Computer Systems
Science and Engineering

Tech Science Press

DOI: 10.32604/csse.2023.039893
Article

Toward Secure Software-Defined Networks Using Machine Learning: A Review, Research Challenges, and Future Directions

Muhammad Waqas Nadeem[1,*], Hock Guan Goh[1], Yichiet Aun[1] and Vasaki Ponnusamy[2]

[1]Faculty of Information and Communication Technology, Universiti Tunku Abdul Rahman, Kampar, Perak, Malaysia
[2]Higher Colleges of Technology, Fujairah Men's Campus, Fujairah, UAE

*Corresponding Author: Muhammad Waqas Nadeem. Email: waqasnadeem@1utar.my
Received: 22 February 2023; Accepted: 09 May 2023; Published: 28 July 2023

Abstract: Over the past few years, rapid advancements in the internet and communication technologies have led to increasingly intricate and diverse networking systems. As a result, greater intelligence is necessary to effectively manage, optimize, and maintain these systems. Due to their distributed nature, machine learning models are challenging to deploy in traditional networks. However, Software-Defined Networking (SDN) presents an opportunity to integrate intelligence into networks by offering a programmable architecture that separates data and control planes. SDN provides a centralized network view and allows for dynamic updates of flow rules and software-based traffic analysis. While the programmable nature of SDN makes it easier to deploy machine learning techniques, the centralized control logic also makes it vulnerable to cyberattacks. To address these issues, recent research has focused on developing powerful machine-learning methods for detecting and mitigating attacks in SDN environments. This paper highlighted the countermeasures for cyberattacks on SDN and how current machine learning-based solutions can overcome these emerging issues. We also discuss the pros and cons of using machine learning algorithms for detecting and mitigating these attacks. Finally, we highlighted research issues, gaps, and challenges in developing machine learning-based solutions to secure the SDN controller, to help the research and network community to develop more robust and reliable solutions.

Keywords: Botnet attack; deep learning; distributed denial of service; machine learning; network security; software-defined network

1 Introduction

The development of the Internet is rapidly growing, and the limitations of traditional networks have been explored. The emerging issues of the conventional networks can be solved by patching the network, which makes the network more bloated and the control ability of the network becomes weaker. The invention of Software Defined Networking (SDN) [1,2] has resolved these problems by decoupling the data and control planes. SDN became famous among the network community due to

its novel architecture and can fulfill the demands of fast-growing networks. The resulting decoupling enables the network to be easily managed and organized. In SDN, a centralized controller acts as a Network Operating System (NOS) to manage network resources. The controller is programmable, dynamically programming the entire network and possessing the network's global view by accumulating and monitoring real-time network data and packet information.

This programmability nature of SDN opens ways to implement efficient machine-learning techniques for several reasons. Firstly, the recent developments in computing technologies, such as Tensor Processing Units (TPUs) and Graphics Processing Units (GPUs), have enabled the application of machine learning techniques, including convolutional neural networks and deep neural networks, in the network field [1]. Secondly, the SDN controller's global view allows it to collect different types of network data, making it well-suited for data-driven machine-learning algorithms. Thirdly, machine learning techniques can analyze real-time and historical network data to provide intelligent network services, optimize network performance, and automate network services. Fourthly, machine learning-based Network Intrusion Detection Systems (NIDS) can be deployed in the controller to improve network security. Finally, machine learning algorithms can execute optimal network solutions in real-time, such as configuration and resource allocation [2].

Although SDN has significant capabilities and an architectural framework for modernizing future networks, it is still confronted with diverse security concerns. For instance, the SDN controller is a central point of attack, and an attacker who targets it can bring down the entire network [3]. Additionally, attackers can seize the SDN network by attacking the control plane, making SDN a relatively unknown field among the security community [4]. The number of SDN research papers on security is considerably fewer than those focused on enhancing network performance. Nonetheless, in recent years, researchers' interest has increased regarding SDN's security issues [5–7]. Therefore, the primary objective of this paper is to provide a comprehensive review of the current SDN security concerns and examine possible attacks on SDN controllers. The study also covers machine learning-based advancements to identify and mitigate these attacks on the controller side.

