detection and mitigation strategies for SDN.

4.2.2. Machine learning strategies

There are three identifiable approaches to ML for DDoS detection and mitigation, namely the definition of frameworks or architectures that make use of ML algorithms internally to enhance detection rates, the enhancement of an IDS through ML techniques, and the ensemble of multiple ML algorithms to improve detection accuracy.

Peng et al. (2018) propose a classification and detection architecture for anomaly SDN flows using Double *P*-value of Transduction Confidence Machine — k-Nearest Neighbor (DPTCM-KNN) classifier. Matching the attack index becomes hard as the network burden enhances. A flow collection module collects information from the flow table in the controller and extracts the flow features, the detection mechanism performs classification by pre-processing it using DPTCM-KNN algorithm. The experiment uses 53,174 data flows, 2000 samples from training set and matlab2014a for simulation purposes. The limitation of this work is, massive network flows increases pressure on the controllers which leads to low precision and scalability. Mininet and Ryu controllers are also used for SDN simulation to test the performance. Authors claim that this technique has lower false positive rate and it reaches the maximum detection rate when normal to abnormal subset ratio is 4 to 3. The DPTCM-KNN is responsible for generating the anomaly detection report periodically with increased accuracy rate. DPTCM-KNN algorithm helps in effectively detecting the anomaly flow in the SDN environment.

Pérez-Díaz et al. (2020) also propose a flexible modular architecture that provides detection and mitigation of Low-Rate DDoS attacks in SDN. The architecture is composed of an IDS, trained with six ML models (i.e., J48, Random Tree, REP Tree, Random Forest, Multi-Layer Perceptron (MLP), and Support Vector Machines (SVM); and an IPS to mitigate the attack if a certain threshold is met. LR-DDoS are known to be difficult to detect since they follow a different approach to DDoS by carefully triggering specific protocol mechanisms to deplete the target's computing resources. The solution managed an accuracy up to 95% in a real simulated environment using ONOS as a controller and the SlowHttpTest library to simulate the attacks.

Macías et al. (2020) present ORACLE, an architecture implemented over an SDN/P4 environment that promotes the cooperation of both the control and data planes by means of programmable data planes to detect network attacks using Machine Learning models. In the evaluation of ORACLE, we obtained up to 96% of accuracy in the testing phase, using a K-Nearest Neighbor model.

On the other hand, there are several algorithms in ML that are used to enhance detection rates in IDSs. Latah and Toker (2018) study anomaly-based IDS using different supervised ML techniques in SDN controllers to tackle DoS, Remote to local (R2L), User to remote (U2R), and probe attacks. The framework is a lightweight IDS that performs detection along with routing and load balancing. The classifiers that are tested in this study include Neural Networks (NNs), Extreme Learning Machine (ELM), Support Vector Machines (SVM), AdaBoost, RUSBoost, LogitBoost, Random Forest (RF), K-Nearest-Neighbor (KNN), Decision Trees (DTs), BaggingTrees, Linear Discriminant Analysis (LDA), and

Table 6
Machine learning-based detection and mitigation strategies. DoS: Denial of Service.

Authors (Year)	Focus of the paper	Type	Domain	Strategy	Shortcomings
Latah and Toker (2018)	Comparison of multiple ML algorithms for the development of an anomaly-based IDS.	DoS	SDN	Multiple ML algorithms are compared in an IDS installed on a controller. Decision trees performs better for detection of potential SDN attacks.	Decision trees inefficiency towards incremental learning is not addressed.
Haider et al. (2020)	A deep convolution neural network framework for efficient and early detection of DDoS attacks in SDN.	DoS	SDN	An ensemble CNN algorithm that merges outputs of two CNNs to classify flows.	Does not achieve high accuracy, but runs on modest hardware.
Deepa et al. (2019)	Ensemble method with different ML algorithms for the detection of DDoS attacks.	DoS	SDN	Ensemble a combination of KNN, NB, SVM, and SOM algorithms are used to classify anomalous flows.	Ensemble methods often require demand more processing power.
Peng et al. (2018)	Classification and detection for anomaly SDN flows with KNN.	DoS	SDN	Information is collected from the flow tables to extract flow features. The detection mechanism performs classification using the DPTCM-KNN algorithm.	Massive network flows lowers the precision of the algorithm.

Table 7
(Cont.) Machine learning-based detection and mitigation strategies. DoS: Denial of Service. DL: Data Leakage.

Authors (Year)	Focus of the paper	Type	Domain	Strategy	Shortcomings
Zhu et al. (2018)	The goal is to achieve high accuracy and reasonable time consumption in cross-domain attack detection schemes.	DoS and DL	SDN	Predis employs KNN algorithm to provide cross-domain detection schemes with privacy preservation. The KNN is optimized through the collaboration multiple controllers completes the detection process.	Lower accuracy and higher time consumption of KNN algorithm.
Macías et al. (2020)	An architecture that promotes the coordination of control and data planes to detect network attacks using ML techniques	DoS	SDN	This architecture delegates to the data plane the extraction and processing of traffic information collected per flow while the control plane uses these features to perform the attack detection using the ML techniques RF and KNN	Detection of just one type of attacks.
Pérez-Díaz et al. (2020)	A flexible modular architecture for the detection and mitigation of low-rate DDoS attacks in SDN.	LR-DoS	SDN	A modular approach to detect and mitigate LR-DoS attacks where the ML model is executed and trained outside of the controller.	The communication between the controller and the detection module is not secured.

Naive Bayes (NB). Authors use a subset of features extracted from NSL-KDD[3] dataset based on principal component analysis (PCA). Results show that Logitboost was efficient in terms of false alarm rare (FAR) and Recall. ELM in terms of execution time. Despite KNN's good accuracy, due to its bad testing time bagging and boosting is preferred. Overall DTs have maximum accuracy, and is best suited in terms of precision, F1 measure, AUC and McNemar's test. DT's inefficiency towards incremental learning is not mentioned in this work. This Comparative study on different classifiers to choose an efficient classifier showed that Decisions Tree is more effective in detecting the potential SDN attacks.

Zhu et al. (2018) follows the approach of enhancing IDS by proposing Predis, a cross-domain attack detection scheme which preserves privacy in SDN. It combines digital cryptography with perturbation encryption to address data privacy. To perform cross-domain attack detection, a large amount of real traffic data is required which may lead to network privacy concerns. Existing schemes provide privacy by compromising accuracy and time cost. In their work, authors consider DDoS, user-to-remote (U2R), probing and DoS attacks suitable for their study. Predis employs a k-Nearest Neighbors (KNN) algorithm to overcome this problem and improve efficiency. The dataset is obtained by traffic traces which are public and one among them is CAIDA trace.[4] Even tough KNN has no ideal accuracy or ideal time consumption, it is employed because it is easy to calculate. The KNN is optimized and the collaboration of servers completes the detection process. Results show that Predis consumed less time relatively in detecting the given anomaly with high accuracy.

Finally, the ensemble of multiple ML algorithms is gaining attention due to their high accuracy when classifying DDoS attacks. Haider et al. (2020) propose a deep convolution neural network (CNN) framework for early detection of DDoS attacks and a deep CNN aimed at detecting flow-based DDoS attacks. A hybrid deep learning model is used for an anomaly-based IDS where the outputs of two CNNs are merged and passed to an ensemble CNN for classifying traffic as legitimate or malicious traffic. Authors use the Keras Library[5] with a Tensorflow backend (Abadi et al., 2016) and the CICIDS2017 dataset (Sharafaldin. et al., 2018), which is a purely flow-based state-of-the-art SDN dataset for performance evaluation. Results show that ensemble CNN has an accuracy of 99.45% running on an Intel i7-6700 CPU at 3.4 GHz and 8 GB of RAM.

Another ensemble is described by Deepa et al. (2019), where KNN, NB, SVM, and Self-Organizing Maps (SOM) are used to detect anomalous traffic flows in the controller. For experimental evaluation, Mininet is used for the creation of virtual hosts, controllers and switches, POX as a controller, and the CAIDA dataset. Results show that NB–SOM and SVM–SOM provide 97.19% and 98.12% accuracy respectively.

Summary and critical analysis. We are seeing a recent trend in ensemble methods where several ML algorithms are combined to enhance accuracy in classification tasks. ML strategies are being used for detection and mitigation of DDoS attacks since the controller provides programmability capabilities that enable an easy implementation. In the near future, many ML strategies are expected to be present in multi-domain solutions, exchanging information about attacks in a distributed manner to achieve better decision-making capabilities. Tables 6 and 7 show a summary of our literature review.

Broadly speaking, one challenge that is recurrent in all ML strategies is the lack of organic normalized SDN flow-based datasets to evaluate their performance (Nguyen, 2018). There are plenty of datasets that contain information about real-world DDoS attacks on conventional packet-based networks. SDN on the other, logs flow-based traffic. The

[3] https://www.unb.ca/cic/datasets/nsl.html.
[4] https://www.caida.org/data.
[5] https://github.com/keras-team/keras.

conversion of packet-based to flow-based traffic data becomes a problem because there is no universal procedure to achieve the translation. This leads to another problem regarding the variability in input data for the algorithms. The way data is stored in DDoS attack information datasets may differ. Therefore, some datasets may be easier to use than others. This becomes even more evident when a translation from packet-based to flow-based data is required. Additional challenges are presented when the trained models are tested in real or simulated environments, it is observed that the very high accuracy gotten with the datasets are not obtained in real environments this could be the reason most of the papers only present the results gotten in the training and testing phased with the datasets. These differences are presented most of the times because the attacks in the real or simulated environment are not executed with the same tools that were used to create the dataset, so this problem opens a research issue to solve for future investigations.

Nguyen (2018) states that there is a gap between academic strategies on ML based solutions for SDNs and their operational deployments. This is because academic proposals do not meet strict requirements imposed by commercial deployments. The author recommends making all approaches auditable, to follow a secure development process and to follow best practices.

Finally another tendency to detect DDoS attacks using Artificial intelligence algorithms is the use of deep learning algorithms which have shown better performance than ML algorithms in the detection of DDoS attacks (Liang and Znati, 2019; Tang et al., 2020). For more information about ML defense strategies for SDN, please refer to the following surveys (Sultana et al., 2019; Xie et al., 2019; Zhao et al., 2019).

4.2.3. Blockchain strategies

Blockchains for DDoS attack detection and mitigation in SDN are being used for secure data exchange and trust management to aid in the installation of flow rules and data exchange across inter-domain networks. Most blockchain implementation in SDN are based on private or consortium ledgers. In such cases, the SDN controller is issued as a central server that is often used as a trusted point to help manage data sharing and trust computation among collaborating parties.

Singh et al. (2019) study different approaches to DDoS mitigation using blockchain in conventional and SDNs networks. Most DDoS protection strategies use blockchain as a mean to maintain and record transactions through smart contracts. Authors state that the use of SDN and blockchain in mitigating DDoS is plausible when SDN is used for filtering and detecting whereas blockchain is used for publishing attack trust information on inter and intra-domains.

Tselios et al. (2017) consider the use of blockchain to record transactions inside SDN-based IoT environments. Recent DDoS attacks such as the Mirai botnet in 2016 attracted a lot of attention from the community and researchers as well (Kolias et al., 2017). Moreover, blockchain-based identity and access management can be leveraged to strengthen IoT security (Kshetri, 2017). Authors envisioned sensors and interconnected nodes having access to the Genesis block to create new immutable blocks and update the existing blockchain about the data that is generated by IoT devices. This enables authentication and authorization tasks to be straightforward, preventing DDoS attacks from generating inside the SDN. No expensive packet inspection is needed, and inter-cloud communication becomes more efficient since blockchain enables trustless networks. However, there is no clear information about the type of blockchain (public, private, or consortium) that could be implemented in an IoT deployment based on SDN.

Shafi and Basit (2019) explore the idea of using blockchain and SDN in IoT networks to prevent DDoS attacks and to prevent hosts of becoming part of the botnet force. Bots are applications that run automated tasks over the Internet. Several of these bots are used together to perform distributed DoS attacks. To prevent such attacks, SDN controllers are linked to a distributed blockchain. The system verifies flow rules by downloading an authenticated flow table and its features, backed by the blockchain to avoid IoT devices being compromised and becoming part of the botnet. Network resources must be prevented not only from DDoS attacks, but also from becoming the botnet itself that launches DDoS attacks. The architecture presented in the paper provides a mechanism to identify security policy changes implemented in the system data plan and network's topological changes. This aids to detect IoT traffic being diverted to incorrect destinations. A relocation agent is also used for the detection of botnet targets and for the management of an alarm. Experiments show that the architecture can accurately detect IoT devices under botnet attack. The Installation of flow rules is done by the controller on any switch where potential unwanted traffic originates, which can lead to botnet prevention.

Blockchain is also being used for enabling collaborative DDoS defenses in SDNs. Existing approaches such as the DDoS Open Threat Signaling (DOTS) (Mortensen et al., 2020) require specialized hardware and software to exchange context information about DDoS attacks. Rodrigues et al. (2017) use existing DDoS detection and mitigation modules to detect flooding attacks. Upon attack detection, an Autonomous System (AS) requests cooperative defense by submitting a transaction to the blockchain through a Smart Contract (SC) to the AS that is managing the IP network address of the attacker. El Houda et al. (2019) propose a collaborative scheme for IoT environments around the same idea of using blockchain as a secure inter and intra-domain communication channel.

Abou El Houda et al. (2019) also propose CoChain-SC, a blockchain-based approach to perform intra-domain and inter-domain DDoS mitigation. During a typical DDoS attack, a large number of compromised devices are requested by an attacker to flood the target directly to saturate its resources. CoChain is an approach which enhances the accuracy of attack detection model by combining an intra-domain DDoS mitigation scheme using machine learning techniques and an inter-domain DDoS scheme to efficiently mitigate along the path of an attack and near the origin of attack. The four modules in intra-based domain mitigation schemes architecture are Intra Entropy based scheme (I-ES) which uses sFlow to measures the randomness of data inside the domain, Intra Bayes-based scheme(I-BS) which uses the entropy values to classify the illegitimate flows, Intra-Domain mitigation scheme mitigates illegitimate flows inside the domain and blockchain layer. In inter-domain mitigation scheme, multiple SDN based domains are allowed to collaborate securely and transfer attack information in a decentralized manner based on blockchain using smart contracts. Ethereum blockchain is the decentralized platform used for developing smart contracts. The experiment conducted for attacks varying in the range between 100 to 500 Mbps, where in the intra-domain mitigation scheme, the Cochain-SC achieved a 100% detection rate and a 26% False Positive Rate (FPR) for 100 Mbps and similarly a 100% detection rate was achieved for 500 Mbps and a 23% FPR was detected. The CoChain-SC provides an easy to deploy, secure and low cost DDoS attacks mitigation scheme. Authors claim that Cochain-SC preserves pseudo-anonymity and does not allow any IP address traceability. Also, since it runs on Ethereum, there is no single point of failure and due to the decentralized scheme, there is no centralized authority to maintain the collaboration system. The reliability and the availability of the record on the blockchain are guaranteed.

Finally, Steichen et al. (2017) propose ChainGuard, an OpenFlow-based firewall for blockchain applications for mitigating flooding attacks. ChainGuard keeps track of all remote blockchain nodes to determine their legitimacy. If a node is allowed to connect with the blockchain, ChainGuard considers it is a legitimate node and thus, it does not interfere with the node's forwarding mechanisms and the node is added to a whitelist. If a node is yet to be considered by ChainGuard, it alerts of an attempt of connecting remote nodes to the blockchain, it then adds the node to a fixed-size graylist. If ChainGuard determines that the remote node does not have permission to connect with the blockchain, it treats the node as Illegitimate and is added to a blacklist.

Table 8
Blockchain-based detection and mitigation strategies. DoS: Denial of Service. CI: Configuration Issues.

Authors (Year)	Focus of the paper	Type	Domain	Strategy	Shortcomings
Rodrigues et al. (2017)	To provide a coordinated protection effort to extend defense capabilities of a single system using blockchain for signaling and also financial incentives.	DoS	SDN	The system relies on existing DDoS detection and mitigation modules to obtain addresses to be reported through the blockchain. Individual Autonomous Systems (AS) retrieved reported addresses by other ASes.	Masive amount of storage if the number of reported IP addresses scale. It is not clear why SDN is part of the solution. The use of a public blockchain eliminates the use of the SDN controller as a trusted entity.
Tselios et al. (2017)	Architectural design to secure blockchain-based inter-cloud communication for SDN in IoT environments.	UA, DM, DoS	IoT	Secures the control plane by introducing blockchain to the communication between controllers and devices. Authentication is straight-forward and does not require packet inspection.	Node heterogeneity (processing power, battery life, communication technology) capabilities of IoT nodes limits their involvement in block verification process.
Steichen et al. (2017)	Blockchain-based OpenFlow firewall for flooding attack mitigation.	DoS	SDN	ChainGuard whitelists legitimate nodes, whilst greylisting nodes wanting to participate in the blockchain and blacklisting malicious nodes. Install short-lived flow entries for malicious nodes.	Experimentation used a limited number of nodes. It is not clear if viable for a large-scale network with heterogeneous nodes.

