A Survey on Machine Learning Applications for Software Defined Network Security

Juliana Arevalo Herrera[1(✉)] and Jorge E. Camargo[2(✉)]

[1] Universidad Santo Tomás, Bogotá, Colombia
julianaarevalo@usantotomas.edu.co
[2] Universidad Nacional de Colombia, Bogotá, Colombia
jecamargom@unal.edu.co
http://www.unsecurelab.org

Abstract. The number of machine learning (ML) applications on networking security has increased recently thanks to the availability of processing and storage capabilities. Combined with new technologies such as Software Defined Networking (SDN) and Network Function Virtualization (NFV), it becomes an even more interesting topic for the research community. In this survey, we present studies that employ ML techniques in SDN environments for security applications. The surveyed papers are classified into ML techniques (used to identify general anomalies or specific attacks) and IDS frameworks for SDN. The latter category is relevant since reviewed paers include the implementation of data collection and mitigation techniques, besides just defining a ML model, as the first category. We also identify the standard datasets, testbeds, and additional tools for researchers.

Keywords: Software defined networks · Machine learning · Network security

1 Introduction

Separation of control and data planes is not a new idea, but only recently it has obtained high interest from the scientific community and commercial vendors with the popularization of Software Defined Networks (SDN). There have been several contributions to the technology, but it is still under development by the industry and academic community. In combination with other technologies such as Network Function Virtualization (NFV), SDN approach presents a solution to everyday problems existing in traditional networks like scalability and manageability issues. Additionally, it offers alternatives to monitor and control the traffic in the network, providing new possibilities for security applications. However, the de-facto protocol for control-data communication, OpenFlow [42], has been identified as a vulnerable solution [26]. It also presents additional security issues, as we will show in Sect. 3.

J. Zhou et al. (Eds.): ACNS 2019 Workshops, LNCS 11605, pp. 70–93, 2019.
https://doi.org/10.1007/978-3-030-29729-9_4

SDN definition is comprised of three layers. However, as technology develops, additional elements are required. In [18] Clark et al. propose a Knowledge Plane or KP as an individual entity for the network that aims to maintain a high-level view of the network and help in the operation, management, and improvement. Knowledge Defined Networking (KDN) [44], adds a knowledge plane (KP) to the SDN architecture, intending to integrate behavioral models and reasoning processes oriented to decision making. One of the tools to leverage the KP is Machine Learning.

Machine Learning is a powerful tool to provide cognitive capabilities for identifying security breaches. It has a significant improvement due to the processing and storage capabilities as well as the availability of large datasets. However, SDN is not broadly used in operative networks, though there is an important reference: Google's B4 [43] is a deployment of SDN over WAN network to connect several data centers. It included a switch design to handle the interconnection with traditional networks and ONIX [30] as the controller. It was proven to be a useful technique for the gradual integration of traditional to SDN infrastructure. The implementation did not present any contribution related to security, except for the use of the Paxos algorithm [33] for fault tolerance. Considering there are no available data on security research in SDN, obtaining realistic datasets for IDS becomes a challenge.

On [50], authors present an overview of the challenges and opportunities to use ML in new technologies such as SDN, however it is not exhaustive in the description or study. Other works such as [19,32,49,61] have shown different ML techniques applicable to SDN anomaly detection but focus on the methods and lack of an analysis from the network security perspective.

In this paper, we present the most recent research (to the best of our knowledge) for network security in an SDN environment using ML techniques. Our motivation is to contribute to the creation of a KP for SDN, focused on security. The study presents the surveyed papers organized per network attacks, in contrast to other surveys related to ML methods used in SDN. It also shows the testbeds and datasets commonly used in the literature.

The rest of the paper is organized as follows. Section 2 presents the methodology used to select and classify the studies. Section 3 presents an overview of the SDN architecture and security issues. Section 4 presents the studies for ML-based techniques for IDS, only with a proposal of the detection model. Section 5 presents studies that include methods to collect data to feed the ML model, as well as mitigation schemes once the anomaly is detected. Section 6 aims to provide additional tools for researches with studies related to security, as well as used datasets and testbeds in the surveyed works. Finally, Sect. 7 concludes the presented survey.

2 Methodology

This survey focuses on the works that use machine learning (ML) including deep learning (DP) techniques to address security issues for software defined networks (SDN). We initially set the period of the publication to be used in the study as

five years; however, during the first search within databases, we found valuable literature since the year 2013. Because of this, the publication period covers papers from that year until the beginning of 2019.

To search for papers for our study, we reviewed the IEEE Xplore, ScienceDirect, and Wiley databases, as well as Google Scholar to expand the scope to other repositories. The key-words used to conduct the study were: "SDN," "Security," "Machine learning," and "Deep learning." We combined the terms to create different search streams such as: ("SDN security" AND "machine learning"), ("SDN security" AND "deep learning"), OR ("SDN Security"). Only the titles of the studies were considered to select an initial list of 200 papers. Later, we classified the articles into those to be used in the survey and those to be excluded by reviewing the abstract, introduction and conclusion, only.

We selected papers that included all areas of the keywords (SDN, ML/DP, and network security), but also those that presented traffic classification or monitoring, since those methods are useful for securing the network. Out of the selected papers, we classified them into the following categories:

- Surveys
- Proposal for framework or security application
- Experiment of existing tools

Using this classification, we selected a total of 70 papers and excluded out of the initial list. These papers were reviewed in detail, and using them we identified other studies to be included.

3 SDN Architecture and Security

SDN was born out of the need to break the vertical integration of the network equipment. Its premise is to separate the control from the data plane, and the interface between them is OpenFlow (OF) [42] protocol, proposed in 2008, which leveraged its development. It also allows defining network functions (e.g., routing, firewall, load balancing, bandwidth optimization) as software applications that can run on top of the control plane. The architecture has three parts: data plane (composed of switches), the control plane (composed of one or more controllers), and application plane (composed of one or more network applications). Figure 1 shows an SDN architecture.

Within SDN, a flow is a set of packets with similar features that go from one endpoint (or group) to another endpoint (or group) in a single direction. Each flow has its entry in the flow table, which is a database within the switches consulted to determine what to do with each packet that arrives at the switch. The flow-tables are created by order of the controller. At the beginning of a transmission (new flow), the switch will receive a packet without an entry on the flow table. The OF protocol sends the "Packet_in" message, from the switch to the controller for analysis and definition of a new flow-table entry. The "Packet_in" is a particular feature that could become a vulnerability to the system. OF also defines the information collection, using a request from the controller that the switch answers with parts of the flow-table along with packet counters.

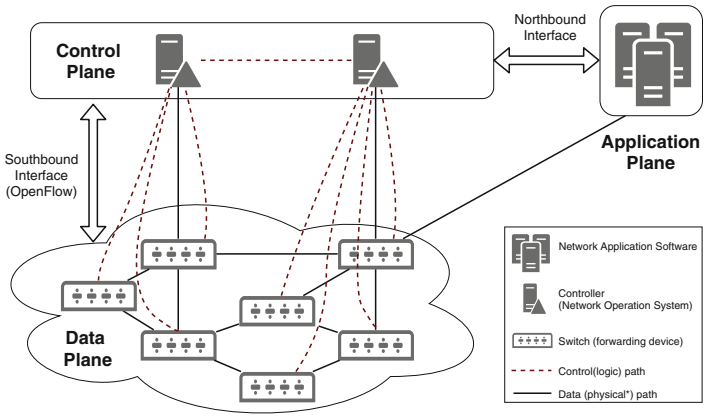

Fig. 1. SDN Architecture: Data plane, Control plane and Applications plane as its main components

This new paradigm represents a solution to several problems of traditional networks, such as manageability, configuration, scalability, and security. Under this perspective, a clear advantage for security with SDN is the ability to gather traffic information without additional elements. This is due to the centralized role of the controller, which communicates with the switches in the data plane. Proposals such as [8,27,47] take advantage of this ability to implement security functions such as Intrusion Detection Systems (IDS) and protection against Distributed Denial of Service (DDoS) within the network. SDN is, however, a model under development with open research lines and security challenges common to traditional networks, as well as unique to it. Different studies [7,31,57] presented analysis to network security from different viewpoints. A common conclusion is that security applications in SDN are still not mature enough for widespread implementation. A non-exhaustive review of SDN security architecture issues is presented below.