In summary, the main contributions of this paper are as follows.

- We investigate and highlight the security issues and four possible attacks on the SDN controller.
- We comprehensively summarize the latest machine learning advancements in detecting and mitigating four potential SDN controller attacks, including Distributed Denial of Services (DDOS), botnet attacks, saturation attacks, and ransomware attacks.
- We highlight the successful contribution of machine learning to secure the SDN controller.
- Finally, we discussed the open research challenges and future directions based on the current literature to help the researchers interested in conducting further research in this area.

2 Security Issues in SDN

This section presents a concise overview of potential threats to the Software Defined Networking (SDN) domain, emphasizing security vulnerabilities that may arise in the SDN controller and their impact on the entire network. The SDN environment is susceptible to various security threats, which attackers may exploit to compromise the system. These threats can differ from those encountered in traditional networks, such as attacks on the SDN controller through the generation of malicious traffic flow or control plane communication attacks. Establishing network traffic flows can also be leveraged by various hosts to create attacks. The controller can implement multiple measures to mitigate these security risks, such as monitoring resource utilization, applying permission structures to applications, and containing processes.

A centralized controller can be vulnerable to two types of attacks. Firstly, a malformed OpenFlow header can cause the controller to crash. Secondly, if an attacker frequently sends flow requests, it may result in degraded overall performance and prevent legitimate requests from servicing. In a distributed SDN control architecture, the authentication of controller instances through verification and validation protocols is essential, among other security issues. In a hybrid SDN control plane, DoS and impersonation attacks are common due to the network's combination of two control planes and multiple devices. Security protocols are also necessary at the SDN interfaces to prevent integrity threats.

The centralized network control of the SDN environment increases its susceptibility to security risks. In terms of security, malicious actors may focus their attacks on a singular point in the network, namely the controller, with the intent of manipulating or causing harm to the entire system. Furthermore, the SDN controller is vulnerable to being compromised in three ways: through malicious software bugs or errors, threats from applications running on or inside the controller, and risks posed by the underlying network devices, such as OpenFlow switches.

The controller and forwarding network devices can be vulnerable to different types of attacks, including flooding attacks that can cause the controller to crash. These vulnerabilities can enable attackers to launch various attacks, including Distributed Denial of Service (DDoS), saturation attacks, botnet, and ransomware attacks. The subsequent sections provide further details on these attacks and the development of machine learning techniques for their detection and mitigation in SDN.

3 Machine Learning in SDN

The SDN controller provides a centralized network view and facilitates control and management. Leveraging machine learning algorithms and techniques can introduce intelligence into the controller, enabling it to analyze network data, optimize network performance, and automatically provide network services. With this learning capability, the SDN controller can make optimum decisions to adapt to the changes in the network environment. This section offers a comprehensive review of recent machine learning developments to address security concerns in SDN. It also discusses how real-time machine-learning algorithms are employed in this context [1,2].

3.1 Distributed Denial of Service Attacks Detection and Mitigation

A DDoS attack overwhelms a network by sending excessive traffic that consumes network resources, rendering the server inaccessible to legitimate traffic and causing the network to become unreliable. DDoS attacks can be classified into protocol-exploitation, volumetric, and application-layer attacks [8]. The decision-making process in a network based on SDN architecture is handled by the control plane, which can make intelligent decisions. Still, it presents a Single Point of Failure (SPF) vulnerability since attackers who access the controller can damage the entire network infrastructure [9]. SDN uses southbound protocols, such as OpenFlow, to take action against flow table entries, which contain various fields for specific tasks, such as timeouts, priorities, action fields, and counters [10]. Malicious flow rules can be inserted into the flow tables, causing a DDoS attack on the controller. Fig. 1 shows the general procedure of a DDoS attack in an SDN environment. Machine learning techniques are being developed to detect and mitigate DDoS attacks in the SDN controller, and the next section describes these techniques in detail. In the context of SDN, many DDoS attack detection and mitigation methods are adaptations of techniques used in traditional networks. Among these, the most popular methods for detecting DDoS attacks are based on statistical information entropy algorithms. These methods offer the advantage of rapidly processing traffic data with minimal