Table 9
(Cont.) Blockchain-based detection and mitigation strategies. DoS: Denial of Service. CI: Configuration Issues.

Authors (Year)	Focus of the paper	Type	Domain	Strategy	Shortcomings
Shafi and Basit (2019)	A DDoS botnet prevention mechanism for IoT devices based on blockchain.	DoS	SDN	A link between SDN controllers based on blockchain with a shared and authenticated flow table for switches. It identifies policy changes based on a data plan and topology changes.	Authors use virtual links between VM hosts to share blockchain information, assuming that channels are secured. This channel can be however compromised since nodes in IoT can be appended dynamically into the network.
Abou El Houda et al. (2019)	A blockchain-based intra and interdomain DDoS mitigation.	DoS	SDN	Multiple SDN-based domains collaborate to share information using a private blockchain. Authorized participants are added to the blockchain to store suspicious IP addresses.	Experimental evaluation shows high false positive rates.
El Houda et al. (2019)	To provide a secure and decentralized DDoS collaboration scheme based on blockchain using SC.	DoS	SDN	Multiple SDN-based domains collaborate to share information using a private blockchain. Authorized participants are added to the blockchain to store suspicious IP addresses.	There is not clear information about the owner of the smart contract and what a participant is (SDN controllers, hosts on a domain, etc.). When using a public blockchain, the need of a trusted third-party entity such as the SDN controller becomes irrelevant.

In order to mitigate flooding attacks from multiple sources, the firewall generates and installs short-lived flow entries on the switches to drop packets for hosts that belong to the graylist. The small timeout values allows quick removal of installed entries in the switch, preventing overflow issues. Experiments show that ChainGuard is able to drop multiple malicious packets before reaching the blockchain.

Summary and critical analysis. Blockchain strategies are still in early stages. Current solutions hardly scale in terms of size, cost, and effectiveness. SDN technology is a vital tool along with blockchain technology for mitigating DDoS attacks (Singh et al., 2019). The relation between SDN, DDoS, and blockchain is mainly based on securing inter and intra-domain communication(s) between controllers and switches. Thus, blockchain can be used to securely transport critical and context attack information, which can be traced and validated, such as flow rules for switches and information about malicious nodes in a collaboration between controllers to create databases for quick malicious node detection and response. Such approaches still require capable IDSs and IPSs to detect anomalies and malicious nodes and to select the most appropriate flow rules for different attacks. Current approaches to blockchain and DDoS detection and mitigation are the definition of frameworks that allow the collaboration between networks to share information in a secure, verifiable manner of malicious nodes and attacks to preemptively take actions in avoiding DDoS attacks. In the same way, flow tables and their features backed by the blockchain are helpful to identify IoT devices being compromised and becoming part of the botnet that could potentially launch a DDoS attack.

Secure inter-cloud communication in SDN is important since new nodes are dropped into the network that can act as malicious hosts and harm the network. Blockchain is an emerging technology with a distributed data structure that cannot be tampered. Therefore, blockchain can play a significant role in securing the inter-domain communication in SDN such as controller-to-controller communication, switch-to-controller communication, and switch-to-switch communication.

Blockchain can be used to differentiate between legitimate and malicious nodes in a network, leading to different levels of treatment for such nodes. By design, blockchain also provides reliable past and current information because data in the blockchain is verified by other participating nodes. Therefore, blockchain technology can be also used as a tool for the design of digital forensics systems in order to recollect evidence and perform a root cause analysis of the attack.

One key challenge in all blockchain technology is scalability, with an increased size of distributed ledgers and a growing number of participants, the storage required for blockchain to operate must be optimized. If storage is not properly designed, performance on the operation of the blockchain can be greatly degraded. In IoT environments, due to the heterogeneous nature in processing power, memory capacity, etc., of devices, some may not be able to properly hash and encrypt data blocks fast enough, leading to soft forking and scalability issues.

Another challenge for blockchain, and specially in IoT environments, is the energy and cost at which mining blocks for the calculation and verification of transactions implies. Due to the complexity of blockchain-based transactions, the participating parties may act on out-of-date information. Tables 8 and 9 show the summary of blockchain-based cyberattack detection and mitigation strategies.

4.3. Strategies that use SDN as an enabler

As mentioned in Section 3.3, emerging technologies such as 5G could not be possible, or at least could be really hard, to deploy without the features of SDN. Therefore, this section describes the strategies that provide detection and mitigation of DDoS attacks in environments that use SDN as an enabling technology. We further categorize the strategies

into NFV, Honeynet, NS, and MTD. We also provide critical analysis for each subcategory.

4.3.1. Network function virtualization strategies

NFV has been proposed as a new network paradigm complementary to SDN (Matias et al., 2015). Therefore, it is not possible to think on detection and mitigation strategies in NFV to attacks targeted at the SDN architecture; in fact, the approaches present in the literature target security attacks using NFV and SDN as complementary technologies.

Monshizadeh et al. (2017) propose the use of an intrusion detection system (IDS) based on SDN and NFV to perform early DDoS detection to provide isolated environments for target hosts. The architecture mirrors incoming traffic into a clustering system and forwards original traffic to its destination. The mirrored packet is then analyzed by a corresponding Detection as a Service (DaaS) node to determine whether the traffic is malicious or normal. If malicious, the DaaS informs the SDN application that the flow should be blocked. The proposed system implements a load balancer to determine to which DaaS node the traffic will be sent. In this case, NFV is used to perform DDoS detection and SDN is used as the mitigation strategy, nevertheless, neither the method to detect if traffic is malicious nor the load balancing algorithm for the cluster node are provided. However, the architecture provides the basis for IDS based on NFV that can be further be developed in future research.

Zhou and Guo (2017) investigate a mitigation framework employing NFV and SDN to detect and block DDoS attacks. Authors provide an architecture based on three functional blocks: an NFV orchestrator, an NFV manager and a virtual infrastructure manager (VIM) as proposed in the NFV MANO architecture (Mijumbi et al., 2016). The orchestrator is responsible for the management and coordination of software resources and virtualized hardware infrastructure in order to provide optimal resource allocation. The NFV manager is responsible of the instantiation and termination of virtual instances. Finally, the VIM is responsible for virtualizing and managing networking resources in the NFV infrastructure. In this paper, authors propose the integration of these modules with the SDN architecture to perform attack detection and mitigation. They propose an architecture based on three planes: data, control and application. The data plane is composed of VNFs, NFVI virtualized resources and physical resources; the control plane contains all the management functionalities: NFV orchestrator, VNF Manager, VIM and SDN controller; finally, the application plane may include applications such as load balancing, access control, maintenance, network security and routing. The idea is that the SDN controller implements a DDoS detection module using its monitoring capabilities and, when an anomaly is detected, it is communicated to the application plane that, in turn, instructs the control plane to start virtualizing and instantiating the needed VNF to mitigate the attack. It is worth notice that this is a theoretical framework and is not clear which methods for anomaly detection are used nor how instantiation and management are performed. It is also important to note that this proposal is not implemented in any NFV framework so it is very difficult to validate it.

Luo et al. (2020) study the use of service function chains (SFC) with NFV and SDN to provide network security in a cloud computing environment. A SFC defines an ordered list of service functions to dynamically forward network traffic through various service function paths. A security service function tree architecture combines multiple SFCs for providing a variety of security services. The main goal of this work is to optimize the resource allocation when deploying multiple security SFC for multiple tenants in a cloud environment. To do that, authors propose to combine multiple SFCs into a security service function tree (SecSFT) to reduce requirement for resources in allocating virtual security functions. SDN is used in the deployment of the SecSFT; the SDN controller send different flow tables to OpenFlow switches according to the types of security services and the tree topologies. When network traffic arrives at the entry of a SecSFT, it will be led to different branches of the tree according to the decision rules. Incoming traffic attributes are collected by NFV nodes and matched with the decision rules of the node, mainly to detect SYN and UDP flooding (DDoS) and (scanning) attacks. The decision rules are different for each subsection of the tree and go through leaf NFV nodes, to inner foremost VNFs according to the tree structure. Experimental results were gathered using ONOS, Docker for NFVs, and OVS for the switches, showing accuracy of 98.2% for SYN flooding DDoS attacks, 97.2% for UDP flooding, and 98.6% for IPsweep attacks. While this approach does perform an evaluation based on real experiment settings, it still does not implement an integrated SDN–NFV architecture.

Authors of Liu et al. (2018) propose a Moving Target Defense (MTD)-based approach for SDN/NFV networks to achieve DDoS detection and mitigation. The paper propose to use the programmability of SDN and the scalability of NFV in the following way: a fuzzy logic analysis system is implemented to detect DDoS traffic, in addition to prevent this traffic to affect the target victim, NFV is used by implementing a virtual relay node as the base of the MTD mechanism (this virtual node transfers the DDoS attack surface isolating suspicious traffic from normal traffic, and redirects suspicious traffic to a restricted node using SDN by modifying flow rules). Results show a detection rate of 99.4%. In this paper, the evaluation is devoted to the accuracy of the proposed MTD-based approach, but the experiment settings are not implemented in a real NFV–SDN architecture.

Summary and critical analysis. NFV opens new opportunities for the dynamic deployment of security mechanisms along an SDN network. Moreover, NFV allows to implement isolated environments to detect and mitigate current cyberattack threats with automated countermeasures. NFV is a promising technology not only for security-related issues, but for complete network automation. The main shortcoming identified in the reviewed works is the lack of real implementations using NFV-enabled platforms that implement the NFV-MANO architectures (i.e. OPNFV, OSM, etc.) and their integration with SDN controllers in order to validate the DDoS detection and mitigation mechanisms. Also, existing proposals do not clearly specify how the integration of SDN and NFV can handle DDoS detection and mitigation. With regard to DDoS detection, a good branch for future research can be the study of early detection using SDN and programmable data planes (Foundation, 2020) triggering the deployment of virtual IDS as VNFs close to the detection point in order to verify the presence of anomalous traffic and activate a mitigation mechanism. Regarding mitigation, VNFs to quarantine traffic can be deployed and interconnected using SDN, also SDN rules could be installed to divert or reject anomalous traffic; in order to know more about the attack, fake VNFs could also be installed so that the attacker thinks he/she succeeded.

The shortcomings associated with NFV are the computing resources needed to deploy several services. Tables 10 and 11 show a summary of NFV-based strategies for cyberattack detection and mitigation.

4.3.2. Honeynet strategies

Honeynets are a defensive technology used to capture attackers by deploying several hosts and networks to trick malicious hosts to perform an attack. The samples gathered are then analyzed for further inspection and eventually build defenses into a production stage. Honeypots can be classified as HIH (High Interaction Honeypots) and LIH (Low Interaction Honeypots), both depends on the level of interaction, while a LIH just simulates some network features a HIH is the whole OS. However, a HIH is expensive to deploy and maintain because the application (the attacking target service) and the OS are implemented on real hardware, thus a choice must be made before deploying either options. On the other hand, a LIH implements the application and the OS as a virtual environment. A mix of LIH and HIH are called Hybrid Honeypots. They usually have several front-ends (LIH's) which can simulate thousands of IPs to attract attacker's and several back-ends (HIH's) to interact directly with the attacker and gather information

Table 10
NFV-based detection and mitigation strategies. DoS: Denial of Service.

Authors (Year)	Focus of the paper	Type	Domain	Strategy	Shortcomings
Monshizadeh et al. (2017)	An architecture that combines IDS with programmability features of SDN and NFV for detection and mitigation of malicious traffic	DoS	SDN	Incoming traffic is sent into a cluster to determine if it is malicious or not. If malicious, an SDN application running on top of the SDN controller blocks the flow.	Bandwidth usage inside the SDN is doubled since the incoming traffic is mirrored into a cluster of IDS implemented as NFVs. There is no description of detection schemes for malicious traffic. No real implementation is provided.
Zhou and Guo (2017)	A DDoS mitigation framework employing NFV and SDN to detect and block attacks.	DoS	SDN	An NFV module is responsible for virtualization and instantiation of network functions for specific mitigation requirements as detected by the controller.	Anomaly detection mechanisms are not specified. Network function requirements must be defined beforehand to mitigate known attacks. No real implementation is provided.

Table 11
(Cont.) NFV-based detection and mitigation strategies. DoS: Denial of Service.

Authors (Year)	Focus of the paper	Type	Domain	Strategy	Shortcomings
Liu et al. (2018)	DDoS defense mechanism based on SDN/NFV and moving target defense	DoS	SDN	It uses fuzzy decision to detect DDoS attack and MTD for mitigation.	Detection delay is not mentioned, also the time spent by the mitigation mechanism is not analyzed. It is difficult to know if the strategy is adequate for real time. The evaluation is more concentrated in the accuracy of the method than in the implementation of NFV–SDN.
Luo et al. (2020)	A tree-based architecture using SFC and NFV to provide multiple security mechanisms at different levels of the SDN in a cloud environment.	DoS	Cloud	Each leaf NFV node has its own SFC with flow rules for different security requirements. Incoming traffic is classified accordingly at each NFV node to determine if the traffic should go deeper in the tree to eventually block the flow.	For each type of attack, a set of SFCs and NFV nodes must be deployed. Thus, several virtual instances must be set for large network deployments. The work does not implement an integrated SDN–NFV architecture for experimentation.

about the attack. However Hybrid honeypot architecture have issues maintaining data flow.

Wang and Wu (2019) propose an SDN hybrid honeypot architecture consisting of four parts: an OpenFlow switch, an SDN controller, a HIH group, and a LIH. An *attack traffic migrating system* differentiates and classifies attack types based on SDN for low and high level attacks. The SDN controller determines the type of the attack and an OpenFlow switch forwards the traffic according to its type. Once the honeypot model distinguishes the attack type, a message is sent to an HIH if it is a low level attack, and to a LIH if it is a high level attack. An *SDN-based topology simulation model*, uses virtual OVS nodes and controllers to simulate the main functions of a router. Mininet and Floodlight are used to provide experimental results that show low delay during the attack traffic migration phase.

The sharing information about Cyber Threat Intelligence (CTI) can get an upper edge against cyber criminals and cyberattacks. Authors in Pan et al. (2016) propose HogMap, an infrastructure that facilitates collaboration, embracing SDN's paradigms of programmable network switches which facilitates seamless exchange of global attack activity across distributed participants. It enables a dynamic threat intelligence collaboration to produce a unified threat intelligence surveillance system. HogMap uses a honeygrid, which provides all basic infrastructure needed to make the infrastructure work. Providers require an OpenFlow switch gateway to deploy certified apps to participate in several services and earn revenue from it. HogMap allows to filter malicious traffic, migrate virtual machine instances to mitigate attacks, and enables session migration. Moreover, collaborative information across service providers may be used to elaborate effective mitigation techniques.

Kim and Shin (2017) state that Link Flooding Attacks (LFA) are indistinguishable and undetectable DDoS attacks that aim at exhausting bandwidth of intermediate links. An attacker can build a link map of intermediate routers with tools such as traceroute and launch attacks. A honeynet topology is used to expose a fake topology to the outside world in order to attract intruders to perform LFA and redirect traceroute commands to the honeynet. This allows to clean up the traffic from the real network and forces the attacker to visit the furthermost node in the honeynet topology so the attacker visits the fake topology. The mitigation technique used in this work allows to gather information of the attacker whilst redirecting malicious traceroute packets into the honeynet.

Zarca et al. (2020) propose a dynamic IoT honeynet deployment framework to alleviate complex and tedious tasks of configuring, setting up, and deploying honeynets in heterogeneous environments such as IoT. The network administrator defines policies for authentication, authorization, filtering, channel protection and forwarding, using high-level models which are processed by the system and translated into low-level policies required to configure different honeynets. The framework is aimed at simulating a real IoT sensor network for distraction and reaction accordingly to provide countermeasures to mitigate attacks and for gathering information for future threats. Performance evaluation of the framework involves several technologies such as ONOS as controller, real and virtual Contiki motes, Cooja as an IoT network simulator, and NFV orchestrators. Results show better memory consumption (102.4 MB) than other IoT honeynet deployments (726 MB) (Fan et al., 2017).