Kreutz et al. [31], created one of the firsts attempts to determine the vulnerabilities in SDN architecture. In this survey, the authors presented seven threat vectors: (1) Forged or faked traffic flow; (2) Attacks on vulnerabilities in switches; (3) Attacks on control plane communications; (4) Attacks on and vulnerabilities in controllers; (5) Lack of mechanisms to ensure trust between the controller and management applications; (6) Attacks on and vulnerabilities in administrative stations; and (7) Lack of trusted resources for forensics and remediation. Other studies [56,71] also used this scheme to analyze SDN security. The paper also proposes the mechanisms required to secure a controller: Replication, Diversity, Self-healing mechanisms, Dynamic device association, Trust between devices and controllers, as well as between controllers and applications, Security domains, Secure components, and Dependable maintenance of software.

The first attack vector was exploited in [58]. Initially, they detect if a given network uses SDN by comparing the response times. If it is SDN, at the

beginning of the transmission the response time is longer, since the network has a "flow setup" latency. The times have subtle differences, so the authors present a solution with an SDN scanner. After the confirmation that the network is an SDN, specially crafted traffic is sent to the network to cause data plane resource consumption or Distributed Denial of Service (DDoS) attacks.

Moving Target Defense (MTD) is a widespread approach used by several studies. In [17], authors proposed a framework to prevent, detect and mitigate attacks. The research was directed to virtualized environments in the cloud and presented two areas to secure resources. First, the authors studied MTD for network programmability and software vulnerability. Then, traffic engineering was reviewed. The latter allows the provision of different tenants securely. For the former one, a set of countermeasures must be included to enforce after the detection and analysis with an attack graph (AG) based vulnerability analysis.

The same approach was studied in detail in [16]. The authors presented AG techniques to reconfigure the network automatically and used MTD as a countermeasure. However, it does not present information on the attack detection but assumes the intrusion detection already in place. It still needs a phase for attack analysis in which ML could be used.

Few studies present machine learning solutions for the SDN architecture security problems identified by [31]. However, some works suggest the possibility to use it. In [64], authors presented three levels of complexity to use cognition: Reactive reasoning (rule-based reaction), Tactical reasoning (Profiling based on classification with dynamic multi-objective optimization), and Strategical reasoning (Anticipation with online multi-objective optimization). The study proposes to formulate optimization functions related to the security concerns in the network.

On the other hand, in [29] authors presented a framework to provide autonomous response and mitigation against attacks in an SDN/NFV network. The approach is called SARNET and has a transverse loop with five stages: Detect, Analyze, Decide, Respond, and Learn. An essential contribution of the study is the definition of an efficiency estimation that allows measuring the performance of the proposed framework. A group of simulations of different attacks (UDP DDoS attack, CPU utilization attack, Password attack) showed that the efficiency measure helps in selecting the best countermeasure. Within all the loop, it is suitable to use ML, and the authors present it as future research.

As presented, there is not extensive research to secure the SDN architecture using ML. However, SDN architecture can leverage network security since it allows the managers to know, rather than infer, the specific status of the network. OF gives the opportunity to collect statistics and traffic information that could be used to identify anomalies, intruders or configuration failures within the controller, devices or applications.

These abilities present the possible implementation of security applications on top of the SDN architecture. They are also leveraged by the use of Network Function Virtualization (NFV). NFV intends to apply IT virtualization technology for networking functions [15], and the objective is to break the dependence

of hardware. In this scenario, security applications can be implemented on commodity devices, and the necessity of specific equipment could be eliminated.

In the following sections, we will discuss the different proposals to use SDN as a mean to improve network security. Our approach is to analyze the use of machine learning to achieve the desired result. As presented in Fig. 2, we classified the papers into Type 1: ML-based intrusion detection Systems in SDN, and Type 2: ML-based intrusion detection Systems in SDN. In the first case, the sub-classification depends on the type of detected attack. In the second case, the sub-classification depends on the data collection method to feed the ML-Model.

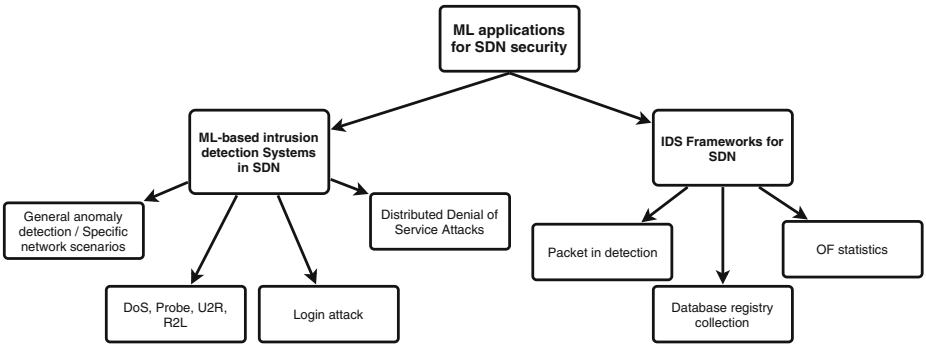

Fig. 2. Classification of studies in the survey

4 ML-Based Intrusion Detection Systems in SDN

Intrusion detection systems (IDS) are one of the most widespread applications for security in SDN. Since OF provides traffic statistics using the messages "StatsRequest" and "StatsResponse," it becomes a compelling tool to identify anomalies and intruders.

Fundamentals of IDS operations apply equally for traditional and SDN environments. Considering the location of the method IDS techniques can be divided into Network IDS and Host IDS. The former performs intrusion detection by analyzing the overall situation of the network. On the other hand, HIDS is host-based detection that monitors the operation of a particular device.

As detection mechanisms, IDS employ two types of strategies: (1) Traditional, signature-based detection that compares data to an existing database; and (2) Anomaly-based detection, which identifies odd-behaviour traffic, and can make use of ML techniques for better results. Examples of IDS proposals with the traditional approach in SDN are [14,40,72]. For instance in [40] of the first attempts to identify anomalies issues leveraging on SDN. The intention was to determine the main security issues related to the cloud computing environment to propose an SDN-based approach that allows the network to react in case of an attack. On the other hand, in [72] the authors proposed a Deep Packet

Inspection system for network intrusion detection and prevention using NFV. It was implemented, and it presented reasonable performance. Finally, authors on [14] proposed to detect and mitigate anomalies in SDN, with a statistical approach for detection. A definition of a "normal traffic" profile is the base for the statistical analysis.