computational costs. However, their accuracy relies heavily on choosing an appropriate threshold, which can be limiting and result in one-sided detection.

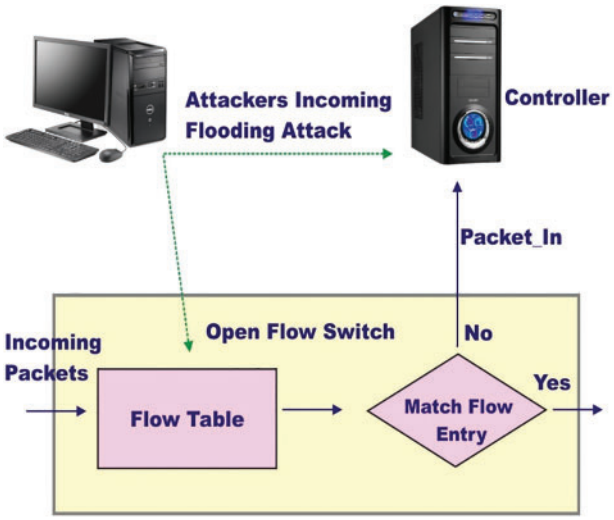

Figure 1: The general procedure of a DDoS attack on both control and forwarding planes

Reference [11] proposed a Joint Scoring System (JESS) based on entropy for detecting and mitigating DDoS attacks, which utilizes the joint entropy tool without increasing the workload of switches. Meanwhile, another study [12] used traffic entropy statistical analysis to protect networks from DDoS attacks. They validate their model in the Mininet emulator. Reference [13] also developed a technique to identify and counteract SYN flooding attacks in SDN using destination IP address entropy and specific TCP flags as random variables. To determine the attacker, an adaptive threshold is employed. In [14], the authors proposed a lightweight algorithm for detecting DDoS attacks that utilizes traffic features. This algorithm processes switch data using the NOX controller and performs traffic analysis using the unsupervised learning algorithm Self-Organizing Map (SOM) and competitive learning of ANN. KNN is a simple and effective machine learning algorithm that calculates the abstract distance between traffic feature vectors to classify flows. Furthermore, [15] proposed a set of finer-grained flow indices that extract nine single and 39 dual attributes from various dimensions such as category, time, space, and intensity. This subset of fine-grained traffic features successively improves the detection accuracy for attacks.

Reference [16] introduced a modular detection system based on K-Means++ and Fast K-Nearest Neighbors (K-FKNN), which enhances the controller's accuracy, efficiency, and stability against DDoS attacks. The paper [17] proposed a new framework that employs a trigger mechanism to work with detection and defense methods in data and control planes to respond quickly to DDoS attacks and ease the workload on switches and controllers. Reference [18] presented a Hidden Markov Models-based detection system for the SDN environment that uses the Baum-Welch algorithm for training. They utilize the Mininet emulator and OpenFlow Floodlight controller to carry out their experimental setup. Reference [19] proposed a new technique to tackle NetFlow and Misbehavior attacks. Their system periodically collects network information and applies ML techniques to classify network flow as normal or attack traffic. Reference [20] developed an entropy and SVM-based flooding attack detection and mitigation system that collects traffic information using an SDN controller and sFlow agents. They also introduce a mitigation agent to block attack traffic while still allowing network

resources to be accessed by legitimate users. Reference [21] proposed a KNN and φ-entropy-based hybrid system, where the φ-entropy is used for feature selection, and KNN is used for classification. Reference [22] introduced an ensemble method that detects anomalous network traffic behavior toward the SDN controller using algorithms such as SVM, KNN, NB, and SOMs.