Summary and critical analysis. Honeynet is a defensive mechanism that imitates the activities and configurations of a production network in order to deceive attackers, making them waste time while attacking an isolated virtual network environment. Honeypots are considered to be a strong defensive tool for modern IDS in complementing their functionality (Dalamagkas et al., 2019). Wang and Wu (2019) provide a hybrid honeynet architecture in which a HIH or a LIH are deployed depending on the attack. One of the main goals of honeynets is to gather information about attacks in isolated environments in order to determine proper countermeasures for future attacks. Pan et al. (2016) propose an infrastructure to share cyberattack information among network operators whilst detecting and filtering cyberattacks using honeynets. Kim and Shin (2017) use honeynets to locate link flooding attacks. Finally, Zarca et al. (2020) implement a framework for the deployment of policy-based security honeynets using high-level policies for authentication, authorization, filtering, channel protection and forwarding and enforcing such policies to mitigate attacks. Tables 12 and 13 show a summary of honeynet-based detection and mitigation strategies.

In order to capture high quality data from a Honeynet, a higher interaction honeypot is required (Fan et al., 2018). However, this

Table 12
Honeynet-based detection and mitigation strategies. DoS: Denial of Service.

Authors (Year)	Focus of the paper	Type	Domain	Strategy	Shortcomings
Wang and Wu (2019)	An architecture for SDN using hybrid honeypot systems based on network topology simulation and attack traffic migration.	Several	SDN	An attack traffic migration mechanism that classifies different attack types and migrates traffic to honeynets according to such classification.	Experimental evaluations are performed at a rate of 10 connections per second, which is fairly low compared to traffic generated in large-scale network deployments.
Pan et al. (2016)	To transform cyberthreat monitoring landscape by integrating an SDN-based HoneyGrid as it allows to collaborate and overcome malware propagation through affiliate programs.	Several	SDN	HogMap is a marketplace that facilitates threat intelligence across honeynet implementations. The infrastructure is a honeygrid that shares information about threats between HogMap-certified SDN applications. It enables dynamic traffic filter schemes and intelligent honeypot management.	The communication among HogMap providers is not encrypted. Future work describes the use of VPNs to support secure channel communication.

Table 13
(Cont.) Honeynet-based detection and mitigation strategies. DoS: Denial of Service.

Authors (Year)	Focus of the paper	Type	Domain	Strategy	Shortcomings
Kim and Shin (2017)	An SDN system that exposes a fake network topology to attackers and find potential bottleneck links.	DoS	SDN	SDHoneyNet exposes a fake topology to attackers by finding potential bottleneck links and deploying a honey topology. A random graph algorithm for generating scale-free networks is used to deceive attackers and to force the attacker to fully visit the honey topology. This allows to detect an attacker and mitigate the attack by filter its connections.	Authors do not yet consider the weights of network properties that allow to accurately locate bottleneck links in real-world infrastructure.
Zarca et al. (2020)	A policy-based honeynet deployment framework for IoT.	Several	IoT	The framework simulates a real IoT sensors network for distraction and reaction accordingly by enforcing pre-defined policies to mitigate attacks.	It is not clear how the Security Orchestrator module deals with contradictory or mutual exclusive policies. This is specially important in multi-controller deployments where several policies are applied network-wide.

leads to higher risks of being compromised since, as mentioned earlier, either the Operating System (OS) where the honeypot is deployed, or the application and the OS where the honeypot is deployed are implemented on real hardware; thus sacrificing scalability due to the use of real hardware for deployments.

4.3.3. Network slicing strategies

The upcoming 5G mobile network aims at providing massive machine-type communications (mMTC), enhanced mobile broadband (eMBB) with high data rate requirements, and ultra-reliable low latency communications (uRRLC) for supporting critical applications with real-time constraints. In order to provide such services, 5G uses virtual dedicated networks called network slices (NS). NS provide control plane functionalities such as policy control, access, mobility, session, and authentication management as well as network slice selection functions. As in SDN, firewalls, DPIs, load balancers, IDS and IPS are data plane functions provided by NS. Although NS aims at providing different service configuration through dedicated virtualized networks, it is also being proposed for several attack detection and mitigation strategies (Sathi et al., 2020).

Sattar and Matrawy (2019) provide a mathematical model to deploy network slices based on security constraints and reduce the impact of DDoS attacks in 5G networks. Slice isolation (both resource and network isolation) is a requirement for 5G since it allows better management and administration (Li et al., 2017) and can be further categorized in inter-slice and intra-slice isolation. In inter-slice isolation, dedicated hardware infrastructure is allocated for a slice, providing strong isolation since service degradation only affects one slice. On the other hand, inter-slice isolation refers to the separation of components of slices between hardware infrastructure. Authors use a mathematical model to allocate network slices (inter and intra) depending on network requirements and evaluate the use of both isolation techniques for DDoS attack mitigation. Results show a trade-off between strong inter-slice isolation and intra-slice isolation. Under attack, strong isolation allows other slices to continue functioning at the cost of reducing efficiency of network resource utilization. Intra-slice isolation provides better performance than no isolation.

Thantharate et al. (2020) develop a Secure5G model using deep learning techniques for NS aiming at the identification of incoming packets to assign optimal slices for each traffic type whilst performing threat analysis on the packets for potential threats on further actions. The aim of the Secure5G model is to mitigate DDoS initiation attacks from user equipment, by making sure the devices can access network slices only after being authenticated and/or authorized (i.e., minimizing the risk of DDoS attacks). The model also predicts future traffic for accurate NS allocation into one of the three categories: eMBB, uRRLC, and mMTC. Authors evaluate Secure5G by mitigating flooding attacks and masking botnets (spoofing attacks). Results show 98% detection accuracy with the DeepSlice dataset.[6]

Bonfim et al. (2020) propose the FrameRTP4 framework aimed at detection and mitigation of DDoS attacks in real-time for the upcoming 5G mobile technology. In order to achieve real-time detection, the rate at which flows are processed should match the rate at which the used communication technology works. Therefore, the framework makes use of P4 to provide flow rules to switches based on ACL and wildcards for fast processing. The detection is performed through offline ML techniques based on Random Forest in a module placed at the controller called *decision making module*. The module is responsible for the creation of proper flow rules based on ACL and wildcards with their respective set of actions to mitigate malicious traffic. The *FrameRTP4 Orchestrator*, controls one or more FrameRTP4 Controllers to provide security for end-to-end Network Slides, it also enables a third-party management system to perform the life-cycle management of SFCs (ie, NS instances) and P4 table rules (adding and deletion). Authors evaluate the monitoring tool and wildcard compression accuracy. No detection and mitigation of DDoS accuracy evaluation is performed.

Summary and critical analysis. Network slicing is an important component of the upcoming 5G mobile networks, since NFV and SDN are envisioned as the key enabling approaches of this technology, they get intrinsically the security features managed by network slicing

[6] https://github.com/adtmv7/DeepSlice.

Table 14
Network slicing-based detection and mitigation strategies. DoS: Denial of Service.

Authors (Year)	Focus of the paper	Type	Domain	Strategy	Shortcomings
Sattar and Matrawy (2019)	Proactive mitigation of DDoS using network slicing allocation and isolation.	DoS	5G	Efficient intra and inter slice allocation to reduce DDoS attacks effects.	The allocation technique is static, if a DDoS targets a currently overloaded slice there is no mechanism to reallocate slices.
Thantharate et al. (2020)	To mitigate DDoS initiation attacks by making sure user equipment can access network slices only after being authenticated and/or authorized.	DoS	5G	The Secure5G model analyzes overall traffic patterns to predict future traffic and allocate resources accordingly.	Experimental evaluation uses a limited data set containing only 65,000 entries.
Bonfim et al. (2020)	FrameRTP4 aims at detecting and mitigating DDoS attacks in real-time for the upcoming 5G mobile technology.	DoS	5G	The framework uses ML techniques for attack detection and ACL wildcard rule generation. 4P is used to exchange information between control and data planes. Wildcards are used for faster pattern matching on the data plane.	Authors fails to provide clear mechanisms on how network slicing is used to detect and mitigate DDoS attacks.

that allows the provision of different quality of service and quality of experience to users by running network configurations according to the type of user and/or traffic requirements. NS are executed as VMs and are deployed dynamically when needed which allows the installations of an IDS or IPS inside VNFs to detect DDoS attacks. So, since NS allows to isolate DDoS attacks in a particular network segment and allows to modify the resources of any slide dynamically NS could help to mitigate an attack and keep safe the underlying SDN infrastructure and then providing or improving security in SDN environments.

Sattar and Matrawy (2019) studied inter and intra-slice isolation effectiveness as countermeasures to DDoS attacks in 5G networks that rely on an SDN infrastructure. Inter-slice isolation does not share hardware resources among slides while intra-slice isolation manages better availability for each slice because the components of the slice are hosted on different hosts, minimizing the degradation of the service produced by a DDoS attack over any slide. Thantharate et al. (2020) proposed a framework for detection and mitigation of DDoS attacks against 5G using network slicing and deep learning techniques to allocate and configure network slices according to types of traffic/attack and proactively detect and eliminate threats based on incoming connections before they infest the 5G core network. Bonfim et al. (2020) proposed a P4-based framework called FrameRTP4, that delivers real-time detection of DDoS attacks against 5G by using an ACL to detect well known attacks and DL techniques to detect unknown attacks through an Orchestrator that controls several controllers and manage the life-cycle of NS to provide security for end-to-end NS.

NS is not a technology natively designed to provide security to the SDN controllers nor to the SDN infrastructure but it is an important emerging technology that can be used to improve the security of the networks segments that rely on an SDN infrastructure avoiding DDoS attacks and ensuring the availability of all the components and devices of the SDN network, and hence providing and improving its security. Table 14 shows a summary of network slicing-based strategies.

4.3.4. Moving target defense strategies

Theoretical foundations for MTD systems are discussed by Zhuang et al. (2014). The ultimate goal of such system is to eliminate or to obsolete the attacker's advantage of time in the reconnaissance phase. The attack surface (Manadhata and Wing, 2011) indicates which are the exploitable resources on the network, therefore another objective of MTD is to reduce the attack surface either by strengthening or removing existing configurations. MTD wants to move network configurations over time whilst transforming them to make sure past configurations are not usable for attacks. Authors define the MTD entropy hypothesis, which states that the greater entropy of the configuration of a MTD system, the more effective the MTD system is. This is important since current work on MTD-based approaches to DDoS detection and mitigation use MTD entropy as indicator to evaluate performance.

Steinberger et al. (2018) introduce a DDoS defense solution using MTD for high-speed SDNs and Software-Defined Exchange (SDX) (Gupta et al., 2014) for collaboration efforts between ISPs in order to investigate the effectiveness of MTD approaches in the context of ISPs. The solution performs network-level MTD by using multiple on-demand routers that shape the topology of the network, and host-level MTD by making use of IP-hopping by setting up a honeypot. Evaluation setup involves Mininet and ONOS for virtualization and controller respectively. Five hosts are deployed inside the network, each representing an ISP. Also, each hosts implements SDX in order to collaborate between them and share security events. A flooding attack is performed aimed at one of the collaborating hosts. Its detection engine identifies malicious network traffic, initiating an MTD strategy to change the attack surface and to inform the other hosts about the attack. After receiving the information, collaborative hosts also change their network configuration. Through experimental evaluations, authors show the effectiveness of MTD to mitigate DDoS attacks. Moreover, analytical models show that when the number of collaborative hosts increases, the probability of success for DDoS attacks decreases.

Luo et al. (2019) propose a hybrid approach to DDoS attack mitigation by combining MTD and honeypots for IoT environments. The MTD strategy is to constantly use different IP addresses on IoT devices and servers at random rates for harder reconnaissance information gathering. Honeypots are virtually deployed inside the network, imitating IoT devices for detection and mitigation purposes. Based on the Mirai botnet attack (Kolias et al., 2017), the honeypot detects when Telnet and SSH connections are performed by the same IP addresses and mitigates by installing drop rules on the switches. Experimental evaluation uses Ryu as the SDN controller and Open vSwitch in a Mininet network. Mirai malware is implemented in 20 IoT devices out of 100 in the network. Results show hybrid MTD and honeypot strategies effectively detecting and mitigating port scanning and SYN flooding attacks.

Narantuya et al. (2019) describe how multiple controllers can be used to provide multiplexing capabilities to MTD strategies for large-scale networks. Authors propose shuffling short-lived, random virtual IP addresses assigned to the hosts in order to hide real IP addresses from attackers. Evaluation setup consists of a set of ONOS controllers, Open vSwitches (OVS), and a total of 256 hosts to simulate a large network. Results show that the probability of successful attacks decrements the more controllers are being used in the network. However, the paper does not address performance evaluations with regards of the shuffling cost since a large number of virtual IPs are being processed and assigned, nor the latency for service provision.

Aydeger et al. (2018) argue that existing MTD approaches do not obfuscate network attributes and operate on real infrastructure, limiting the possible configurations determined by the MTD strategy. Consider that route mutation is employed, information about the network can be obtained since attackers can map the routes every time MTD changes the routes. Moreover, if MTD is used in real infrastructure, node and path configurations to mute become limited. Therefore, authors propose the use of NFV to increase network topology diversity to bring

Table 15
MTD-based detection and mitigation strategies. DoS: Denial of Service.

Authors	Focus(es)	Type	Domain(s)	Strategy(ies)	Limitation(s)
Steinberger et al. (2018)	A collaborative DDoS defense using MTD aimed at high-speed networks for limiting attackers knowledge and gathering insights into the current threat landscape.	DoS	SDN	A set of ISPs collaborate through MTD and SDX to share information. When an ISP is under attack, a detection engine identifies malicious traffic, initiates an MTD strategy and informs other ISPs through SDX the characteristics of the attack. Participating hosts when receiving this information change their network configuration accordingly.	The set of network configurations, the when and where to move are not discussed in the paper, which are important parameters of any MTD approach.
Luo et al. (2019)	An MTD strategy and honeypot architecture to hide network resources and mitigate DDoS attacks in IoT environments.	DoS	SDN	The MTD strategy constantly changes IP addresses of IoT devices. A honeypot is deployed to detect port scanning and to gather information in order to install proper drop rules through the controller.	The MTD entropy of the system is not measured. The rate of moving configurations is not mentioned.

Table 16
(Cont.) MTD-based detection and mitigation strategies. DoS: Denial of Service.

Authors	Focus(es)	Type	Domain(s)	Strategy(ies)	Limitation(s)
Narantuya et al. (2019)	A multi-controller MTD strategy to assign virtual IP addresses to hosts in large-scale networks.	DoS	SDN	Multiple controllers collaborate to assign shuffled virtual IP addresses to hide real IPs from hosts. The more controllers there are in the network, the more the probability of attack success decreases.	No performance evaluation on shuffling costs nor latency for service provision. No clear communication mechanism for inter-controller communication is described.
Aydeger et al. (2018)	MTD and NFV-based strategy for link flooding attacks moving network path configurations.	DoS	SDN	A randomly chosen strategy hides real paths by using route mutation (select alternate paths), overlay networks to deploy additional virtual nodes and route packets through virtual paths, and mirror networks with fully virtualized nodes and paths.	Additional network configurations to discover destination hosts are required depending on the selected strategy.

more options for MTD strategies. Authors use two pre-defined parameters to trigger MTD: a timeout and a threshold on a number of traceroute requests received. The following strategies are randomly selected after triggering: (i) direct route mutation strategy which periodically changes paths of traceroute requests; (ii) overlay network strategy which adds more routes to an existing network by deploying NFV nodes; and (iii) mirror network strategy which mirrors and mimics a real segment of the network with virtualized hosts and switches. Depending on the strategy, additional NFV operations and configurations may be needed (e.g., configuring NAT). Experimental results show that MTD can in fact be used to detect and mitigate link flooding attacks in SDNs. However, since the strategy is chosen randomly, additional network configurations are needed. One solution to this is to select an strategy based on the current network configurations.

Summary and critical analysis. Moving Target Defense strategies are a relatively new approach for detection and mitigation of DDoS attacks. The programmable nature of SDN enables MTD strategies to be fully deployed and dynamically configured on production networks. Steinberger et al. (2018), Luo et al. (2019), and Narantuya et al. (2019) use MTD strategies based on virtual IP addresses to hide real IP addresses. Aydeger et al. (2018) propose the use of multiple path strategies to hide network topology. Tables 15 and 16 show a summary of MTD-based strategies reviewed. More information about MTD not specifically related to SDN can be found in Sengupta et al. (2020) and Sengupta et al. (2019).