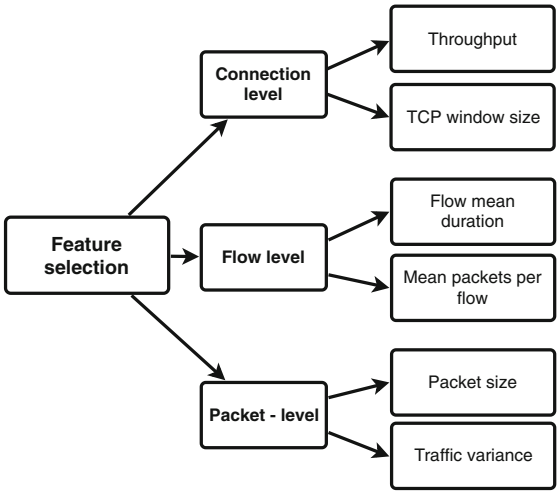

Fig. 3. Features to select in network traffic

At the packet level, the information can be statistical for the network and related to packet size, variance, root mean square. It is useful to characterize traffic in the network, for example with the Hurst parameter H, used to measure the self-similarity and burstiness (the burstier the traffic, the higher H) [37]. Flow and connection level features are most commonly used in SDN, as we will show below. Examples of each level are presented in Fig. 3.

In the sections below, we will present the surveyed papers and a summary in Table 1.

4.1 General Anomaly Detection

Some studies propose general anomaly detection with ML. For example, in [21,22], authors present IDS with deep learning techniques applied to SDN environments. Both studies implemented the IDS as a component of the control plane, instead of deployment as an application. The location allows interacting directly through the network hence protect the controller itself. In [21], they presented a general SDN environment with unsupervised learning. The approach is to use an autoencoder, which has two phases (encoder + decoder) to detect and minimize the reconstruction error for each test sample. The development library was Tensorflow although it is not clear what was the used dataset. The second

study presents a secure framework for IoT based on SDN with a brief review of the security in SDN architecture, but also presents a ML-based IDS. It uses deep learning with a Restricted Boltzmann Machine (RBM). For simulation, the authors focused on the detection model with Tensorflow, and the dataset used was KDD99. The proposed algorithm showed 94% of accuracy.

Authors in [60] present a proposal for both IDS and an action triggered by it: Moving Target Defense. They created a simulated network to obtain data for the training (about 40,000 packets). For the architecture, they presented a neuro-evolutionary model as a light-weight detector that allows real-time operation. To achieve it, they developed two distinctive detectors, one per each type of attack to identify: DDoS and worm. To combine the detectors, authors use Neuro-evolution of Augmenting Topologies (NEAT), an approach to neuro-evolution with crossover context.

4.2 Specific Network Scenarios

There are also proposals for specific network scenarios. That is the case of [73] that presents the implementation of ML-based IDS in optical SDN, and [55] that proposes an scenario of Intelligent Transport Networks. The study in [73] starts by surveying the attacks in control plane and categorize them into unauthorized access, data leakage, data modification, denial of service, and security policy misuse. Since the scenario is optical networks, anomaly detection must consider features related to optical links. Examples are average bandwidth usage, frequent source and destination nodes, average route length, and modulation formats. The possible attacks in this type of network include light-path creation, modification, and deletion (all directed to the link-data layer of the OSI model in optical networks). The first detection methods are point-anomaly-based, as a data instance represented by a point is outside a common region of normal behavior. It uses an algorithm created by the authors to calculate a probability. The second is a sequence-anomaly based method where anomalies occur together as a sequence and use an improved cumulative sum approach. For testing, the authors use NSFNET topology with an owned dataset, and the results present an average detection accuracy of 85%.

On the other hand, [55] presents the cross-fire attack in ITS. The attack consists of a large number of compromised nodes that generate coordinated and low-intensity traffic to disconnect victims (hosts or links) from the network. A ML approach is used to classify the coordinated attacks using three deep learning algorithms: (1) Artificial Neural Networks (ANN); (2) Convolutional Neural Networks (CNN); and (3) Long Short-Term Memory (LSTM) networks. The authors created a testbed in mininet [65] to generate a dataset of their own, with increased traffic for the compromised nodes. They later used this dataset to train and test the model. The results proved the efficiency of the proposal with a slight reduction of performance when the speed of the vehicles increases. A highlight from the study is that it presents the training time and it is about 100 seconds for each algorithm. The short time allows the system to be re-trained as necessary.

4.3 Login Attack

From the surveyed papers, we only found one that addresses login attacks in [46]. The proposal includes defining security rules on the SDN controller to identify and block that type of threat. The study presents the feasibility with the use of four ML techniques: C4.5, BayesNet (BN), Decision Table (DT), and Naive-Bayes (NB). The intention is to give the network the ability to act against a chain of attacks from multiple IP addresses used by each attacker. The used features for the models are attacker IP, attacked host, number of attempts in an attack, and timestamp. The study shows that even a small probability of attack should not be ignored and security rules on the SDN controller must be accordingly modified. For experimentation, the "long tail" dataset was used [23].

4.4 DoS, Probe, U2R and R2L

The studies presented in this section address four kinds of attacks: DoS, Probe, User to Root (U2R), and Remote to local (R2L). The common characteristic between all of them is the dataset used: NSL-KDD [13] that classifies the attacks in the aforementioned categories.

In [62], authors proposed the use of deep neural networks to detect anomalies based on six flow-based features regarded as suitable for SDN: duration, protocol_type, src_bytes, dst_bytes, count and srv_count. The authors trained and tested the model, and compared their proposal with other algorithms such as J48, Naive Bayes (NB), NB Tree, Random Forest (RF), Random Tree (RT), Multilayer Perceptron, and Support Vector Machines (SVM). The paper states the potential of deep learning for the flow-based anomaly detection system. Authors also argue that ML is not fully developed.

In [35], an study of nine ML classifiers with supervised machine learning approaches is presented. They perform tests for accuracy, false alarm rate, precision, recall, f1-measure, the area under the curve (ROC), execution time and Mc Nemar's test. The tests were made with Principal Component Analysis (PCA) for dimensionality reduction with NN, Linear Discriminant Analysis (LDA), DT, RF, Linear SVM, K Nearest-Neighbour (KNN), NB, Extreme Learning Machine (ELM), AdaBoost, RUSBoost, LogitBoost, and BaggingTrees. The results showed that DT, bagging and boosting approaches had better performance than the rest. The selected features were a subset of the features of the dataset, excluding content features.

The same authors proposed in [34] a 5-level hybrid classification system for IDS inspired in the work presented in [9], in a not-SDN network. The paper aims to use flow-statistics provided by the controller to develop a NIDS. The classification methods used are the kNN in the first level, ELM for the second level, and Hierarchical Extreme Learning Machine (H-ELM) for the rest. Each level detects a type of attack using the same features selected in [62]. The system was implemented as a module of POX controller instead of a function of the application plane, for scalability purposes. The approach for selecting these features

is the easiness to get them directly from the controller. The results presented showed improved accuracy, compared to other techniques.

Authors in [53] also place their IDS in the control plane. The technique is a meta-heuristic Bayesian network to classify traffic, and the dataset is NSL-KDD. The proposed process includes a phase of feature selection and extraction to optimize the classifier that consists of the fitness evaluation of the extracted features. It is later fed to the Bayesian classifier. The proposed algorithm is compared with seven other approaches and showed the best overall efficiency for the performance measures with a total of 82.99%.

4.5 Distributed Denial of Service Attacks

Although several of the previous studies consider DoS attacks, they are presented as part of a greater range. In this section, we present studies that investigate specifically DDoS attacks for two reasons. The first one is that a large section of studies for IDS focuses on DDoS attacks. Secondly, with the perspective of the Internet of Things (IoT) and recent threats such as Mirai botnet [28] it is worth to consider the attack individually.