There are several drawbacks associated with using MTD on SDN with the specific purpose of detecting and mitigating DDoS attacks. As we described earlier, most strategies are based on IP randomization, which relies on changing the IP address of servers to make them harder to locate. The first challenge is address spaces: changing IP addresses requires a large number of unallocated IP address to ensure randomness in new allocations. This is a hard requirement to meet when using IPv4 due to the limited addresses available. Researchers have proposed changing combinations of IP address and ports (Kandoi and Antikainen, 2015) and even virtual IP addresses (Wei and Fung, 2015). A more feasible solution is to use IPv6 inside the network. However, the self-configuration mechanism of IPv6 where a node can impose its own address may cause problems with the MTD strategy. The second challenge is connection disruption: if a host is forced to change its IP address when a connection between two hosts is open, the connection may be closed, and the service disrupted. Connections may be monitored to avoid changes when on-going connections are happening.

Another drawback comes from the use of VMs in MTD strategies. The time it takes to start or reload a VM with specific configurations can affect the performance of the network. Zheng and Namin (2019) argue that more research is needed in optimizing virtualization processes. Authors also state that more research efforts are required in studying the effectiveness of SDN-based MTD strategies compared to conventional MTD strategies and in the compatibility of SDN-based MTD strategies to ensure that any strategy will be feasible to implement with the current network infrastructure.

5. Discussion

In this section, we discuss existing surveys and explain how our work differs from these surveys — see also Tables 17 and 18.

5.1. Existing surveys on SDN security

Ahmad et al. (2015) categorized security analysis and security solutions designed for SDN planes. In their work, they focused on authentication and authorization, flow rule insertion, access control and accountability, DoS attacks, and scalability and availability. A number of security solutions for the application plane were identified, which include FRESCO (Shin et al., 2013a) (which provides a framework for developers to implement solutions that comply with network security policies), PermOF (Wen et al., 2013) (which facilitates controlled access to OpenFlow controllers), and Assertion (Beckett et al., 2014) (which checks for flow rule contradictions). In the control plane, most solutions focus on controller scalability and availability. For example, DDoSDetection (Braga et al., 2010) is a detection framework for **DDoS attacks** by using a flow collector, a feature extractor, and a classifier. Solutions in data plane generally focus on flow rules and access control and availability. The authors also identified a number of challenges, namely: the need for standardized programming models and development environments to minimize conflicting modules; the need for class-level application security to ensure high-level security policies in applications; the importance of ensuring scalability and availability in

Table 17
Summary of DDoS Detection and Mitigation Surveys in SDN.

Survey title	Year	Domain	Focuses
SDN Security: A Survey (Scott-Hayward et al., 2013)	2013	SDN	SDN security risks (Table 1).
Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges (Yan et al., 2016)	2015	Cloud	Network layers of SDN Cloud computing.
A Survey of Security in Software Defined Networks (Scott-Hayward et al., 2016)	2016	SDN	SDN security risks (Table 1).
A Survey on Software-Defined Networking Security (Bian et al., 2016)	2016	SDN	Security solutions against DDoS threats, policy enforcement, authentication, and controller trust.
A Survey on the Security of Stateful SDN Data Planes (Dargahi et al., 2017)	2017	SDN	Stateful SDN data planes vulnerabilities.
Software-Defined Network (SDN) Data Plane Security: Issues, Solutions and Future Directions (Shaghaghi et al., 2018)	2018	SDN	SDN data plane security issues.
A Survey on Emerging SDN and NFV Security Mechanisms for IoT Systems (Farris et al., 2019)	2019	IoT	SDN security issues with NFV-based solutions in IoT environments.
Software-defined Networking-based DDoS Defense Mechanisms (Swami et al., 2019b)	2019	SDN	Taxonomy of defense mechanisms against DDoS attacks.
A Survey: Security Threats and Countermeasures in Software Defined Networking (Mubarakali and Alqahtani, 2019)	2019	SDN	Security against MitM attacks, memory depletion, DDoS attacks, and malicious applications.
A Survey: Typical Security Issues of Software-Defined Networking (Liu et al., 2019)	2019	SDN	Security defense techniques classified by SDN layers.
A Survey of Machine Learning Techniques Applied to Software Defined Networking (SDN): Research Issues and Challenges (Xie et al., 2019)	2019	SDN	Machine Learning techniques for traffic classification, routing optimization, QoS/QoE prediction, resource management, and security.

Table 18
(Cont.) Summary of DDoS Detection and Mitigation Surveys in SDN.

Survey title	Year	Domain	Focuses
Survey on SDN based Network Intrusion Detection System using Machine Learning Approaches (Sultana et al., 2019)	2019	IDS	Machine learning techniques for IDS on SDN.
A Survey of Networking Applications Applying the Software Defined Networking Concept Based on Machine Learning (Zhao et al., 2019)	2019	SDN	Machine Learning techniques for resource management, route planning, traffic scheduling, fault diagnosis, and security.
This Survey	2020	SDN	Detection and mitigation strategies for DDoS attacks based on statistical, ML, SDN architecture, blockchain, honeynet, NFV, MTD, and network slicing approaches.

order to overcome bottlenecks; the importance of intelligence trade-off between control and data plane to minimize switch dependency on the controller; the need for security and forwarding policies for secure forwarding mechanisms; the need for network security automation; and the importance of host identity to provide a persistent identity. However, emerging technologies such as blockchain, NFV, Honeynets, MTDs, and network slicing are not discussed in the survey (Ahmad et al., 2015).

Yan et al. (2016) studied **DDoS attacks** in cloud computing environments, as well as SDN-based mitigation approaches. While SDN-based approaches can potentially achieve higher quality of service (QoS), virtual machine orchestration, and security, there are a number of limitations, such as those summarized in Table 1. The authors also provided a classification of DDoS attacks at the different logical protocol layers (i.e., application, transport, Internet, and network interface). However, no taxonomy on mitigation techniques is presented. The authors observed that existing solutions for application-level attacks do not provide optimal trade-off between performance and security, and highlighted the need to extend traffic intelligence from network to application layers.

Scott-Hayward et al. (2016, 2013) also surveyed potential security risks (e.g., unauthorized access, data leakage, data modification, malicious/compromised applications, **DoS**, and wrong/faulty configuration) at the different SDN layers and interfaces (i.e., application, control, data, application-control interface, and control-data interface). They then recommended that controller solutions should provide a policy conflict resolution subsystem to avoid unauthorized access, and irregularities, as well as mutual authentication to prevent data manipulation attacks, component impersonation and secure identification. They also highlighted the importance of network slicing for control plane isolation; containerized applications to enable application's access rights on infrastructure, and limit resource usage per application; rate-limiting, flow aggregation, and short timeouts to ensure correct packet forwarding behavior; and logging in monitoring applications for troubleshooting and debugging the infrastructure.

Dargahi et al. (2017) focused on security implications associated with the data plane's programmable nature (e.g., vulnerabilities associated with stateful in-switch processing). Stateful operations in SDN reduce the switch-to-controller signaling since the switch can avoid relying on the controller for any possible update in the forwarding state. The security issues studied in stateful data planes are as follows: (a) unbounded flow state memory allocations, where an attacker can take advantage of the in-memory space that each switch requires to store flow state information; (b) CPU intensive operations, where an attacker forces the execution of intensive operations on a switch; (c) authentication mechanisms in data plane, where an attacker can impersonate a switch and inject fake event/packet information; and (d) a lack of central data plane state management, where state inconsistency can cause network misbehavior. The authors only considered **DoS attacks** in memory saturation and CPU exhaustion attacks. Our survey, on the other hand, has a broader coverage (i.e., DDoS attack detection and mitigation on several planes).

Farris et al. (2019) focused on SDN- and NFV-based security mechanisms for IoT systems. Security threats in IoT environments differ from traditional because IoT devices are potentially more vulnerable to hardware hijacking, replication attacks, tampering attacks, battery draining attacks, among others. Moreover, IoT applications are also prone to malicious application attacks, data leakage, phishing, and malicious scripts. Also, compared to traditional IoT solutions, scalability

can be achieved by the use of SDN and NFV. The authors focused on **DoS attacks** in IoT environments such as Ping of Death, TearDrop, UDP/SYN flood, and SYN flood. They also identified a number of research opportunities, for example those relating to definition of IoT security policies, orchestration of SDN/NFV-based security solutions over heterogeneous IoT environments, optimal selection of SDN and NFV frameworks, and granularity of protection required by the network. Our survey goes further by including emerging technologies such as blockchain, MTD, Honeynets and network solicing not only for IoT environments, but also for conventional and 5G scenarios, where SDN is an enabling technology.

Shaghaghi et al. (2018) studied the challenges in securing the data plane of SDN, and presented different attacks against SDNs based on their scope and impacts. The survey discussed **DoS attacks** in the application plane where next generation firewalls can be used to prevent such attacks, and in the southbound API where flooding the channel limits its availability. The authors also provided one taxonomy of attacks for each of three main categories: implementation attacks, enforcement attacks and policy attacks, as well as identifying potential attacks scenarios and theirs attack vectors considering the vulnerabilities related to access control and those affecting the availability as the most severe. They also categorized current SDN security solutions into those that protect SDN's core features and those that protect SDN's implementations and SDN-based security services. According to the authors' evaluation, the least explored areas in SDN is solutions that secure the data plane. Finally, they established a set of requirements for securing the data plane of SDN, and suggested that a working solution should include a scanning methodology, distinguish malicious actions, locate malicious forwarding devices, intelligent response to threads and support stateful data plane. They observed that existing solutions to secure the data plane suffer from inaccurate adversarial representation, and pointed out only two solutions (i.e., SPHINX (Dhawan et al., 2015) and WedgeTail (Shaghaghi et al., 2017)) satisfied the requirements they studied. While they identified a small number of potential solutions, it would be interesting to introduce a proposal considering stateful data planes or machine learning engines that could detect different attacks accurately at the data plane. The survey, however, does not consider different mechanisms to mitigate the various identified attacks. Also, the proposed taxonomy is limited to data plane.

Swami et al. (2019b) categorized existing **DDoS defense mechanisms** into statistical-, machine learning-, and application-based approaches. They also identified features in a SDN-based approach that are vulnerable to DDoS attacks, namely: (a) limited space for flow rules allocation; (b) the controller's single point of failure; (c) the communication protocol between control and data planes; and (d) the dependency of the switches with the controller in order for proper functioning. They emphasized the importance of focusing on security of the switches, since these switches could be easy targets for attackers. Also, a number of security countermeasures require modifications to OpenFlow switches or additional appliances; thus, it is important to minimize the overall network setup cost and provide some level of intelligence to switches to reduce traffic volume to the controller during decision-making. The authors also pointed out that slow DDoS attacks in SDN are relatively understudied, the lack of standard communication protocols between applications on the Northbound interface, the lack of a east–west bound interface for controller-to-controller communication to provide seamless scalability and interoperability, and the need for an efficient network traffic analysis tool (since current tools must be adapted to perform on SDN flow-based analysis). While NFV, Blockchain, IoT, and Honeynet are briefly mentioned in the survey, solutions based on these technologies are not included in the discussion.

Bian et al. (2016) reviewed existing **DDoS threats and solutions** in SDN, and identified a number of open challenges (i.e., lack of a comprehensive SDN security framework that can resists several attacks, weakness of the controller, and trustworthiness of the controllers). Similarly, Mubarakali and Alqahtani (2019) studied SDN security threats and their countermeasures, but their coverage is broader as it includes man-in-the-middle (MitM) attacks, memory buffer depletion, DDoS attacks on the controller, and malicious applications. They noted that TLS is needed for secure communication inside the network to prevent MitM attacks. In our survey, we highlight that an alternative to the slow adoption of TLS in SDN is the use of blockchain as a secure and verifiable approach to inter and intra-domain information exchange.

Liu et al. (2019) studied existing security defense technologies based on the application, control, and data layers of SDN. They suggested that future research focuses should include **DoS attack** detection on the controller, supporting scalability and cross-domain communication between controllers, ensuring robust authentication mechanisms for applications, the standardization of a Northbound interface, and ensuring overall global network security.

5.2. Existing surveys on machine learning-based approaches

The use of machine learning to mitigate security risks, specifically DoS and DDoS attacks, in SDN environments is also becoming more popular. For example, Xie et al. (2019) surveyed existing machine learning approaches such as those involving traffic classification, routing optimization, QoS/quality of experience prediction, resources management, and security. The authors also observed that the KDD99 (Tavallaee et al., 2009) and NSL-KDD (Revathi and Malathi, 2013) datasets are widely used in IDS research and that NSL-KDD is better to use when simulating performance of ML-based intrusion detection methods. Deep learning algorithms are also widely used as feature extraction method, due to their strong feature representation capability (Fernández Maimó et al., 2018). The performance on supervised learning algorithms depends on the training datasets and these algorithms have higher detection accuracy than other machine learning algorithms such as Bayes' theory, SVM, and decision trees (Tang et al., 2016). The authors noted the importance of having high-quality labeled datasets that can be used for model training.

Other surveys focusing on IDS and machine learning for DoS include those of Nguyen (2018) and Sultana et al. (2019). In the latter, a basic classification of ML techniques and their application in IDS were presented. Machine learning is commonly applied to IDS to improve detection accuracy (García-Teodoro et al., 2009) and to provide a low false alarm rate (Mehdi et al., 2011). Deep learning techniques are also becoming popular in IDS because they provide better accuracy and can be easily implemented in SDN due to its programmable nature. The main objective of an IDS is to monitor flows mainly for unauthorized access, collecting data about possible threats (Hodo et al., 2017). Challenges in adopting machine learning in SDN include the challenge in choosing appropriate features to improve IDS (Nguyen et al., 2010; Wang and Jones, 2017a), inaccurate datasets available, and the challenge in optimizing controller performance for large network deployments (Yan et al., 2016).

Zhao et al. (2019) surveyed the application of machine learning in SDN for areas such as network resource management, route planning, traffic scheduling, fault diagnosis, and network security. Unlike the other machine learning-based surveys discussed so far, the authors included reinforcement learning and the integration of supervised and unsupervised learning work in SDN. They also highlighted the lack of high-quality datasets, as existing datasets can rarely be used directly (most need a format transformation in order to be used). In relation to network security, IDS, **DoS attack detection**, and network fault diagnosis solutions were explored. Most IDS solutions rely on SVM and deep learning (Liu et al., 2016; Meti et al., 2017; Boero et al., 2017; Tang et al., 2016). The authors observed that fault diagnosis in DoS attacks is an understudied area.

5.3. Significance of our survey

As discussed earlier, this survey focuses on emerging detection and mitigation strategies for DDoS attacks in SDN. While DoS/DDoS attacks have been extensively studied, they remain a major concern in SDN networks. Also, other security risks (unauthorized access, data leakage, data modification, malicious applications, and configuration issues) are directly related in most of the cases, with poor network management. This is, perhaps, why DoS and DDoS attacks are studied in several surveys, such as those outlined in Table 1. From the table, one can observe that well-known mitigation strategies include IDS/IPS with statistical or machine learning-based techniques, enhanced software components (e.g., securing communication channels between SDN elements, controlling idle buffers to avoid overflows, providing better configuration), and dedicated hardware components (e.g., to ensure better processing capabilities on SDN switches).

In this survey, we focus on newer emerging technologies that can potentially be used to facilitate the detection and mitigation of DoS attacks, for example using blockchain to secure channel communication and minimize data tampering, NFV to dynamically deploy isolated environments with different VNFs or to change flow rules in switches to deploy an IPS in runtime when the controller is under attack, honeynets to deceive attackers and create loops for packets to be dismissed, NS for isolated environments in next generation mobile communications, and MTD to dynamically change network configuration and prevent attackers from gathering reconnaissance information to launch attacks.

6. Challenges, strategies and research directions

Now that we have discussed several DDoS attack mitigation approaches, we will introduce the challenges and limitations in existing DDoS detection and mitigation strategies. We will also discuss potential research opportunities.

6.1. Application plane

One observation is the lack of standards and open specifications for application-level access rights. For example, applications that require access to networking services and functions provided by the control plane are not properly handled, which may result in abnormal network behavior and manipulation of SDN resources, configurations, and performance (Wen et al., 2013). Therefore, proper access control and accountability mechanisms must be further specified and developed to ensure expected network behavior and performance. There is also a certain independence in the way applications are developed. No standards nor open specifications that facilitate APIs for application to control network services are defined. One alternative to overcome this shortcoming is for SDN to provide class-based application security, where applications are categorized into classes or groups enforcing security procedures based on pre-defined high-level security policies. This reinforces the importance of having a uniform security framework in developing such applications to provide boundaries of what the security policies can access and manipulate.

When several applications change the state of networking devices, e.g., changing flow rules due to the detection of an attack, the network can become unstable if such states contradict each other. Therefore, the development of a subsystem for policy conflict resolution is a must when large-scale networks are managed by several stakeholders (e.g., operational technology teams and cyber security teams). Such subsystem should identify wrongly specified configurations, unauthorized access, and irregularities. For example, Basile et al. (2015) proposed a policy manager framework to define security policies, and Montero et al. (2015) proposed a trusted virtual domain for authentication, security policy management, and application security. OpenNF (Gember-Jacobson et al., 2014) provides efficient and safe allocation of data flows across network functions instances in NFV that can be easily extended to SDN.

Future directions on application plane security can be summarized as follows:

- Provide standards and specifications of APIs to applications that require access to network services to limit its capabilities inside the network.
- Development of opinionated security frameworks for application development.
- Mutual authentication support for SDN components and applications. SDN can greatly benefit from the implementation of blockchain technology to overcome such challenges and provide strong integrity in data sharing.
- More efficient IDS/IPS to avoid sending new flows from switches to the controller to avoid resource exhaustion. Incremental deployment is a viable solution (Agarwal et al., 2013).
- Logging critical information inside IDS/IPS can greatly benefit network performance since troubleshooting and debugging the infrastructure become clearer.

6.2. Control plane

By increasing the number of SDN switches managed by a single controller, the controller's response time for computing and installing flow rules will also increase correspondingly (Tootoonchian et al., 2012). For example, the authors in Jarschel et al. (2011) argued that OpenFlow is not capable of handling large number of new flows using 10 Gbps links. This limitation also implies a research challenge and opportunity, that is network controllers, especially single controllers, are prone to DDoS attacks since they represent a single point of failure for the entire network. An attacker can exhaust the controller's CPU and bandwidth if several switches send requests for processing packets not present in the switch's flow table. Hence, would it be possible to design solutions that provide SDN switches some sort of intelligence to decrease the load on the controller when processing new flows?

Future directions on control plane security include:

- Studying the potential of giving switches greater intelligence to provide them with local decision making capabilities. The Programming Protocol-independent Packet Processors (P4) (Foundation, 2020), for example, is a language for expressing how packets are processes. Hence, P4 can potentially be able to provide certain intelligence to switches to avoid bandwidth exhaustion between switches and controllers.
- Designing of secure new generation architectures to ensure high availability.
- Designing of load balancing mechanisms to facilitate multiple controllers in sharing load among OpenFlow switches and avoiding resource exhaustion.

6.3. Data plane

One challenge associated with the data plane is the limited size for flow entries a switch can maintain. For example, the authors in Lin et al. (2017) proposed using idle memory available on switches to maintain large volume of flow rules. This, however, requires secure and efficient mechanisms to share flow entries between switches.

Adapting to changes and updating the network can be a challenge in SDN. Network security automation can potentially allow one to adapt to changing and updating itself with automation techniques.

Hence, a future research direction is the detection and potential mitigation entirely at the data plane or, at least, offloading part of load from the control plane to the data plane. One first approach in that direction is ORACLE (Macías et al., 2020).

6.4. Statistical

Statistical strategies have been shown to be effective techniques, for example in traffic classification to facilitate detection. Such strategies are easier to implement and do not require significant processing or memory requirements. Moreover, the OpenFlow specification mandates SDN switches to include statistical counters for each matching flow entering the device.

As mentioned earlier, statistical approaches tend to perform very well when the entropy of the system is low. One challenge associated with such approaches is dealing with new types of threats (e.g., those involving zero-day exploits/vulnerabilities). Moreover, the time period required by these strategies relates directly to the amount of traffic to be processed and the target accuracy of the strategy. Furthermore, to be able to provide a wide range of traffic detection, statistical strategies need a large number of mathematical models and thus, even larger number of network traffic to be processed.

6.5. Machine learning

Machine learning-based SDN attack detection and mitigation approaches are by far the most popular, although use of ML techniques brings specific challenges, such as the lack of high-quality and standardized datasets (Mestres et al., 2017; Xie et al., 2019; Bakker et al., 2018). Existing datasets generally suffer from limitations non-representative data, lack of labeled data, class imbalances, missing attributes, lack of validation, etc. Yu et al. (2018) and Hammerschmidt et al. (2017). Other challenges include lack of proper feature selection methods for current datasets, and how to determine the optimum number of model parameters (Wang and Jones, 2017b). Machine learning approaches use synthetic datasets for intrusion detection due to the lack of better and more realistic datasets. Hence, it may be necessary to mix more than one technique in order to have a robust mitigation technique that can be deployed in a production environment.

Additional challenges are presented when the trained models are tested in real or simulated environments, it is observed that the very high accuracies gotten by the models with the datasets are not obtained in real environments. These differences are presented most of the times because the attacks in the real or simulated environment are not executed with the same tools that were used to create the dataset, so this problem opens a research issue to solve for future investigations.

Existing machine learning-based approaches (i.e., those that are not transfer learning-based) also use frameworks not developed for SDN, and hence they may not work well for flow-based data in SDN. In other words, the conversion/adaption of the data may introduce integrity and validation issues.

In summary, key challenges include the need to generate high-quality, standardized, well documented and representative SDN-based datasets, that includes attacks performed with different tools per attack in order to validate the models in real or simulated environments getting the same accuracy than in the testing phase with the dataset. Finally the design of machine learning frameworks for SDN is needed since data in such networks (flow-based) differ from traditional networks (packet-based).

6.6. Blockchain

One of the key features of SDN is to provide a complete topology view of the network. Such information allows decision-making, but topology discovery is vulnerable to link fabrication attacks. In the latter, an attacker injects LLDP packets to create invalid network topologies (Alharbi et al., 2015). The use of blockchain on SDN can potentially mitigate such an attack, since it can provide authentication for LLDP packets (Alharbi, 2020). Blockchain can also help with the distribution of authenticated packets traversing through the SDN when using virtualized nodes. This allows to isolate VM traffic and to provide better management for such devices.

Ensuring security and integrity of blocks in a large-scale blockchain implementation can be achieved, as long as the attacker does not have mining power that exceeds that of the network. In other words, blockchain can be implemented in large-scale projects (Christidis and Devetsikiotis, 2016). The authors in Memon et al. (2018) proposed a *block maturity level* (BML), where mining blocks change from "pending" to "approved" once 6–8 new blocks have been prematurely mined. This ensures that there is minimal risk included in the validation of a particular blockchain.

In a physically distributed multi-controller SDN architecture, blockchain data structure should be used to allow secure and efficient communication between nodes in order to maintain coherent and synchronized network states (Yang et al., 2019).

Blockchain requires Proof of Work (PoW) to execute a transfer. When the network is large enough and the number of transactions are several per second, the time associated with the PoW computation is usually larger. In the case of Bitcoin, 7 transactions per second are allowed, but the PoW takes up to 10 minutes to complete. This leads to inefficient transaction times (Haque and Rahman, 2020). Moreover, many related work on detection and mitigation of DDoS using blockchains are based on Smart Contracts (SC). Previous research work describe many vulnerabilities related to SC (Khan and Namin, 2020). More effective security testing criteria should be developed for SC. ML and DL algorithms can also be used to detect and mitigate such vulnerabilities.

6.7. Network function virtualization

The compute domain, hypervisor domain, and network domain constitute an NFV infrastructure (NFVI). Such an infrastructure can be vulnerable to internal threats (e.g., inappropriate operations) as well as human errors. The risk can be minimized by following strict operational procedures (Yang and Fung, 2016). For example, devices inside an NFVI should have security certification processes to minimize the risk, and a NFV framework should be deployed to provide some standard security mechanisms for authentication, authorization, encryption, and validation.

The use of NFV inside the SDN can help to instantiate VMs as needed. For example, NFV allows the creation of temporary networks which can be managed using VNFs such as DHCP, firewall, and DNS. When the network is no longer required, such functions can be removed. However, this highly dynamic behavior of NFV can also result in human errors when creating virtualized interconnection of virtual devices. A possible solution to such challenge is a subsystem for network topology validation and implementation failure (Lal et al., 2017).

VNFs can be compromised. In such scenarios, the compromised VNFs can generate high traffic volume and consequently exhausting CPU, hard disk, and memory resources. In an attempt to mitigate such an attack, the NFVI orchestrator may deploy additional virtual services to overcome traffic load, which can lead to the exhaustion of hardware resources. So, it is important to define a set of metrics to help to evaluate the resource's consumption, like CPU and RAM, so that the NFVI orchestrator can ensure the health of the entire architecture.

The aforementioned challenges are inherent to the NFV architecture. However as we mentioned in Section 4.3.1 there are several challenges that have not been tackled in the cooperation of SDN and NFV to detect and mitigate security attacks.

- To the best of our knowledge, security solutions using the cooperation of SDN and NFV have not been evaluated in realistic scenarios using all the NFV MANO framework integrated with robust SDN controllers.

- Existing proposals do not differentiate where (and how) in a NFV–SDN enabled architecture, is more suitable to perform DDoS detection and mitigation.
- While SDN monitoring is a good way to obtain network information in order to detect possible attacks, NFV can also make use of SDN to steer the mitigation VNFs in an efficient way. Up to now, the use of SDN to enable NFV mitigation strategy has not been sufficiently studied.

In summary, future research should include detection of compromised VNFs, defending against DDoS attacks using NFV, and supporting trust management for multiple vendors and deployments. A trusted platform module (TPM) can be used to validate the platform's firmware, BIOS, bootloader, operating system (OS) kernel, and other systems components. Also, the separation and management of VM traffic prevents VMs from resource depletion and keeps isolated environments. NFV can also be used to automate network configuration, provision, and management. Also, regarding NFV and SDN cooperation to react against DDoS attacks, it is important to perform evaluation in realistic conditions, even emulated environments using tools such as OSM or OPNFV for NFV and mininet with ONOS for SDN are possible. Novel detection strategies can be proposed making use of programmable data planes (Foundation, 2020) where, based on entropy, programmable switches could perform an early detection (Lapolli et al., 2019) that can be later confirmed by deploying a virtual IDS close to that switch. Also, mitigation strategies could include VNFs to quarantine traffic that can be deployed and interconnected using SDN, also SDN rules could be installed to divert or reject anomalous traffic.

6.8. Honeynet

While honeynets can help cyber defenders to collect and share information about attacks (e.g., attacker's tactics, techniques and procedures — TTPs), there are also a number of challenges associated with Honeynet deployments. First, Honeynet requires carefully planned deployments since they introduce network overhead. Secure communication between Honeypots can also be challenging in practice, and existing solutions using Honeynets do not consider scenarios in where attackers have compromised VNFs or applications that may sniff traffic and change flow rules. Hence, one could also consider integrating the use of blockchain in honeynet-based approaches. In addition, proper redirection and isolation of attack traffic should be included in future research.

From our literature review, we did not find a recommendation on which protocols/applications should be supported in a Honeynet for modern network deployments. Honeypots such as Honeyd, HoneydV6, Conpot, CryPLH, SHaPe, and CockpitCI support only HTTP/HTTPS, TFTP/FTP, and SNMP. Honeyd is the only honeypot that supports Telnet/SSH. HoneydV6 is the only honeypot that supports IPv6. Hence, future directions should also focus on the impact of supporting different protocols and performance evaluations on implementing Honeynets on virtual and real hardware. The definition of a robust architectural model for Honeynets for modern networks is also an opportunity area for researchers since third-generation Honeynets are still being adapted to support virtual deployments. Frameworks such as Serbanescu et al. (2015), Kyung et al. (2017) and Han et al. (2016) are being defined to support multiple protocols/applications and native support for virtual deployments.

6.9. Network slicing

Network slicing is an integral part of next-generation mobile communications, as it enables the partition of physical network resources into different classes to meet different QoS/QoE requirements whilst isolating network functions from each other. This also facilitates the management and optimization of network resources at macro levels (Rost et al., 2017).

Khan et al. (2020a) identified a number of security implications in network slicing, namely: disrupting the communication between slices can prevent proper management (attack on inter-network slice communication); different slices may have different security policies and protocols, and therefore an attacker can be granted access to the NS system through less secure slices; DoS attacks are also present in NS either by exploiting the virtualization platform or the physical resources from the network, and consequently result in degrading the performance on other slices and on the overall networking infrastructure; if an attacker gains access through less secure slices, it can attack slices that share the same primary hardware (side-channel attack); and misconfigurations on hypervisors may lead to backdoor attacks.

A number of researchers (Khan et al., 2020a; Barakabitze et al., 2020; Li et al., 2017; Khan et al., 2020b) echoed the need for a set of consistent policies, as well as the need for appropriate mechanisms for controlling inter-network slicing communication and signaling to be carefully defined for proper use and deployment of network slicing in 5G networks. The call for adaptive security mechanisms in network slicing and isolation in SDN may be answered by using machine learning techniques to perform node and path selection based on strict security requirements of each slice.

Network slicing security challenges become more apparent when it is being implemented in multi-domain infrastructures. Efficient multi-domain policy coordination mechanisms have to be studied and developed to ensure that any compromised slice has little to no impact on other slices sharing the same infrastructure (Ordonez-Lucena et al., 2017). Also, network (slicing) forensics may help with the identification, collection, and examination of evidence after an attack has been launched, being more relevant in multi-domain infrastructures due to the complexity of the system (Khan et al., 2016). Pourvahab and Ekbatanifard (2019a,b) propose a SDN-based forensics architectures based on blockchain technology for Cloud and IoT for multi-controller environments, although can be properly adapted into NS and 5G environments. In this context, blockchain is used for the recollection and preservation of reliable data evidence from the environment.

6.10. Moving target defense

While MTD approaches generally have parameters such as what, when and where to move, most approaches focus on the movement of the exploration and attack surface, and configuring pools of virtual IP addresses and virtual ports. While moving the configuration set obsoletes the reconnaissance stage of an attack, the goal of moving detection surfaces is also to improve scalability and QoS (Sengupta et al., 2019).

From our literature review, we observe that most approaches mostly ignore the timing function. Moreover, existing studies on such timing functions are generally limited to specific threat models and topologies. When protecting against jamming attacks, Algin et al. (2017) proposed a 15-second period to move, while 60 seconds are recommended in network mapping attacks (Sengupta et al., 2019). The when to move is still a challenge research area in MTD strategies, since several input parameters must be taken into account to determine optimal moving periods. Experimental evaluation on MTD involves the use of quantitative and qualitative metrics according to Zhuang et al. (2014). Quantitative metrics are based on system entropy, which can be defined as the amount of configuration, randomness, and strategies a MTD strategy has. However, the model to measure such entropy is defined by each author, and does not follow any standard. For qualitative metrics, Sengupta et al. (2019) explained that existing MTD approaches only focus on security or performance, but not in both. Furthermore, since most MTD strategies are mounted over the NFV infrastructure, these are subjected to the same issues and restrictions of the NFV strategies.

Industry adoption on MTD is rather slow. More than 50% of the research works use simulated networks, $\sim$ 34% use emulated environments and only $\sim$ 13% are implemented at commercial level in

Table 19
Best suited strategies for different SDN scenarios. Complexity is based on the amount of resources (software and hardware) required for the deployment of the strategy. UA: Unauthorized access, DM: Data modification, MA: Malicious applications, CI: Configuration issues, DoS: Denial of Service.

Strategy	Scenario	Complexity	Attack
Statistical	Best suited for lightweight network traffic classification for detection of well-known anomalous flows.	Low	DoS
SDN architecture	Best suited for the protection of overflowing buffers on SDN switches.	Medium	DoS
Machine learning	ML strategies shine in detection and mitigation of new attacks due to its great adaptation and ability to learn. Such strategies are expected to be employed in SDN as a part of its architecture.	Medium	DoS
Blockchain	Currently being employed for reliable and traceable communication between entities in SDN. It poses opportunities for network forensics and lightweight detection since malicious nodes are not recognized as trusted entities inside the blockchain.	Low	UA, DM, DoS
NFV	Coupled with robust decision mechanisms, NFV can deploy different network functions on demand in order to mitigate DDoS attacks. Best suited for the mitigation of a wide-range of attacks. Early attack detection can be performed by using programmable data planes to instantiate virtual IDS close to the switch.	High	MA, CI, DoS
Honeynet	With the help of NFV, provides good ground for detection of reconnaissance attacks and mitigation by deploying virtual networks as sink for malicious traffic.	High	DoS
Network Slicing	Can provide optimal allocation of slices in 5G scenarios and mitigation by configuring and isolating pre-defined network functions.	Very high	UA, CI, DoS
MTD	Allows the movement of attack surface to obsolete reconnaissance information by moving/updating network parameters through NFV and configurations. Ideal for small and medium-scale networks.	Very high	DoS

real production networks (Sengupta et al., 2020). The main reason for the slow adoption of MTD strategies in the negative impact it may cause in terms of Quality of Service (QoS) and its cost-overhead for implementing and maintaining a MTD. Future research directions should study commercial-grade frameworks and best-practices for the development and deployment of MTD strategies.