In [68], authors present a specific application for anomaly detection using SDN as a solution to solve scalability challenges. The scenario is a Wireless SDN enable E-Health system. The main feature of this type of network is the massive machine-type communications (mMTC) in which human interaction is minimal. The ML technique used is contrastive pessimistic likelihood estimation (CPLE) for semi-supervised operation with offline training. The intention is to perform online testing to allow running localized detection within the devices to avoid the need to frequently collect network traffics at the controller to update the anomaly detection model. The features used for the classifications are the same defined by [62].

In [11], authors provided an overview of the use of ML for IDS in SDN. The study investigates five ML techniques to mitigate intrusion and DDoS attacks (Neural networks, support vector machine, genetic algorithms, fuzzy logic, Bayesian networks, and decision tree). The authors theoretically analyzed each method and generated a comparison scheme that presents the pros and cons of the techniques. The paper serves as an initial review to select the best approach, according to the needs of the system. However, it does not proposes or test any model.

An analysis of SVM and comparison with other techniques for DDoS detection in SDN is presented in [27]. The paper briefly discussed the types of DDoS attacks and security threats to the controller in SDN. Later, the paper gave four SMV methods and the system description. The 1999 and 1998 DARPA datasets were used for training and testing (about 50/50 ratio), and the technique was compared with RBF, Naive Bayes, Bagging, J48, and Random Forest methods. Accuracy was highest for the proposed SMV with 95%.

In [70], the authors proposed a learning algorithm based on Support Vector Classifier (SVC), leveraged on an Iterative Dichotomiser 3 (ID3) decision tree for feature selection. The model was evaluated in a software testbed with three main components (1) Open vSwitch as a virtual switch, (2) Ryu as the controller, and (3) sFlow Toolkit for data collection. The used dataset is KDD-Cup 1999.

A Dirichlet Process Mixture Model is used in [6], to mitigate DNS-based DDoS attack. Authors used an owned dataset created from the technique to generate them presented in [59].

In [52], authors present an IDS system to identify DDoS attacks. They compare three methods: Naive Bayes, K-Nearest Neighbor (KNN BEST), and Support vector machine (SVM) with an accuracy of 97%, 83%, and 83%, respectively. The features considered as inputs are the number of Packets, Protocol, Delay, Bandwidth, Source IP, and Destination IP. For testing, they use an owned dataset.

In [24], authors present a proposal to improve resiliency in an SDN network, by detecting DoS attacks, specifically SYN flood attack. For classification, the study shows three different techniques: DT, SVM, and NB. The results presented over 99% accuracy, recall, and precision for DT. Dataset KDD 99 is used in the study with the features source IP address, destination IP address, source port, destination port, and protocol. They are later reduced using PCA.

Authors in [48] present an approach to detect and classify DDoS attacks in a cloud environment. For it, they use a two-stage ensemble learning scheme with multivariate Gaussian and Bayesian techniques. The employed features are src_ip, dst_ip, no_of_packets, spoof_dst_ip, blacklist_ip. Although the study is composed of complementary elements to the ML technique, it does not directly try to secure and SDN. Instead, it defines the steps to protect the cloud infrastructure (Virtual machines, orchestrators, etc.).

The previous works were the application of ML techniques for IDS. However, they do not consider implementation issues within the network. In Sect. 5, we present a set of works classified as "frameworks," since they include considerations such as collection and mitigation methods.

4.6 Techniques Comparison

Considering the broad spectrum of cyber-security attacks is noteworthy to have just six specific attacks (DoS, DDoS, Probe, U2R, R2L and login). Even though SDN is an innovative paradigm, we could expect every type of known attack used against an SDN. Also, the research community should prepare to deal with new adapted attacks. It is essential to review how to adapt current techniques to detect, mitigate and prevent different attacks in SDN. Several of the attacks already are recognized using ML techniques applied to them in traditional networks.

Table 1 shows that ML techniques used are very diverse. Most of the papers (9 out of 16) use a single ML technique. The others use at least two methods with one of two approaches: comparison between techniques or combination of them to improve the anomaly detection. Artificial Neural Networks were used in 50%

Table 1. ML techniques proposals for anomaly detection in SDN

Ref.	Detected attack	Detection method	Feature selection	Training dataset
[22]	General anomaly	RBM	41 Features	KDD-Cup 1999
[21]	General anomaly	Autoencoder	41 Features	KDD-Cup 1999
[73]	Optical network	Point anomaly: probability-based. Sequence anomaly: CUSUM	Related to optical links. (e.g. bandwidth, source and destination nodes, route length, and modulation formats)	NSFNET
[60]	DDoS and worm	NEAT	3 packet-level features	Owned: 800000+ packets
[35]	DoS, Probe, U2R, R2L	DT, ELM, NB, LDA, NN, SVM, RT, KNN, AdaBoost, RUSBoost, LogitBoost and BaggingTrees	Subset of features and Principal Components Analysis (PCA) approach	NSL-KDD
[53]	DoS, Probe, U2R, R2L	MHBNC	Preprocessing + feature extraction	NSL-KDD
[62]	DoS, Probe, U2R, R2L	DNN	6-flow-based features	NLS KDD
[34]	DoS, Probe, U2R, R2L	kNN, ELM, and H-ELM for the rest	6-flow-based features	NSL-KDD
[68]	DoS, Probe, R2L and U2R	CPLE	6 features vs 41 features	NSL-KDD
[46]	Login	C4.5, BayesNet, Decision Table (DT), and NB	4-attack-based features	LongTail.
[55]	Crossfire	ANN,CNN,LSTM	3-flow-based features	Owned
[70]	SYN Flood DDoS	SVC	ID3	KDD-Cup 1999
[27]	DDoS	SVM	Grid search method	1999& 1998 DARPA
[52]	DDoS	NB, KNN BEST and SVM	6 fixed features. 6000 data samples	Owned
[48]	DDoS	Ensemble learning with multivariate Gaussian and bayesian	5 flow-based features	Owned
[24]	DDoS	DT, SVM, and NB	4-flow-based features and reduce space withPCA	KDD-Cup 1999

(RBM, NEAT, Generic NN, KNN, ANN, CNN). Another common approach in the reviewed papers is the use of Support Vector Machines. Several articles also presented a Naive Bayes method. However, it was only part of a comparison to other techniques.

Finally, considering feature selection, we found it very diverse. However, in [62] the authors presented a set of six features that were used in four studies, regarded as suitable for SDN. On the other cases, the technique or definition of the features to be included in the ML model was independently selected.

5 IDS Frameworks for SDN

The implementation of the ML techniques for IDS needs to consider articulation with the network environment. That is, define how to collect the data for analysis, as well as mechanisms to activate in case of anomaly detection. For collection, we found three main sources of data to feed the ML model: (1) Statistic collection with OF methods [8,36,38], (2) Getting a copy of the flow table from the switches [45], and (3) With packet-in messages [20,63,67].

Regarding the mitigation, the typical method is to define a module at the control plane (next to the controller) or a dedicated application in the application plane that affects the OF tables of the switches.

In the paragraphs below, we will present the frameworks found in the survey and their main considerations, in contrast to the previous section (studies of the single ML model). The studies are organized regarding the collection method.

5.1 Frameworks Description

Authors in [8] present a system that applies Machine Learning (ML) classification algorithms to detect DDoS attacks. They also propose two defense mechanism for specific SDN attacks: miss-behavior attack and new-flow attack. The first refers to the attack directed to a target using an existing, validated flow. The second exploits the packet-in vulnerability to create a DoS attack. Both are statistical-analysis based. Regarding the DDoS detection mechanism, the system uses a ranker algorithm, a genetic algorithm, and a greedy algorithm for feature selection and Sequential Minimal Optimization (SMO) for classification. The achieved accuracy is 99.40%.

OF statistics are also used in [38] with a 5G scenario implemented with SDN. The study presents Random Forest classifier for feature selection and combines k-means++ with Adaboost for flow classification. The former creates two clusters, which most probably represent the normal and abnormal instances and the later further partition the anomaly clusters into four main classes of attacks. The techniques are part of a complete architecture for ML-based IDS within the SDN scheme. It includes modules in each plane of SDN to allow the collection of data and mitigation action. The ML techniques used are varied and does not evaluate the classification algorithms, but the combination of them with the feature selection techniques. The combinations in the study are RF-KA, RF-GDBT, RF-DT, RF-SVM, Tree-KA, Fisher-KA, and ReliefF-KA. The study presents an analysis of these combinations in an environment that balance the attacks (over-sample the minority intrusion such as R2L, and under-sample

the majority intrusions such as DDoS). For evaluation, the study uses KDD-Cup 1999 [66]. Two relevant conclusions from the study are: (1) Feature selection is critical for better accuracy and lower false rate; and (2) The sampling technique could improve the detection accuracy of minority intrusions dramatically while maintains a reasonable detection rate of the majority ones.

In [36], authors present a framework to use ML for IDS. They propose a NIDS over SDN architecture in which the packets from the switches are captured on a computer with many network cards that act as OpenFlow vSwitch. It sends the Ethernet packet to a Feature Extractor module that analyzes them and extracts 25 features, depending on the transport protocol (TCP, UDP, ICMP). Later the C4.5 algorithm classifies packets for malicious activity. For testing, the authors used the 1999 Darpa dataset [39], and they showed detection of DoS and Probe attacks at high precision. They also proposed and tested a network topology to generate real traffic.

The second type of collection method is to obtain flows from the data plane, using the forwarding.l2_learning Method provided by POX. The technique is used by [45] in combination with an unsupervised RBM algorithm with 92% accuracy. The training method is based on Contrastive Divergence (CD), and the features used for the model are flow-level, and connection-level: total number of packets transmitted (ToP), the ratio of source and destination bytes (RoSD), and connection duration time (CT).

Another technique to collect data is the use of packet-in messages of OF. The method is proposed as part of the framework DaMask in [67]. Even though it is presented as DDoS detection, the study does not present the ML detection technique. According to that, the architecture could be implemented in other types of attacks. The primary goal is to apply DaMask to a cloud computing environment from an enterprise view, which is inherently different than a network. The identified differences are: (1) Control of the computational resources are out of hands of the defender (provider's responsibility); (2) Fast and straightforward resource allocation generates constant topology changes to adapt to; and (3) Network resources are shared with all other users of the cloud, which requires separation mechanisms not considered in traditional DDoS. To answer the requirements, the authors created a three-layer framework (one per each plane in SDN). The system has two main modules (attack detection and attack mitigation) at the application level. For feature selection, they used the Chow–Liu algorithm, and the attack detection is made with a graphical model. The testing was done with the UNB ISCX [59] dataset. As a result of the evaluation, the authors concluded the proposed framework requires little effort from the provider for implementation.

Packet-in detection as a collection method is also used in [20], in combination with a neural network for detection of DDoS attacks. The solution consists of four mechanisms: attack detection trigger, attack detection, attack traceback, and attack mitigation. The study of the detection trigger (when to start the detection process) and traceback (find the source of the attack) are differentiators for this proposal. Similarly to other proposals, [63,67] the authors selected an

abnormal detection of packet-in messages as a trigger to start the detection mechanism (Backpropagation neural network BPNN). It has one input layer (five neurons), one hidden layer (ten neurons) and one output layer (one neuron). On the other hand, the backtracking mechanism seeks for the path followed by the malicious flow by marking the switches, which allows identifying the source. The mitigation method creates new flow entries with the highest priority to drop the traffic directed to the target, and use OpenFlow modification message to clean the flow tables. The study presents the results based on the performance of the detection trigger but not the BPNN classification.

Finally, authors in [63] also use packet-in detection as a collection method and present a Gated Recurrent Unit Recurrent Neural Network as part of a framework for IDS. The detector is implemented as part of the control plane, next to the controller. For this case, the feature srv_count is changed for the dst_host_same_src_port_rate, although they used the same features and dataset of their previous work [62]. The proposal presented low processing impact on the controller and a detection rate of 89%.

5.2 Frameworks Comparison

In Table 2 we present the surveyed frameworks. Only seven (7) out of the studied papers, presented a complete framework to implement in a network. The elements identified in these papers to classify them as frameworks are the description of collection and mitigation methods. They are applied before and after the detection mechanism and provide a clear architecture to deploy the solution in a functioning network.

We identify three types of collection methods: OF statistics, database copy with forwarding.l2-learning command, and packet-in. All of the methods are based on OF possibilities. However, there is diversity in SDN implementation, and it is essential to define other alternatives for other scenarios. An appealing option is sFlow [4], a monitoring tool for packet sampling with an analysis module.

For mitigation, papers [36,45] do not provide a proposal. Frameworks [8,20,38,63] base their technique on the use of OF, with table modification on the data plane. The proposals consider an additional module in the controller to handle the changes.

However, the proposal in [67], DaMask, presents an architecture in which the mitigation is located on the application layer of SDN. That approach would allow some flexibility for the deployment of the design.

Table 2. Framework proposals to use ML in anomaly detection in SDN

Ref.	Collection method	Mitigation method	Detection method	Feature\selection	Metric	Training dataset	Scenario
[8]	OF based, packet recieved counter	Openflow table change from app level	Sequential Minimal Optimization (SMO)	Ranker, genetic, and greedy algorithms	Accuracy: 99.40%	NLS-KDD	SDN
[38]	OF based, regular intervals	Dedicated module to give instructions to switches	(1) Statistical (2) ML techniques (GBDT, DT, SMV, KA)	RF, Tree, Fisher, ReliefF	Accuracy: Probe 97.96% DoS 99.97% U2R 68%, R2L 65.5%	KDD-Cup 1999	5g networks
[36]	OF based, not detailed	Not defined	C4.5	14 derivated from transport features from a basic set of 9	IPS alert: 60%	1999 Darpa	SDN
[45]	Flows from the data plane switches saved in POX database (forwarding.l2-learning)	Not defined	Restricted Boltzmann Machine based	Restricted Boltzman Machine	accuracy was 92%	Owned. Undeclared features	SDN
[63]	Packet-in detection	Openflow table change	Gated Recurrent Unit Recurrent Neural Network (GRU-RNN)	Fixed (six features)	Accuraccy 89%	NSL-KDD	SDN
[20]	Packet-in detection (Abnormal messages trigger ML detection)	Openflow table change from app level	Backpropagation neural network BPNN	Not declared	Time, cpu use and traffic due to the trigger method for detection	Owned. Undeclared features	SDN
[67]	Packet-in detection	Managed from app layer	Graph method (not specified ML)	Chow–Liu algorithm	Detection rate (%) Basic 74.02, Local 86.56, Global 89.30	UNB ISCX	Cloud

6 Complementary Proposals, Datasets and Testbeds

To identify open research problems, as well as the primary tools, we present in this section other ML studies related to security, datasets used from the surveyed studies and used testbeds in the cases a network simulation or emulation was created, that is only for the frameworks.