6.11. Summary

There is no solution that fits all scenarios, nor a panacea for DDoS attack detection and mitigation in SDNs. With the significant increase in the number of DDoS attacks between 2020 and 2021, we can foresee more combinations of the different emerging technologies in order to mitigate the damage of DDoS attacks (Liu et al., 2018; Zarca et al., 2020; Luo et al., 2019). Table 19 summarizes the different solutions and the scenario in which they are more effective, the complexity of implementation/deployment based on the hardware and software resources required for the strategy to work properly, and the type of attack(s) that can be detected and mitigated with each strategy.

7. Conclusions

There are many potential benefits associated with SDN over conventional networks, for example in terms of management and automation due to the former's centralized nature and programmable environment. Moreover, SDN promotes research and innovation since it can be used to also facilitate implementation and performance evaluation of new protocols and techniques that were only available to vendors in conventional networks due to the closed nature of networking devices. There are, however, a number of security challenges we need to consider, such as security issues introduced by SDN's architecture, besides those found in conventional networks and those produced by human error.

This work presents a comprehensive literature review on emerging detection and mitigation strategies for DDoS attacks and some other security issues in SDNs. Each review highlights detection and mitigation mechanisms as well as strengths and weaknesses used. Moreover, we provide a taxonomy based on emerging trends in DDoS defense mechanisms such as the use of statistical models, SDN architecture enhancements, machine learning, blockchain, network function virtualization, honeynet, network slicing, and moving target defense in SDN. This taxonomy includes categories of strategies that, although they were not specifically designed to provide security, they are described because these strategies are being used by researchers to provide or improve security mechanisms in SDN, especially against DDoS attacks. Furthermore, we believe that they can be used as a basis for the development of more robust strategies in different areas where SDN is present. We also provide a table where we point out which security issues or attack can be solved with each specific identification/mitigation strategy, so that it can be helpful as reference for future researchers who must deal with these security problems. We believe that our taxonomy can be useful as a reference in future DDoS attack detection and mitigation strategies for SDN

We also identified and discussed a number of ongoing challenges and research opportunities associated with each DDoS detection and mitigation strategy. For example, DoS/DDoS attacks will likely remain a major concern in future SDN implementations, since the controller provides a single point of failure, and consequently making it a vulnerable target for attackers. Moreover, poor network management and mitigation techniques can lead to network under-performing and unresponsive behavior when the controller experiences high-stress scenarios. Therefore, these are potential research opportunities, which hopefully will be explored by the research and practitioner communities.

Declaration of competing interest

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

Acknowledgments

This research was supported by a joint seed funding award from "The University of Texas at San Antonio (UTSA) and Tecnologico de Monterrey", Mexico, and was also partially supported by the project "Red temática CYTED 519RT0580" funded by the Ibero-American Science and Technology Program CYTED.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mane, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viegas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv:1603.04467.

Abdou, A., van Oorschot, P.C., Wan, T., 2018. Comparative analysis of control plane security of SDN and conventional networks. IEEE Commun. Surv. Tutor. 20 (4), 3542–3559.

Abou El Houda, Z., Hafid, A.S., Khoukhi, L., 2019. Cochain-SC: An intra- and inter-domain ddos mitigation scheme based on blockchain using SDN and smart contract. IEEE Access 7, 98893–98907. http://dx.doi.org/10.1109/ACCESS.2019.2930715.

Adi, E., Baig, Z., Lam, C.P., Hingston, P., 2015. Low-rate denial-of-service attacks against HTTP/2 services. In: 2015 5th International Conference on IT Convergence and Security (ICITCS). pp. 1–5.

Agarwal, S., Kodialam, M., Lakshman, T.V., 2013. Traffic engineering in software defined networks. In: 2013 Proceedings IEEE INFOCOM. pp. 2211–2219. http://dx.doi.org/10.1109/INFCOM.2013.6567024.

Ahmad, I., Namal, S., Ylianttila, M., Gurtov, A., 2015. Security in software defined networks: A survey. IEEE Commun. Surv. Tutor. 17 (4), 2317–2346. http://dx.doi.org/10.1109/COMST.2015.2474118.

Algin, R., Tan, H.O., Akkaya, K., 2017. Mitigating selective jamming attacks in smart meter data collection using moving target defense. In: Proceedings of the 13th ACM Symposium on QoS and Security for Wireless and Mobile Networks. In: Q2SWinet '17, Association for Computing Machinery, New York, NY, USA, pp. 1–8. http://dx.doi.org/10.1145/3132114.3132127.

Alharbi, T., 2020. Deployment of blockchain technology in software defined networks: A survey. IEEE Access 8, 9146–9156.

Alharbi, T., Portmann, M., Pakzad, F., 2015. The (in)security of topology discovery in software defined networks. In: 2015 IEEE 40th Conference on Local Computer Networks (LCN). pp. 502–505. http://dx.doi.org/10.1109/LCN.2015.7366363.

Alsaeedi, M., Mohamad, M.M., Al-Roubaiey, A.A., 2019. Toward adaptive and scalable openflow-SDN flow control: A survey. IEEE Access 7, 107346–107379.

Aydeger, A., Saputro, N., Akkaya, K., 2018. Utilizing NFV for effective moving target defense against link flooding reconnaissance attacks. In: MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM). pp. 946–951.

Azodolmolky, S., Wieder, P., Yahyapour, R., 2013. Performance evaluation of a scalable software-defined networking deployment. In: 2013 Second European Workshop on Software Defined Networks. pp. 68–74.

Bakker, J.N., Ng, B., Seah, W.K.G., 2018. Can machine learning techniques be effectively used in real networks against ddos attacks?. In: 2018 27th International Conference on Computer Communication and Networks (ICCCN). pp. 1–6. http://dx.doi.org/10.1109/ICCCN.2018.8487445.

Bannour, F., Souihi, S., Mellouk, A., 2018. Distributed SDN control: Survey, taxonomy, and challenges. IEEE Commun. Surv. Tutor. 20 (1), 333–354.

Barakabitze, A.A., Ahmad, A., Mijumbi, R., Hines, A., 2020. 5g network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges. Comput. Netw. 167, 106984. http://dx.doi.org/10.1016/j.comnet.2019.106984, URL http://www.sciencedirect.com/science/article/pii/S1389128619304773.

Basile, C., Lioy, A., Pitscheider, C., Valenza, F., Vallini, M., 2015. A novel approach for integrating security policy enforcement with dynamic network virtualization. In: Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft). pp. 1–5. http://dx.doi.org/10.1109/NETSOFT.2015.7116152.

Bavani, K., Ramkumar, M.P., Selvan, G.S.R.E., 2020. Statistical approach based detection of distributed denial of service attack in a software defined network. In: 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS). pp. 380–385.

Beckett, R., Zou, K., Zhang, S., Malik, S., Rexford, J., Walker, D., 2014. An assertion language for debugging SDN applications. In: ACM SIGCOMM HotSDN Workshop. pp. 1–6, URL https://www.microsoft.com/en-us/research/publication/an-assertion-language-for-debugging-sdn-applications/.

Bian, S., Zhang, P., Yan, Z., 2016. A survey on software-defined networking security. In: Proceedings of the 9th EAI International Conference on Mobile Multimedia Communications. In: MobiMedia '16, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Brussels, BEL, pp. 190–198.

Boero, L., Marchese, M., Zappatore, S., 2017. Support vector machine meets software defined networking in IDS domain. In: 2017 29th International Teletraffic Congress (ITC 29), Vol. 3. pp. 25–30. http://dx.doi.org/10.23919/ITC.2017.8065806.

Bonfim, M.S., Dias, K.L., Fernandes, S.F.L., 2019. Integrated NFV/SDN architectures: A systematic literature review. ACM Comput. Surv. 51 (6), http://dx.doi.org/10.1145/3172866.

Bonfim, M., Santos, M., Dias, K., Fernandes, S., 2020. A real-time attack defense framework for 5g network slicing. Softw. - Pract. Exp. n/a (n/a), http://dx.doi.org/10.1002/spe.2800, URL arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/spe.2800 URL https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.2800.

Braga, R., Mota, E., Passito, A., 2010. Lightweight ddos flooding attack detection using NOX/openflow. In: IEEE Local Computer Network Conference. pp. 408–415. http://dx.doi.org/10.1109/LCN.2010.5735752.

Chao, T., Ke, Y., Chen, B., Chen, J., Hsieh, C.J., Lee, S., Hsiao, H., 2016. Securing data planes in software-defined networks. In: 2016 IEEE NetSoft Conference and Workshops (NetSoft). pp. 465–470. http://dx.doi.org/10.1109/NETSOFT.2016.7502486.

Christidis, K., Devetsikiotis, M., 2016. Blockchains and smart contracts for the internet of things. IEEE Access 4, 2292–2303. http://dx.doi.org/10.1109/ACCESS.2016.2566339.

Dacier, M.C., König, H., Cwalinski, R., Kargl, F., Dietrich, S., 2017. Security challenges and opportunities of software-defined networking. IEEE Secur. Priv. 15 (2), 96–100.

Dalamagkas, C., Sarigiannidis, P., Ioannidis, D., Iturbe, E., Nikolis, O., Ramos, F., Rios, E., Sarigiannidis, A., Tzovaras, D., 2019. A survey on honeypots, honeynets and their applications on smart grid. In: 2019 IEEE Conference on Network Softwarization (NetSoft). pp. 93–100. http://dx.doi.org/10.1109/NETSOFT.2019.8806693.

Dargahi, T., Caponi, A., Ambrosin, M., Bianchi, G., Conti, M., 2017. A survey on the security of stateful SDN data planes. IEEE Commun. Surv. Tutor. 19 (3), 1701–1725. http://dx.doi.org/10.1109/COMST.2017.2689819.

Das, T., Sridharan, V., Gurusamy, M., 2020. A survey on controller placement in SDN. IEEE Commun. Surv. Tutor. 22 (1), 472–503.

Deepa, V., Sudar, K.M., Deepalakshmi, P., 2019. Design of ensemble learning methods for ddos detection in SDN environment. In: 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN). pp. 1–6.

Dhawan, M., Poddar, R., Mahajan, K., Mann, V., 2015. SPHINX: Detecting security attacks in software-defined networks. In: Proceedings 2015 Network and Distributed System Security Symposium. Internet Society, pp. 1–15. http://dx.doi.org/10.14722/ndss.2015.23064.

Doria, A., Salim, J.H., Haas, R., Khosravi, H., Wang, W., Dong, L., Gopal, R., Halpern, J., 2010. Forwarding and Control Element Separation (ForCES) Protocol Specification. RFC 5810, RFC Editor, URL http://www.rfc-editor.org/rfc/rfc5810.txt http://www.rfc-editor.org/rfc/rfc5810.txt.

Durner, R., Lorenz, C., Wiedemann, M., Kellerer, W., 2017. Detecting and mitigating denial of service attacks against the data plane in software defined networks. In: 2017 IEEE Conference on Network Softwarization (NetSoft). pp. 1–6. http://dx.doi.org/10.1109/NETSOFT.2017.8004229.

Duy, P.T., Hien, D.T.T., Pham, V., 2018. A role-based statistical mechanism for ddos attack detection in SDN. In: 2018 5th NAFOSTED Conference on Information and Computer Science (NICS). pp. 177–182.

El Houda, Z.A., Hafid, A., Khoukhi, L., 2019. Co-IoT: A collaborative ddos mitigation scheme in IoT environment based on blockchain using SDN. In: 2019 IEEE Global Communications Conference (GLOBECOM). pp. 1–6. http://dx.doi.org/10.1109/GLOBECOM38437.2019.9013542.

2012. Network Functions Virtualisation – An Introduction, Benefits, Enablers, Challenges & Call for Action. Technical Rep., ETSI.

Fan, W., Du, Z., Fernández, D., Villagrá, V.A., 2018. Enabling an anatomic view to investigate honeypot systems: A survey. IEEE Syst. J. 12 (4), 3906–3919. http://dx.doi.org/10.1109/JSYST.2017.2762161.

Fan, W., Fernández, D., Du, Z., 2017. Versatile virtual honeynet management framework. IET Inf. Secur. 11 (1), 38–45.

Farris, I., Taleb, T., Khettab, Y., Song, J., 2019. A survey on emerging SDN and NFV security mechanisms for IoT systems. IEEE Commun. Surv. Tutor. 21 (1), 812–837. http://dx.doi.org/10.1109/COMST.2018.2862350.

Fernández-Caramés, T.M., Fraga-Lamas, P., 2018. A review on the use of blockchain for the internet of things. IEEE Access 6, 32979–33001.

Fernández Maimó, L., Perales Gómez, A.L., García Clemente, F.J., Gil Pérez, M., Martínez Pérez, G., 2018. A self-adaptive deep learning-based system for anomaly detection in 5g networks. IEEE Access 6, 7700–7712. http://dx.doi.org/10.1109/ACCESS.2018.2803446.

Foundation, O.N., 2020. P4. https://www.opennetworking.org/p4/.

García-Teodoro, P., Díaz-Verdejo, J., Maciá-Fernández, G., Vázquez, E., 2009. Anomaly-based network intrusion detection: Techniques, systems and challenges. Comput. Secur. 28 (1), 18–28. http://dx.doi.org/10.1016/j.cose.2008.08.003, URL http://www.sciencedirect.com/science/article/pii/S0167404808000692.

Gember-Jacobson, A., Viswanathan, R., Prakash, C., Grandl, R., Khalid, J., Das, S., Akella, A., 2014. Opennf: Enabling innovation in network function control. In: Proceedings of the 2014 ACM Conference on SIGCOMM. In: SIGCOMM '14, Association for Computing Machinery, New York, NY, USA, pp. 163–174. http://dx.doi.org/10.1145/2619239.2626313.

Göransson, P., Black, C., Culver, T., 2017. Software Defined Networks: A Comprehensive Approach. Elsevier.

Gupta, A., Vanbever, L., Shahbaz, M., Donovan, S.P., Schlinker, B., Feamster, N., Rexford, J., Shenker, S., Clark, R., Katz-Bassett, E., 2014. SDX: A software defined internet exchange. SIGCOMM Comput. Commun. Rev. 44 (4), 579–580. http://dx.doi.org/10.1145/2740070.2631473.

Haider, S., Akhunzada, A., Mustafa, I., Patel, T.B., Fernandez, A., Choo, K.R., Iqbal, J., 2020. A deep CNN ensemble framework for efficient ddos attack detection in software defined networks. IEEE Access 8, 53972–53983.

Hammerschmidt, C.A., Garcia, S., Verwer, S., State, R., 2017. Reliable machine learning for networking: Key issues and approaches. In: 2017 IEEE 42nd Conference on Local Computer Networks (LCN). pp. 167–170. http://dx.doi.org/10.1109/LCN.2017.74.

Han, W., Zhao, Z., Doupé, A., Ahn, G.-J., 2016. Honeymix: Toward SDN-based intelligent honeynet. In: Proceedings of the 2016 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization. In: SDN-NFV Security '16, Association for Computing Machinery, New York, NY, USA, pp. 1–6. http://dx.doi.org/10.1145/2876019.2876022.

Haque, A.B., Rahman, M., 2020. Blockchain technology: Methodology, application and security issues. arXiv:2012.13366.

Heller, B., Sherwood, R., McKeown, N., 2012. The controller placement problem. In: Proceedings of the First Workshop on Hot Topics in Software Defined Networks. In: HotSDN '12, Association for Computing Machinery, New York, NY, USA, pp. 7–12. http://dx.doi.org/10.1145/2342441.2342444.

Hodo, E., Bellekens, X., Hamilton, A., Tachtatzis, C., Atkinson, R., 2017. Shallow and deep networks intrusion detection system: A taxonomy and survey. arXiv:1701.02145.

Jarschel, M., Oechsner, S., Schlosser, D., Pries, R., Goll, S., Tran-Gia, P., 2011. Modeling and performance evaluation of an openflow architecture. In: Proceedings of the 23rd International Teletraffic Congress. In: ITC '11, International Teletraffic Congress, pp. 1–7.

Jiang, Y., Chen, H., Yang, X., Sun, Z., Quan, W., 2019. Design and implementation of CPU & FPGA co-design tester for SDN switches. Electronics 8 (9), 950. http://dx.doi.org/10.3390/electronics8090950.

Kaljic, E., Maric, A., Njemcevic, P., 2019. An implementation of a deeply programmable SDN switch based on a hybrid FPGA/CPU architecture. In: 2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH). pp. 1–6. http://dx.doi.org/10.1109/INFOTEH.2019.8717768.

Kalkan, K., Altay, L., Gür, G., Alagöz, F., 2018. JESS: Joint entropy-based ddos defense scheme in SDN. IEEE J. Sel. Areas Commun. 36 (10), 2358–2372. http://dx.doi.org/10.1109/JSAC.2018.2869997.

Kalkan, K., Gür, G., Alagöz, F., 2017. Sdnscore: A statistical defense mechanism against ddos attacks in SDN environment. In: 2017 IEEE Symposium on Computers and Communications (ISCC). pp. 669–675.

Kandoi, R., Antikainen, M., 2015. Denial-of-service attacks in openflow SDN networks. In: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM). pp. 1322–1326. http://dx.doi.org/10.1109/INM.2015.7140489.

Kaspersky, 2020. Ddos attacks in Q2 2020. Available at https://securelist.com/ddos-attacks-in-q2-2020/98077/ (2020/01/04).

Khalaf, B.A., Mostafa, S.A., Mustapha, A., Mohammed, M.A., Abduallah, W.M., 2019. Comprehensive review of artificial intelligence and statistical approaches in distributed denial of service attack and defense methods. IEEE Access 7, 51691–51713. http://dx.doi.org/10.1109/ACCESS.2019.2908998.

Khan, S., Gani, A., Abdul Wahab, A.W., Iqbal, S., Abdelaziz, A., Mahdi, O.A., Abdallaahmed, A.I., Shiraz, M., Al-Mayouf, Y.R.B., Khan, Z., Ko, K., Khan, M.K., Chang, V., 2016. Towards an applicability of current network forensics for cloud networks: A SWOT analysis. IEEE Access 4, 9800–9820.

Khan, R., Kumar, P., Jayakody, D.N.K., Liyanage, M., 2020a. A survey on security and privacy of 5g technologies: Potential solutions, recent advancements, and future directions. IEEE Commun. Surv. Tutor. 22 (1), 196–248.

Khan, Z.A., Namin, A.S., 2020. A survey on vulnerabilities of ethereum smart contracts. arXiv:2012.14481.

Khan, L.U., Yaqoob, I., Tran, N.H., Han, Z., Hong, C.S., 2020b. Network slicing: Recent advances, taxonomy, requirements, and open research challenges. IEEE Access 8, 36009–36028.

Kim, H., Feamster, N., 2013. Improving network management with software defined networking. IEEE Commun. Mag. 51 (2), 114–119. http://dx.doi.org/10.1109/MCOM.2013.6461195.

Kim, J., Shin, S., 2017. Software-defined honeynet: Towards mitigating link flooding attacks. In: 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W). pp. 99–100. http://dx.doi.org/10.1109/DSN-W.2017.10.

Kolias, C., Kambourakis, G., Stavrou, A., Voas, J., 2017. Ddos in the IoT: Mirai and other botnets. Computer 50 (7), 80–84. http://dx.doi.org/10.1109/mc.2017.201.

Kolias, C., Kambourakis, G., Stavrou, A., Voas, J., 2017. Ddos in the IoT: Mirai and other botnets. Computer 50 (7), 80–84.

Kosba, A., Miller, A., Shi, E., Wen, Z., Papamanthou, C., 2016. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In: 2016 IEEE Symposium on Security and Privacy (SP). pp. 839–858.

Kreutz, D., Ramos, F.M.V., Veríssimo, P.E., Rothenberg, C.E., Azodolmolky, S., Uhlig, S., 2015. Software-defined networking: A comprehensive survey. Proc. IEEE 103 (1), 14–76. http://dx.doi.org/10.1109/JPROC.2014.2371999.

Kshetri, N., 2017. Can blockchain strengthen the internet of things?. IT Prof. 19 (4), 68–72. http://dx.doi.org/10.1109/mitp.2017.3051335.

Kullback, S., Leibler, R.A., 1951. On information and sufficiency. Ann. Math. Stat. 22 (1), 79–86. http://dx.doi.org/10.1214/aoms/1177729694.

Kuzmanovic, A., Knightly, E.W., 2003. Low-rate TCP-targeted denial of service attacks: The shrew vs. the mice and elephants. In: Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. In: SIGCOMM '03, Association for Computing Machinery, New York, NY, USA, pp. 75–86. http://dx.doi.org/10.1145/863955.863966.

Kyung, S., Han, W., Tiwari, N., Dixit, V.H., Srinivas, L., Zhao, Z., Doupé, A., Ahn, G., 2017. Honeyproxy: Design and implementation of next-generation honeynet via SDN. In: 2017 IEEE Conference on Communications and Network Security (CNS). pp. 1–9. http://dx.doi.org/10.1109/CNS.2017.8228653.

Lal, S., Taleb, T., Dutta, A., 2017. NFV: Security threats and best practices. IEEE Commun. Mag. 55 (8), 211–217. http://dx.doi.org/10.1109/MCOM.2017.1600899.

Lapolli, A.C., Adilson Marques, J., Gaspary, L.P., 2019. Offloading real-time ddos attack detection to programmable data planes. In: 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM). pp. 19–27.

Latah, M., Toker, L., 2018. Towards an efficient anomaly-based intrusion detection for software-defined networks. IET Netw. 7 (6), 453–459. http://dx.doi.org/10.1049/iet-net.2018.5080.

Li, X., Samaka, M., Chan, H.A., Bhamare, D., Gupta, L., Guo, C., Jain, R., 2017. Network slicing for 5g: Challenges and opportunities. IEEE Internet Comput. 21 (5), 20–27.

Liang, X., Znati, T., 2019. A long short-term memory enabled framework for ddos detection. In: 2019 IEEE Global Communications Conference (GLOBECOM). pp. 1–6. http://dx.doi.org/10.1109/GLOBECOM38437.2019.9013450.

Lin, P., Bi, J., Wolff, S., Wang, Y., Xu, A., Chen, Z., Hu, H., Lin, Y., 2015. A west-east bridge based SDN inter-domain testbed. IEEE Commun. Mag. 53 (2), 190–197.

Lin, C., Wu, C., Tian, Y., Wen, Z., Ji, S., 2017. PBUF: Sharing buffer to mitigate flooding attacks. In: 2017 IEEE 23rd International Conference on Parallel and Distributed Systems (ICPADS). pp. 392–399. http://dx.doi.org/10.1109/ICPADS.2017.00059.

Liu, C.-C., Huang, B.-S., Tseng, C.-W., Yang, Y.-T., Chou, L.-D., 2018. SDN/NFV-based moving target ddos defense mechanism. In: International Conference of Reliable Information and Communication Technology. Springer, pp. 548–556.

Liu, C., Malboubi, A., Chuah, C., 2016. Openmeasure: Adaptive flow measurement inference with online learning in SDN. In: 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). pp. 47–52. http://dx.doi.org/10.1109/INFCOMW.2016.7562044.

Liu, Y., Zhao, B., Zhao, P., Fan, P., Liu, H., 2019. A survey: Typical security issues of software-defined networking. China Commun. 16 (7), 13–31. http://dx.doi.org/10.23919/JCC.2019.07.002.

Lu, J., Zhang, Z., Hu, T., Yi, P., Lan, J., 2019. A survey of controller placement problem in software-defined networking. IEEE Access 7, 24290–24307.

Luo, X., Chang, R.K.C., 2005. On a new class of pulsing denial-of-service attacks and the defense. In: In Network and Distributed System Security Symposium (NDSS). pp. 61–79.

Luo, X., Yan, Q., Wang, M., Huang, W., 2019. Using MTD and SDN-based honeypots to defend ddos attacks in IoT. In: 2019 Computing, Communications and IoT Applications (ComComAp). pp. 392–395.

Luo, J., Yu, S., Peng, S., 2020. SDN/NFV-based security service function tree for cloud. IEEE Access 8, 38538–38545.

Macías, S.G., Gaspary, L.P., Botero, J.F., 2020. ORACLE: Collaboration of data and control planes to detect ddos attacks. arXiv preprint arXiv:2009.10798.

Manadhata, P.K., Wing, J.M., 2011. An attack surface metric. IEEE Trans. Softw. Eng. 37 (3), 371–386.

Matias, J., Garay, J., Toledo, N., Unzilla, J., Jacob, E., 2015. Toward an SDN-enabled NFV architecture. IEEE Commun. Mag. 53 (4), 187–193. http://dx.doi.org/10.1109/MCOM.2015.7081093.

McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J., 2008. Openflow: Enabling innovation in campus networks. SIGCOMM Comput. Commun. Rev. 38 (2), 69–74. http://dx.doi.org/10.1145/1355734.1355746.

Mehdi, S.A., Khalid, J., Khayam, S.A., 2011. Revisiting traffic anomaly detection using software defined networking. In: Proceedings of the 14th International Conference on Recent Advances in Intrusion Detection. In: RAID'11, Springer-Verlag, Berlin, Heidelberg, pp. 161–180. http://dx.doi.org/10.1007/978-3-642-23644-0_9.

Memon, M., Hussain, S.S., Bajwa, U.A., Ikhlas, A., 2018. Blockchain beyond bit-coin: Blockchain technology challenges and real-world applications. In: 2018 International Conference on Computing, Electronics Communications Engineering (ICCECE). pp. 29–34. http://dx.doi.org/10.1109/iCCECOME.2018.8658518.

Mestres, A., Rodriguez-Natal, A., Carner, J., Barlet-Ros, P., Alarcón, E., Solé, M., Muntés-Mulero, V., Meyer, D., Barkai, S., Hibbett, M.J., et al., 2017. Knowledge-defined networking. SIGCOMM Comput. Commun. Rev. 47 (3), 2–10. http://dx.doi.org/10.1145/3138808.3138810.

Meti, N., Narayan, D.G., Baligar, V.P., 2017. Detection of distributed denial of service attacks using machine learning algorithms in software defined networks. In: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI). pp. 1366–1371. http://dx.doi.org/10.1109/ICACCI.2017.8126031.

Mijumbi, R., Serrat, J., Gorricho, J., Bouten, N., De Turck, F., Boutaba, R., 2016. Network function virtualization: State-of-the-art and research challenges. IEEE Commun. Surv. Tutor. 18 (1), 236–262. http://dx.doi.org/10.1109/COMST.2015.2477041.

Mohammed, A.R., Mohammed, S.A., Shirmohammadi, S., 2019. Machine learning and deep learning based traffic classification and prediction in software defined networking. In: 2019 IEEE International Symposium on Measurements Networking (M N). pp. 1–6. http://dx.doi.org/10.1109/IWMN.2019.8805044.

Monshizadeh, M., Khatri, V., Kantola, R., 2017. Detection as a service: An SDN application. In: 2017 19th International Conference on Advanced Communication Technology (ICACT). pp. 285–290. http://dx.doi.org/10.23919/ICACT.2017.7890099.

Montero, D., Yannuzzi, M., Shaw, A., Jacquin, L., Pastor, A., Serral-Gracia, R., Lioy, A., Risso, F., Basile, C., Sassu, R., Nemirovsky, M., Ciaccia, F., Georgiades, M., Charalambides, S., Kuusijarvi, J., Bosco, F., 2015. Virtualized security at the network edge: a user-centric approach. IEEE Commun. Mag. 53 (4), 176–186. http://dx.doi.org/10.1109/MCOM.2015.7081092.

Morales, L.V., Murillo, A.F., Rueda, S.J., 2015. Extending the floodlight controller. In: 2015 IEEE 14th International Symposium on Network Computing and Applications. pp. 126–133. http://dx.doi.org/10.1109/NCA.2015.11.

Mortensen, A., Reddy.K, T., Andreasen, F., Teague, N., Compton, R., 2020. DDoS Open Threat Signaling (DOTS) Architecture. RFC 8811, RFC Editor.

Mubarakali, A., Alqahtani, A.S., 2019. A survey: Security threats and countermeasures in software defined networking. In: 2019 IEEE 2nd International Conference on Information and Computer Technologies (ICICT). pp. 180–185. http://dx.doi.org/10.1109/INFOCT.2019.8711319.

Narantuya, J., Yoon, S., Lim, H., Cho, J., Kim, D.S., Moore, T., Nelson, F., 2019. SDN-based IP shuffling moving target defense with multiple SDN controllers. In: 2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks – Supplemental Volume (DSN-S). pp. 15–16.

Nguyen, T.N., 2018. The challenges in ML-based security for SDN. 2018 2nd Cyber Security in Networking Conference (CSNet) http://dx.doi.org/10.1109/csnet.2018.8602680.

Nguyen, H.T., Petrović, S., Franke, K., 2010. A comparison of feature-selection methods for intrusion detection. In: Kotenko, I., Skormin, V. (Eds.), Computer Network Security. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 242–255.

Oo, N.H., Htein Maw, A., 2019. Effective detection and mitigation of SYN flooding attack in SDN. In: 2019 19th International Symposium on Communications and Information Technologies (ISCIT). pp. 300–305.

Ordonez-Lucena, J., Ameigeiras, P., Lopez, D., Ramos-Munoz, J.J., Lorca, J., Folgueira, J., 2017. Network slicing for 5g with SDN/NFV: concepts, architectures, and challenges. IEEE Commun. Mag. 55 (5), 80–87.

Pan, X., Yegneswaran, V., Chen, Y., Porras, P., Shin, S., 2016. Hogmap: Using SDNs to incentivize collaborative security monitoring. In: Proceedings of the 2016 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization. In: SDN-NFV Security '16, ACM, New York, NY, USA, pp. 7–12. http://dx.doi.org/10.1145/2876019.2876023, URL http://doi.acm.org/10.1145/2876019.2876023.

Peng, H., Sun, Z., Zhao, X., Tan, S., Sun, Z., 2018. A detection method for anomaly flow in software defined network. IEEE Access 6, 27809–27817. http://dx.doi.org/10.1109/ACCESS.2018.2839684.

Pérez-Díaz, J.A., Valdovinos, I.A., Choo, K.K.R., Zhu, D., 2020. A flexible SDN-based architecture for identifying and mitigating low-rate ddos attacks using machine learning. IEEE Access 8, 155859–155872. http://dx.doi.org/10.1109/ACCESS.2020.3019330.

Pourvahab, M., Ekbatanifard, G., 2019a. Digital forensics architecture for evidence collection and provenance preservation in iaas cloud environment using SDN and blockchain technology. IEEE Access 7, 153349–153364.

Pourvahab, M., Ekbatanifard, G., 2019b. An efficient forensics architecture in software-defined networking-IoT using blockchain technology. IEEE Access 7, 99573–99588.

Revathi, S., Malathi, A., 2013. A detailed analysis on NSL-kdd dataset using various machine learning techniques for intrusion detection. Int. J. Eng. 2 (12).

Rinaldi, G., Adamsky, F., Soua, R., Baiocchi, A., Engel, T., 2019. Softwarization of SCADA: Lightweight statistical SDN-agents for anomaly detection. In: 2019 10th International Conference on Networks of the Future (NoF). pp. 102–109.

Rodrigues, B., Bocek, T., Stiller, B., 2017. Enabling a cooperative, multi-domain ddos defense by a blockchain signaling system (bloss). In: 42nd IEEE Conference on Local Computer Networks 2017 (LCN 2017). IEEE, Singapore, Singapore, pp. 1–3, URL https://doi.org/10.5167/uzh-146079.

Rost, P., Mannweiler, C., Michalopoulos, D.S., Sartori, C., Sciancalepore, V., Sastry, N., Holland, O., Tayade, S., Han, B., Bega, D., Aziz, D., Bakker, H., 2017. Network slicing to enable scalability and flexibility in 5g mobile networks. IEEE Commun. Mag. 55 (5), 72–79.

Salman, T., Zolanvari, M., Erbad, A., Jain, R., Samaka, M., 2019. Security services using blockchains: A state of the art survey. IEEE Commun. Surv. Tutor. 21 (1), 858–880.

Sarwar, M.A., Hussain, M., Anwar, M.U., Ahmad, M., 2019. Flowjustifier: An optimized trust-based request prioritization approach for mitigation of SDN controller ddos attacks in the IoT paradigm. In: Proceedings of the 3rd International Conference on Future Networks and Distributed Systems. In: ICFNDS '19, Association for Computing Machinery, New York, NY, USA, pp. 1–9. http://dx.doi.org/10.1145/3341325.3342037, URL https://doi.org/10.1145/3341325.3342037.

Sathi, V.N., Srinivasan, M., Kaliyammal Thiruvasagam, P., Murthy, S.R., 2020. Novel protocols to mitigate network slice topology learning attacks and protect privacy of users' service access behavior in softwarized 5g networks. IEEE Trans. Dependable Secure Comput. 1.