6.1 Other ML Studies Related to Security

Additional to the use of ML for IDS, we identify other studies to consider. On the first place, we recognize the issue related to adversarial machine learning, which was addressed by Nguyen in [49]. The author presented a cyber kill-chain directed to attack machine learning models. The study provides an analysis of the current use of ML in SDN security as well as attacks directed to ML models such as equation-solving, model inversion, pathfinding, and others. It then presents the cyber kill chain, composed of seven steps: (1) Recon; (2) Weaponization; (3) Delivery; (4) Exploitation; (5) Installation; (6) Command and control; and (7) Action. The paper concludes with four recommendations to use ML in network security: (1) Invest time and effort in the threat models while designing ML solutions; (2) Make the ML model auditable; (3) Follow a secure development process; and (4) Produce an initial operational cost model.

An open, available implementation of ML techniques for IDS is [1]. Authors in [41] perform tests on the platform and concluded that the ML algorithm a large training dataset to reduce the false positives. They also present the possibility to create poisoning attacks to cause miss-classifications.

Additionally, it is important to identify tools that could be used in the analysis of traffic. Studies such as [10,12,25,54,69] present ML-based traffic classifiers to identify applications or flow features in different SDN scenarios. Although the proposals are not specific for security, they might leverage the implementation of security applications.

6.2 Datasets and Testbeds

Regarding datasets, from Tables 2 and 1, we identify a total of six public datasets used on the studies. In Table 3 we present the available datasets (items 1 to 6) and also a type that was created by the researches (item 7). The last two columns of the table indicate how many studies use a particular dataset for Type 1 studies (Sects. 4.1 to 4.5), and the second represents Type 2 studies (frameworks presented in Sect. 5).

It is noteworthy that most of the studies use similar datasets, which could cause the same bias issues in the models. Twelve studies use the KDD-Cup 1999 and NSL-KDD datasets that are 20 and 10 years old respectively. Even though they are used extensively in the research community, it is crucial to consider that attacks become more and more sophisticated every day. Besides the owned datasets, LongTail is the newest, but a single study uses it.

Table 3. Datasets used for ML-based IDS in SDN

Item	Dataset	Year	Studies	
			Type 1	Type 2
1	DARPA 99	1999	1	1
2	KDD-Cup 1999	1999	4	1
3	LongTail	2015	1	0
4	NSFNET topology	NA	1	0
5	NSL-KDD	2009	5	2
6	UNB ISCX	2012	0	1
7	Owned	NA	4	2

Regarding the datasets generated by the authors (classified as owned), standard tools are Mininet, Scapy [5], Distributed Internet TrafficGenerator (D-ITG) and the DDoS attack tool TFN2K.

A common approach for creating datasets is to use the guide provided in [59]. The study presents a systematic approach to develop datasets although it is not focused on SDN.

A more modern methodology is presented in [51]. The paper describes a controlled environment to experiment and create datasets for training supervised ML components and validate supervised and unsupervised solutions. The intention is to fill two gaps: (1) The need for threat data generation; and (2) Lack of new datasets to design, train and validate ML models, instead of the old, overused dataset. That is the case of the NSL-KDD. The proposal is an application of NFV/SDN than ML. It presents, however, the possibility to obtain data to be used in these type of systems.

Table 4. Testbeds used for ML-based IDS in SDN

Framework	Testbed
[8]	Emulation on mininet with pox controller and four OVS switches
[20]	Emulation on mininet with RYU controller and 25 switchES with 200 hosts (2 different computers)
[36]	Network implementation with Opendaylight controller and single computer with many network cards acting as an Openflow vSwitch.
[38]	Not defined
[45]	Emulation on mininet with POX controller and one switch with 5 hosts
[63]	Emulation with Cbench with POX controller
[67]	Emulation on mininet implemented in public cloud (AWS EC2) and extended in a privated cloud with Floodlight controller one switch and two hosts. One of the hosts is a web server

On the other hand, for testing of the complete frameworks, the most common tools was Mininet. Authors also used Open VSwitch [3], and Cbench [2] for emulation, as well as network implementation in the case of [36]. Authors in [36], used public cloud environment AWS EC2 in combination of an emulated private cloud.

We present the description of each testbed in Table 4.

7 Conclusion

We present the state of the art of ML-based SDN security proposals. The classification into ML techniques and frameworks allows identifying that very few attacks are being studied in this context. Considering the broad spectrum of cyber-security, there should be more work on different kind of attacks. Additionally, most of the proposals do not include collection techniques to feed the ML model, or mitigation methods to act after detection. There is a need to define specific schemes to implement the ML techniques in SDN. We also identify the need to use updated and SDN-specific datasets that allow creating models to fit actual networks and current attacks. Finally, we present the typical testbeds for the proposals that include network implementation, where there is no implementation on any operative network. This survey allows scholars to find out new research directions that address open problems in SDN security at different levels. There are also opportunities to involve ML techniques to solve such problems.

We also show in this paper that the use of ML techniques in SDN scenarios is an interesting topic for the research community. However, some aspects receive little attention and could be studied further. One key finding is related to the absence of enough open datasets that can be used to compare new methods. From the networking perspective, there is a lack of a comprehensive attack detection that considers a broad spectrum.

As future work, we want to extend the analysis of the ML techniques used in the reviewed papers with a more detailed study.

References

1. Apache spot. http://spot.incubator.apache.org
2. CTools:CBench - cTuning.org. http://ctuning.org/wiki/index.php/CTools:CBench
3. Open vSwitch. https://www.openvswitch.org/
4. sFlow.org - Making the Network Visible. https://sflow.org/
5. Welcome to Scapy's documentation!—Scapy 2.4.2-dev documentation. https://scapy.readthedocs.io/en/latest/
6. Ahmed, M.E., Kim, H., Park, M.: Mitigating DNS query-based DDoS attacks with machine learning on software-defined networking. In: Proceedings - IEEE Military Communications Conference MILCOM (2017). https://doi.org/10.1109/MILCOM.2017.8170802

7. Ali, S.T., Sivaraman, V., Radford, A., Jha, S.: A survey of securing networks using software defined networking. IEEE Trans. Reliab. **64**(3), 1086–1097 (2015). https://doi.org/10.1109/TR.2015.2421391
8. Alshamrani, A., Chowdhary, A., Pisharody, S., Lu, D., Huang, D.: A defense system for defeating DDoS attacks in SDN based Networks. In: Proceedings of the 15th ACM International Symposium on Mobility Management and Wireless Access - MobiWac 2017, pp. 83–92. ACM Press, New York (2017). https://doi.org/10.1145/3132062.3132074
9. Al-Yaseen, W.L., Othman, Z.A., Nazri, M.Z.A.: Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system. Expert Syst. Appl. **67**, 296–303 (2017). https://doi.org/10.1016/j.eswa.2016.09.041
10. Amaral, P., Dinis, J., Pinto, P., Bernardo, L., Tavares, J., Mamede, H.S.: Machine learning in software defined networks: data collection and traffic classification. In: 2016 IEEE 24th International Conference on Network Protocols (ICNP), pp. 1–5. IEEE, November 2016. https://doi.org/10.1109/ICNP.2016.7785327
11. Ashraf, J., Latif, S.: Handling intrusion and DDoS attacks in software defined networks using machine learning techniques. In: 2014 National Software Engineering Conference, pp. 55–60. IEEE, November 2014. https://doi.org/10.1109/NSEC.2014.6998241
12. Bakhshi, T.: Multi-feature enterprise traffic characterization in openflow-based software defined networks. In: 2017 International Conference on Frontiers of Information Technology (FIT), pp. 23–28. IEEE, December 2017. https://doi.org/10.1109/FIT.2017.00012. http://ieeexplore.ieee.org/document/8261006/
13. Canadian Institute for Cybersecurity: NSL-KDD Datasets. https://www.unb.ca/cic/datasets/nsl.html
14. Carvalo, L.F., Abrao, T., de Souza Mendes, L., Proença, M.L.: An ecosystem for anomaly detection and mitigation in software-defined networking. Expert Syst. Appl. **104**, 121–133 (2018). https://doi.org/10.1016/j.eswa.2018.03.027
15. Paper, N.W.: Network functions virtualisation: an introduction, benefits, enablers, challenges & call for action. Issue 1 (Technical report, ETSI) (2012)
16. Chowdhary, A., Pisharody, S., Huang, D.: SDN based Scalable MTD solution in cloud network. In: Proceedings of the 2016 ACM Workshop on Moving Target Defense - MTD 2016, pp. 27–36. ACM Press, New York (2016). https://doi.org/10.1145/2995272.2995274
17. Chung, C.J., Xing, T., Huang, D., Medhi, D., Trivedi, K.: SeReNe: on establishing secure and resilient networking services for an SDN-based multi-tenant datacenter environment. In: 2015 IEEE International Conference on Dependable Systems and Networks Workshops, pp. 4–11. IEEE, June 2015. https://doi.org/10.1109/DSN-W.2015.25. http://ieeexplore.ieee.org/document/7272544/
18. Clark, D.D., Partridge, C., Ramming, J.C., Wroclawski, J.T.: A knowledge plane for the internet. In: Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications - SIGCOMM 2003, p. 3. ACM Press, New York (2003). https://doi.org/10.1145/863955.863957
19. Coughlin, M.: A survey of SDN security research. Technical report. http://ngn.cs.colorado.edu/~coughlin/doc/a_survey_of_sdn_security_research.pdf
20. Cui, Y., et al.: SD-Anti-DDoS: fast and efficient DDoS defense in software-defined networks. J. Netw. Comput. Appl. **68**, 65–79 (2016). https://doi.org/10.1016/J.JNCA.2016.04.005. https://www-sciencedirect-com.ezproxy.unal.edu.co/science/article/pii/S1084804516300480

21. Dawoud, A., Shahristani, S., Raun, C.: A deep learning framework to enhance software defined networks security. In: 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp. 709–714. IEEE, May 2018. https://doi.org/10.1109/WAINA.2018.00172. https://ieeexplore.ieee.org/document/8418157/
22. Dawoud, A., Shahristani, S., Raun, C.: Deep learning and software-defined networks: towards secure IoT architecture. Internet Things **3–4**, 82–89 (2018). https://doi.org/10.1016/J.IOT.2018.09.003. https://www.sciencedirect.com/science/article/pii/S2542660518300593
23. Eric Wedaa: LongTail (2015). http://longtail.it.marist.edu/honey/dashboard.shtml
24. Gangadhar, S., Sterbenz, J.P.G.: Machine learning aided traffic tolerance to improve resilience for software defined networks, pp. 1–7 (2017)
25. He, L., Xu, C., Luo, Y.: vTC. In: Proceedings of the 2016 ACM International Workshop on Security in Software Defined Networks and Network Function Virtualization - SDN-NFV Security 2016, pp. 53–56. ACM Press, New York (2016). https://doi.org/10.1145/2876019.2876029
26. Kloti, R., Kotronis, V., Smith, P.: Openflow: a security analysis. In: 2013 21st IEEE International Conference on Network Protocols (ICNP), pp. 1–6. IEEE (2013)
27. Kokila, R.T., Thamarai Selvi, S., Govindarajan, K.: DDoS detection and analysis in SDN-based environment using support vector machine classifier. In: 6th International Conference on Advanced Computing, ICoAC 2014 (2015). https://doi.org/10.1109/ICoAC.2014.7229711
28. Kolias, C., Kambourakis, G., Stavrou, A., Voas, J.: Ddos in the IoT: mirai and other botnets. Computer **50**(7), 80–84 (2017). https://doi.org/10.1109/MC.2017.201
29. Koning, R., de Graaff, B., Polevoy, G., Meijer, R., de Laat, C., Grosso, P.: Measuring the efficiency of SDN mitigations against attacks on computer infrastructures. Future Gener. Comput. Syst. **91**(1), 144–156 (2019). https://doi.org/10.1016/j.future.2018.08.011
30. Koponen, T., et al.: Onix: a distributed control platform for large-scale production networks. In: Proceedinds of the 9th USENIX Conference on Operating Systems Design and Implementation, vol. 16, no, 2, pp. 133–169 (2010). https://dl.acm.org/citation.cfm?id=279229
31. Kreutz, D., Ramos, F.M., Verissimo, P.: Towards secure and dependable software-defined networks. In: Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking - HotSDN 2013, p. 55. ACM Press, New York (2013). https://doi.org/10.1145/2491185.2491199
32. Kwon, D., et al.: A survey of deep learning-based network anomaly detection. Cluster Comput. https://doi.org/10.1007/s10586-017-1117-8
33. Lamport, L.: The part-time parliament. ACM Trans. Comput. Syst. (TOCS) **16**, 133–169 (1998). https://doi.org/10.1145/279227.279229
34. Latah, M., Toker, L.: An efficient flow-based multi-level hybrid intrusion detection system for software-defined networks. CoRR, June 2018. http://arxiv.org/abs/1806.03875
35. Latah, M., Toker, L.: Towards an efficient anomaly-based intrusion detection for software-defined networks. CoRR, March 2018. http://arxiv.org/abs/1803.06762
36. Le, A., Dinh, P., Le, H., Tran, N.C.: Flexible network-based intrusion detection and prevention system on software-defined networks. In: 2015 International Conference on Advanced Computing and Applications (ACOMP), pp. 106–111. IEEE (2015)

37. Leland, W.E., Willinger, W., Taqqu, M.S., Wilson, D.V.: On the self-similar nature of ethernet traffic. ACM SIGCOMM Comput. Commun. Rev. **25**(1), 202–213 (2004). https://doi.org/10.1145/205447.205464
38. Li, J., Zhao, Z., Li, R.: A machine learning based intrusion detection system for software defined 5G network. CoRR, July 2017. http://arxiv.org/abs/1708.04571
39. Lincoln Laboratory, Massachusetts Institute of Technology: 1999 DARPA Intrusion Detection Evaluation Dataset—MIT Lincoln Laboratory (1999). https://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset
40. Marotta, A., Carrozza, G., Avallone, S., Manetti, V.: An OpenFlow-based architecture for IaaS security. In: Proceedings of the 3rd International Conference on Application and Theory of Automation in Command and Control Systems - ATACCS 2013, p. 118. ACM Press, New York (2013). https://doi.org/10.1145/2494493.2494510
41. Mathas, C.M., et al.: Evaluation of Apache Spot's machine learning capabilities in an SDN/NFV enabled environment. In: Proceedings of the 13th International Conference on Availability, Reliability and Security - ARES 2018, pp. 1–10. ACM Press, New York (2018). https://doi.org/10.1145/3230833.3233278
42. Mckeown, N., Anderson, T., Peterson, L., Rexford, J., Shenker, S., Louis, S.: OpenFlow: enabling innovation in campus networks. ACM SIGCOMM Comput. Commun. Rev. **38**(2), 69–74 (2008). http://ccr.sigcomm.org/online/files/p69-v38n2n-mckeown.pdf
43. Jain, S., et al.: B4: Experience with a globally-deployed software defined WAN. ACM SIGCOMM Comput. Commun. Rev. **43**(4), 3–14 (2013). https://doi.org/10.1145/2534169.2486019
44. Mestres, A., et al.: Knowledge-defined networking. ACM SIGCOMM Comput. Commun. Rev. **47**(3), 4–10 (2016). https://doi.org/10.1145/3138808.3138810
45. Mohanapriya, P., Shalinie, S.M.: Restricted Boltzmann machine based detection system for DDoS attack in software defined networks. In: 2017 4th International Conference on Signal Processing, Communication and Networking, ICSCN 2017, pp. 14–19 (2017). https://doi.org/10.1109/ICSCN.2017.8085731
46. Nanda, S., Zafari, F., DeCusatis, C., Wedaa, E., Yang, B.: Predicting network attack patterns in SDN using machine learning approach. In: 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), pp. 167–172. IEEE, November 2016. https://doi.org/10.1109/NFV-SDN.2016.7919493
47. Navid, W., Bhutta, M.N.M.: Detection and mitigation of denial of service (DoS) attacks using performance aware software defined networking (SDN). In: 2017 International Conference on Information and Communication Technologies (ICICT), pp. 47–57. IEEE, December 2017. https://doi.org/10.1109/ICICT.2017.8320164
48. Neupane, R.L., et al.: Dolus. In: Proceedings of the 19th International Conference on Distributed Computing and Networking - ICDCN 2018, pp. 1–10. ACM Press, New York (2018). https://doi.org/10.1145/3154273.3154346
49. Nguyen, T.N.: The challenges in SDN/ML based network security: a survey. CoRR abs/1804-0, April 2018. https://doi.org/10.1109/CSNET.2018.8602680. http://arxiv.org/abs/1804.03539
50. Pan, J., Yang, Z.: Cybersecurity challenges and opportunities in the new "edge computing + IoT" world. In: Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks and Network Function Virtualization - SDN-NFV Sec 2018, pp. 29–32. ACM Press, New York (2018). https://doi.org/10.1145/3180465.3180470