Sattar, D., Matrawy, A., 2019. Towards secure slicing: Using slice isolation to mitigate ddos attacks on 5g core network slices. In: 2019 IEEE Conference on Communications and Network Security (CNS). pp. 82–90.

Sayadi, B., Gramaglia, M., Friderikos, V., von Hugo, D., Arnold, P., Alberi-Morel, M.-L., Puente, M.A., Sciancalepore, V., Digon, I., Crippa, M.R., 2016. SDN for 5g mobile networks: NORMA perspective. In: International Conference on Cognitive Radio Oriented Wireless Networks. Springer, pp. 741–753.

Scott-Hayward, S., Natarajan, S., Sezer, S., 2016. A survey of security in software defined networks. IEEE Commun. Surv. Tutor. 18 (1), 623–654. http://dx.doi.org/10.1109/COMST.2015.2453114.

Scott-Hayward, S., O'Callaghan, G., Sezer, S., 2013. SDN security: A survey. In: 2013 IEEE SDN for Future Networks and Services (SDN4FNS). pp. 1–7. http://dx.doi.org/10.1109/SDN4FNS.2013.6702553.

Sengupta, S., Chowdhary, A., Sabur, A., Alshamrani, A., Huang, D., Kambhampati, S., 2019. A survey of moving target defenses for network security. arXiv:1905.00964.

Sengupta, S., Chowdhary, A., Sabur, A., Alshamrani, A., Huang, D., Kambhampati, S., 2020. A survey of moving target defenses for network security. IEEE Commun. Surv. Tutor. 1.

Serbanescu, A.V., Obermeier, S., Yu, D.-Y., 2015. ICS Threat analysis using a large-scale honeynet. In: Proceedings of the 3rd International Symposium for ICS & SCADA Cyber Security Research. In: ICS-CSR '15, BCS Learning & Development Ltd., Swindon, GBR, pp. 20–30. http://dx.doi.org/10.14236/ewic/ICS2015.3.

Shafi, Q., Basit, A., 2019. Ddos botnet prevention using blockchain in software defined internet of things. In: 2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST). pp. 624–628. http://dx.doi.org/10.1109/IBCAST.2019.8667147.

Shaghaghi, A., Kaafar, M.A., Buyya, R., Jha, S., 2018. Software-defined network (SDN) data plane security: Issues, solutions and future directions. arXiv:1804.00262.

Shaghaghi, A., Kaafar, M.A., Jha, S., 2017. Wedgetail: An intrusion prevention system for the data plane of software defined networks. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security. In: ASIA CCS '17, Association for Computing Machinery, New York, NY, USA, pp. 849–861. http://dx.doi.org/10.1145/3052973.3053039.

Sharafaldin., I., Lashkari., A.H., Ghorbani., A.A., 2018. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: Proceedings of the 4th International Conference on Information Systems Security and Privacy - Volume 1: ICISSP,. SciTePress, INSTICC, pp. 108–116. http://dx.doi.org/10.5220/0006639801080116.

Sherry, J., Hasan, S., Scott, C., Krishnamurthy, A., Ratnasamy, S., Sekar, V., 2012. Making middleboxes someone else's problem: Network processing as a cloud service. In: Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication. In: SIGCOMM '12, Association for Computing Machinery, New York, NY, USA, pp. 13–24. http://dx.doi.org/10.1145/2342356.2342359.

Shevtekar, A., Anantharam, K., Ansari, N., 2005. Low rate TCP denial-of-service attack detection at edge routers. IEEE Commun. Lett. 9 (4), 363–365.

Shin, S., Porras, P., Yegneswaran, V., Fong, M., Gu, G., Tyson, M., 2013a. FRESCO: Modular composable security services for software defined networks. In: In Proceedings of Network and Distributed Security Symposium, pp. 1–16.

Shin, S., Yegneswaran, V., Porras, P., Gu, G., 2013b. AVANT-Guard: Scalable and vigilant switch flow management in software-defined networks. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security. In: CCS '13, Association for Computing Machinery, New York, NY, USA, pp. 413–424. http://dx.doi.org/10.1145/2508859.2516684.

Singh, R., Tanwar, S., Sharma, T.P., 2019. Utilization of blockchain for mitigating the distributed denial of service attacks. Secur. Priv. 3 (3), http://dx.doi.org/10.1002/spy2.96.

Siris, V.A., Papagalou, F., 2004. Application of anomaly detection algorithms for detecting SYN flooding attacks. In: IEEE Global Telecommunications Conference, 2004. GLOBECOM '04., Vol. 4. pp. 2050–2054, Vol.4.

Steichen, M., Hommes, S., State, R., 2017. Chainguard — A firewall for blockchain applications using SDN with openflow. In: 2017 Principles, Systems and Applications of IP Telecommunications (IPTComm). pp. 1–8. http://dx.doi.org/10.1109/IPTCOMM.2017.8169748.

Steinberger, J., Kuhnert, B., Dietz, C., Ball, L., Sperotto, A., Baier, H., Pras, A., Dreo, G., 2018. Ddos defense using MTD and SDN. In: NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium. pp. 1–9.

Sultana, N., Chilamkurti, N., Peng, W., Alhadad, R., 2019. Survey on SDN based network intrusion detection system using machine learning approaches. Peer-to-Peer Netw. Appl. 12 (2), 493–501. http://dx.doi.org/10.1007/s12083-017-0630-0.

Swami, R., Dave, M., Ranga, V., 2019a. Software-defined networking-based ddos defense mechanisms. ACM Comput. Surv. 52 (2), 28:1–28:36. http://dx.doi.org/10.1145/3301614, URL http://doi.acm.org/10.1145/3301614.

Swami, R., Dave, M., Ranga, V., 2019b. Software-defined networking-based ddos defense mechanisms. ACM Comput. Surv. 52 (2), http://dx.doi.org/10.1145/3301614.

Tang, T.A., Mhamdi, L., McLernon, D., Zaidi, S.A.R., Ghogho, M., 2016. Deep learning approach for network intrusion detection in software defined networking. In: 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM). pp. 258–263. http://dx.doi.org/10.1109/WINCOM.2016.7777224.

Tang, T.A., Mhamdi, L., McLernon, D., Zaidi, S.A.R., Ghogho, M., El Moussa, F., 2020. Deepids: Deep learning approach for intrusion detection in software defined networking. Electronics 9 (9), 1533. http://dx.doi.org/10.3390/electronics9091533.

Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A., 2009. A detailed analysis of the KDD cup 99 data set. In: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications. pp. 1–6. http://dx.doi.org/10.1109/CISDA.2009.5356528.

Thantharate, A., Paropkari, R., Walunj, V., Beard, C., Kankariya, P., 2020. Secure5g: A deep learning framework towards a secure network slicing in 5g and beyond. In: 2020 10th Annual Computing and Communication Workshop and Conference (CCWC). pp. 0852–0857.

Tootoonchian, A., Gorbunov, S., Ganjali, Y., Casado, M., Sherwood, R., 2012. On controller performance in software-defined networks. In: Proceedings of the 2nd USENIX Conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services. In: Hot-ICE'12, USENIX Association, USA, p. 10.

Tselios, C., Politis, I., Kotsopoulos, S., 2017. Enhancing SDN security for IoT-related deployments through blockchain. In: 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN). pp. 303–308. http://dx.doi.org/10.1109/NFV-SDN.2017.8169860.

Walfish, M., Stribling, J., Krohn, M., Balakrishnan, H., Morris, R., Shenker, S., 2004. Middleboxes no longer considered harmful. In: Proceedings of the 6th Conference on Symposium on Operating Systems Design & Implementation - Volume 6. In: OSDI'04, USENIX Association, USA, p. 15.

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

Wang, L., Jones, R., 2017a. Big data analytics for network intrusion detection: A survey. Int. J. Netw. Commun. 7 (1), http://dx.doi.org/10.5923/j.ijnc.20170701.03.

Wang, L., Jones, R., 2017b. Big data analytics for network intrusion detection: A survey. Int. J. Netw. Commun. 24–31.

Wang, Z., Qian, Z., Xu, Q., Mao, Z., Zhang, M., 2011. An untold story of middleboxes in cellular networks. SIGCOMM Comput. Commun. Rev. 41 (4), 374–385. http://dx.doi.org/10.1145/2043164.2018479.

Wang, H., Wu, B., 2019. SDN-based hybrid honeypot for attack capture. In: 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). pp. 1602–1606.

Wang, G., Zhao, Y., Huang, J., Wang, W., 2017. The controller placement problem in software defined networking: A survey. IEEE Netw. 31 (5), 21–27.

Wei, L., Fung, C., 2015. Flowranger: A request prioritizing algorithm for controller dos attacks in software defined networks. In: 2015 IEEE International Conference on Communications (ICC). pp. 5254–5259.

Wen, X., Chen, Y., Hu, C., Shi, C., Wang, Y., 2013. Towards a secure controller platform for openflow applications. In: Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking. In: HotSDN '13, Association for Computing Machinery, New York, NY, USA, pp. 171–172. http://dx.doi.org/10.1145/2491185.2491212.

Xie, J., Yu, F.R., Huang, T., Xie, R., Liu, J., Wang, C., Liu, Y., 2019. A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges. IEEE Commun. Surv. Tutor. 21 (1), 393–430. http://dx.doi.org/10.1109/COMST.2018.2866942.

Yan, Q., Yu, F.R., Gong, Q., Li, J., 2016. Software-defined networking (SDN) and distributed denial of service (ddos) attacks in cloud computing environments: A survey, some research issues, and challenges. IEEE Commun. Surv. Tutor. 18 (1), 602–622. http://dx.doi.org/10.1109/COMST.2015.2487361.

Yang, W., Aghasian, E., Garg, S., Herbert, D., Disiuta, L., Kang, B., 2019. A survey on blockchain-based internet service architecture: Requirements, challenges, trends, and future. IEEE Access 7, 75845–75872. http://dx.doi.org/10.1109/ACCESS.2019.2917562.

Yang, W., Fung, C., 2016. A survey on security in network functions virtualization. In: 2016 IEEE NetSoft Conference and Workshops (NetSoft). pp. 15–19. http://dx.doi.org/10.1109/NETSOFT.2016.7502434.

Yazdinejad, A., Bohlooli, A., Jamshidi, K., 2019. Performance improvement and hardware implementation of open flow switch using FPGA. In: 2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI). pp. 515–520. http://dx.doi.org/10.1109/KBEI.2019.8734914.

Yeganeh, S.H., Tootoonchian, A., Ganjali, Y., 2013. On scalability of software-defined networking. IEEE Commun. Mag. 51 (2), 136–141.

Yu, Y., Guo, L., Liu, Y., Zheng, J., Zong, Y., 2018. An efficient SDN-based ddos attack detection and rapid response platform in vehicular networks. IEEE Access 6, 44570–44579. http://dx.doi.org/10.1109/ACCESS.2018.2854567.

Zarca, A.M., Bernabe, J.B., Skarmeta, A., Alcaraz Calero, J.M., 2020. Virtual IoT honeynets to mitigate cyberattacks in SDN/NFV-enabled IoT networks. IEEE J. Sel. Areas Commun. 38 (6), 1262–1277.

Zhang, H., Liu, N., Chu, X., Long, K., Aghvami, A.-H., Leung, V.C., 2017. Network slicing based 5g and future mobile networks: mobility, resource management, and challenges. IEEE Commun. Mag. 55 (8), 138–145.

Zhao, Y., Li, Y., Zhang, X., Geng, G., Zhang, W., Sun, Y., 2019. A survey of networking applications applying the software defined networking concept based on machine learning. IEEE Access 7, 95397–95417. http://dx.doi.org/10.1109/ACCESS.2019.2928564.

Zheng, J., Namin, A.S., 2019. A survey on the moving target defense strategies: An architectural perspective. J. Comput. Sci. Tech. 34 (1), 207–233. http://dx.doi.org/10.1007/s11390-019-1906-z.

Zheng, Z., Xie, S., Dai, H.-N., Chen, X., Wang, H., 2018. Blockchain challenges and opportunities: A survey. Int. J. Web Grid Serv. 14, 352. http://dx.doi.org/10.1504/IJWGS.2018.095647.

Zhou, L., Guo, H., 2017. Applying NFV/SDN in mitigating ddos attacks. In: TENCON 2017 - 2017 IEEE Region 10 Conference. pp. 2061–2066. http://dx.doi.org/10.1109/TENCON.2017.8228200.

Zhu, L., Tang, X., Shen, M., Du, X., Guizani, M., 2018. Privacy-preserving ddos attack detection using cross-domain traffic in software defined networks. IEEE J. Sel. Areas Commun. 36 (3), 628–643. http://dx.doi.org/10.1109/JSAC.2018.2815442.

Zhuang, R., DeLoach, S.A., Ou, X., 2014. Towards a theory of moving target defense. In: Proceedings of the First ACM Workshop on Moving Target Defense. In: MTD '14, Association for Computing Machinery, New York, NY, USA, pp. 31–40. http://dx.doi.org/10.1145/2663474.2663479.

Ismael Amezcua Valdovinos obtained his B.Sc. degree in Computer Science from Universidad de Colima in 2007 and earned his Ph.D. from Tecnológico de Monterrey, Campus Cuernavaca in 2013, where he worked on developing communication protocols for multi-homed devices. He is a professor at Facultad de Telemática, Universidad de Colima, in México. His research interests are Wireless Sensor Networks (WSN), Industrial Internet of Things (IIoT), and Software-Defined Networks (SDN).

Jesús Arturo Pérez Díaz obtained his B.Sc. degree in computer science from the Autonomous University of Aguascalientes in 1995, where he received the best student award. He received his Ph.D. degree in New Advances in Computer Science Systems from the Universidad de Oviedo in 2000. He became a full associate professor at University of Oviedo from 2000 to 2002. He was recognized by the COIMBRA group as one of the best young Latin-American researchers in 2006 and received a research stay at Louvain le nouveau University in Belgium. He has been awarded by the CIGRE and by Intel for the development of innovative systems. He has published more than 30 scientific papers in several journals and international conferences. Currently he is a researcher and professor in the ITESM — Campus Querétaro, México and member of the Mexican Researchers National System. His research field focus on the mitigation of DDoS attacks in SDN and the design of communications protocols where he has supervised several master and Ph.D. theses in the field.

Kim-Kwang Raymond Choo received the Ph.D. in Information Security in 2006 from Queensland University of Technology, Australia. He currently holds the Cloud Technology Endowed Professorship at The University of Texas at San Antonio (UTSA). He is the founding chair of IEEE Technology and Engineering Management Society (TEMS)'s Technical Committee on Blockchain and Distributed Ledger Technologies, an ACM Distinguished Speaker and an IEEE Computer Society Distinguished Visitor (2021–2023), and included in Web of Science's Highly Cited Researcher in the field of Cross-Field - 2020. In 2015, he and his team won the Digital Forensics Research Challenge organized by Germany's University of Erlangen–Nuremberg. He is the recipient of the 2019 IEEE Technical Committee on Scalable Computing Award for Excellence in Scalable Computing (Middle Career Researcher), the 2018 UTSA College of Business Col. Jean Piccione and Lt. Col. Philip Piccione Endowed Research Award for Tenured Faculty, the Outstanding Associate Editor of 2018 for IEEE Access, the British Computer Society's 2019 Wilkes Award Runner-up, the 2014 Highly Commended Award by the Australia New Zealand Policing Advisory Agency, the Fulbright Scholarship in 2009, the 2008 Australia Day Achievement Medallion, and the British Computer Society's Wilkes Award in 2008. He has also received best paper awards from the IEEE Systems Journal in 2021, the IEEE Consumer Electronics Magazine for 2020, EURASIP Journal on Wireless Communications and Networking in 2019, IEEE TrustCom 2018, and ESORICS 2015; the Korea Information Processing Society's Journal of Information Processing Systems (JIPS) Outstanding Research Award (Most-cited Paper) for 2020 and Survey Paper Award (Gold) in 2019; the IEEE Blockchain 2019 Outstanding Paper Award; and Best Student Paper Awards from Inscrypt 2019 and ACISP 2005.

Juan Felipe Botero received the B.S. degree in computer science from University of Antioquia, Medellín, Colombia, in 2006, and the M.Sc. and Ph.D. degrees in telematics engineering from the Network Engineering Department of the Technical University of Catalonia, Barcelona, Spain, in 2008 and 2013 respectively. Since 2013, he has been an Associate Professor with the Telecommunications Engineering Department, University of Antioquia. Currently, he belongs to the applied telecommunications research group at the University of Antioquia. His main research interests include Internet of the Future, in particular Network Virtualization, Mathematical Programming, Routing, Energy Efficiency and Network flows, SDN and NFV.