51. Pastor, A., Mozo, A., Lopez, D.R., Folgueira, J., Kapodistria, A.: The Mouseworld, a security traffic analysis lab based on NFV/SDN. In: Proceedings of the 13th International Conference on Availability, Reliability and Security - ARES 2018, pp. 1–6. ACM Press, New York (2018). https://doi.org/10.1145/3230833.3233283
52. Prakash, A., Priyadarshini, R.: An intelligent software defined network controller for preventing distributed denial of service attack. In: 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), pp. 585–589. IEEE, April 2018. https://doi.org/10.1109/ICICCT.2018.8473340
53. Prasath, M.K., Perumal, B.: A meta-heuristic Bayesian network classification for intrusion detection. Int. J. Netw. Manag. **29**, e2047 (2018). https://doi.org/10.1002/nem.2047
54. Qazi, Z.A., et al.: Application-awareness in SDN. ACM SIGCOMM Comput. Commun. Rev. **43**, 487–488 (2013). https://doi.org/10.1145/2534169.2491700
55. Raj, A., Truong-Huu, T., Mohan, P.M., Gurusamy, M.: Crossfire attack detection using deep learning in software defined ITS networks. CoRR, December 2018. http://arxiv.org/abs/1812.03639
56. Rawat, D.B., Reddy, S.R.: Software defined networking architecture, security and energy efficiency: a survey. IEEE Commun. Surv. Tutor. **19**(1), 325–346 (2017). https://doi.org/10.1109/COMST.2016.2618874
57. Scott-Hayward, S., Natarajan, S., Sezer, S.: Survey of security in software defined networks. Surv. Tutor. 18(1), 623–654 (2016). https://doi.org/10.1109/COMST.2015.2474118. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7150550
58. Shin, S., Gu, G.: Attacking software-defined networks. In: Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking - HotSDN 2013, p. 165. ACM Press, New York (2013). https://doi.org/10.1145/2491185.2491220
59. Shiravi, A., Shiravi, H., Tavallaee, M., Ghorbani, A.A.: Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Comput. Secur. **31**(3), 357–374 (2012). https://doi.org/10.1016/J.COSE.2011.12.012. https://www.sciencedirect.com/science/article/pii/S0167404811001672
60. Smith, R.J., Zincir-Heywood, A.N., Heywood, M.I., Jacobs, J.T.: Initiating a moving target network defense with a real-time neuro-evolutionary detector. In: Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion - GECCO 2016 Companion, pp. 1095–1102. ACM Press, New York (2016). https://doi.org/10.1145/2908961.2931681
61. Sultana, N., Chilamkurti, N., Peng, W., Alhadad, R.: Survey on SDN based network intrusion detection system using machine learning approaches. Peer-to-Peer Netw. Appl. **12**, 1–9 (2018). https://doi.org/10.1007/s12083-017-0630-0
62. Tang, T.A., Mhamdi, L., McLernon, D., Zaidi, S.A.R., Ghogho, M.: 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. IEEE, October 2016. https://doi.org/10.1109/WINCOM.2016.7777224
63. Tang, T.A., Mhamdi, L., McLernon, D., Zaidi, S.A.R., Ghogho, M.: Deep recurrent neural network for intrusion detection in SDN-based networks. In: 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), pp. 202–206. IEEE, June 2018. https://doi.org/10.1109/NETSOFT.2018.8460090

64. Tantar, E., Palattella, M.R., Avanesov, T., Kantor, M., Engel, T.: Cognition: a tool for reinforcing security in software defined networks. In: Tantar, A.-A., et al. (eds.) EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V. AISC, vol. 288, pp. 61–78. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07494-8_6
65. Mininet Team: Mininet: an instant virtual network on your laptop (or other PC) - Mininet (2012). http://mininet.org/
66. University of California, Irvine: KDD Cup 1999 Data (1999). http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
67. Wang, B., Zheng, Y., Lou, W., Hou, Y.T.: DDoS attack protection in the era of cloud computing and software-defined networking. Comput. Netw. **81**, 308–319 (2015). https://doi.org/10.1016/J.COMNET.2015.02.026. https://www.sciencedirect.com/science/article/pii/S1389128615000742
68. Wang, B., Sun, Y., Yuan, C., Xu, X.: LESLA - a smart solution for SDN-enabled mMTC E-health monitoring system. In: Proceedings of the 8th ACM MobiHoc 2018 Workshop on Pervasive Wireless Healthcare Workshop - MobileHealth 2018, pp. 1–6. ACM Press, New York (2018). https://doi.org/10.1145/3220127.3220128
69. Wang, P., Ye, F., Chen, X., Qian, Y.: Datanet: deep learning based encrypted network traffic classification in SDN home gateway. IEEE Access **6**, 55380–55391 (2018). https://doi.org/10.1109/ACCESS.2018.2872430
70. Wang, P., Chao, K.M., Lin, H.C., Lin, W.H., Lo, C.C.: An efficient flow control approach for SDN-based network threat detection and migration using support vector machine. In: Proceedings - 13th IEEE International Conference on E-Business Engineering, ICEBE 2016 - Including 12th Workshop on Service-Oriented Applications, Integration and Collaboration, SOAIC 2016, pp. 56–63 (2017). https://doi.org/10.1109/ICEBE.2016.020
71. Yan, Q., Yu, F.R., Gong, Q., Li, J.: 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 (2016). https://doi.org/10.1109/COMST.2015.2487361
72. Yasrebi, P., Monfared, S., Bannazadeh, H., Leon-Garcia, A.: Security function virtualization in software defined infrastructure. In: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), pp. 778–781. IEEE, May 2015. https://doi.org/10.1109/INM.2015.7140374
73. Zhang, H., Wang, Y., Chen, H., Zhao, Y., Zhang, J.: Exploring machine-learning-based control plane intrusion detection techniques in software defined optical networks. Opt. Fiber Technol. **39**, 37–42 (2017). https://doi.org/10.1016/J.YOFTE.2017.09.023. https://www-sciencedirect-com.ezproxy.unal.edu.co/science/article/pii/S1068520017303644