Reference [23] proposed a back propagation neural network-based method for extracting the temporal behavior of an attack. The back propagation neural network is trained to extract attack patterns, which are used to identify the network attacks. Once the detection module detects the attack, it refers to the defense module, which blocks the port from where the malicious packets are coming. However, it also blocks the ports for the legitimate user, and then the port recovery module is used to recover the ports for all legitimate users. Reference [24] suggested a deep learning-based DDoS detection mechanism in SDN. In this method, the raw data samples are preprocessed by the feature processing units, and then the processed data set is used by the deep learning algorithm for training. After that, the Information Statistics Module (ISM) provides the network statistics to the Flow Table Generator module, which determines the priorities for the different attacks based on flow entries, drops the attack packets, and sends feedback to the OpenFlow switch. The authors used the ISCX dataset for training and verified the defense module using a real-time DDoS attack, and for validation, they used a hardware-based experimental setup. Reference [25] proposed a system that combines neural networks with statistical methods to identify abnormal network behavior. This approach uses an entropy metric to select essential features from a set of features and then applies a Self-Organizing Map (SOM) to classify network behavior. The proposed approach is validated using the POX controller on the Mininet emulator. In another study, [26] introduced a hybrid system for detecting and mitigating Port scan and DDoS attacks, which involves three phases: characterization, attack detection, and mitigation. The authors used an entropy metric to quantify network features and Long Short-Term Memory (LSTM) to learn the attributes of normal network traffic. Finally, they used fuzzy logic to detect attacks on the network. Their approach was simulated on the Mininet emulator using a Floodlight controller.

Another approach is based on SVM assisted by Genetic Algorithm (GA) and Kernel Principal Component Analysis (KPCA), which is proposed in [27]. The authors used KPCA to reduce the dimension of feature vectors and then optimized the parameters of SVM through GA. To reduce noise caused by feature differences, they proposed an improved kernel function (N-RBF). The proposed model was installed into the controller to define different security rules to detect the attack, and a DDoS mitigation module was designed separately inside the controller. The POX controller on the Mininet emulator was used to validate this approach. In the study, [28] proposed a DSM-based SVM algorithm for DDoS attack detection and mitigation, which involved pre-processing the input data, feature extraction using the MCA algorithm, and attack prediction with the DSM-SVM. The mitigation server started to drop attack traffic and absorb the rest of the normal traffic following the detection of an attack. The authors trained their machine learning algorithm using the KDD dataset and deployed it into the RYU controller (i.e., Mininet emulator) for real-time evaluation.

Another study [29] proposed an architecture for detecting DoS/DDoS attacks on the SDN application and transport layers, consisting of four modules: flow collector, preprocessing, attack detection, and flow manager. The CICFlowMeter application was used in the flow collector module to collect and generate flows. PCA was employed in the preprocessing module to reduce the dimension of the flow features. The detection module used pre-trained machine learning models, including KNN, SVM, RF, MLP, Convolutional Neural Network (CNN), Gated Recurrent Units (GRU), and Long Short-Term Memory (LSTM), to classify the input flows as normal or suspicious. The Flow manager module then sent information regarding the suspicious flows to the controller for further

action to create and visualize flow logs for the classification received from the detection module. The approach was validated using the ONOS controller in the Mininet emulator. An SVM-based Intrusion Detection System (IDS) for detecting DDoS attacks in SDN was proposed by [30], where traffic information is received by SVM and then classified as normal or attack traffic. Reference [31] presented a model for mitigating different DDoS attacks by introducing adaptive polling, sFlow-based sampling, deep learning, and a Snort Intrusion Detection System (SIDS). In this model, adaptive polling and sFlow-based sampling are individually deployed in the data plane to reduce network overhead. In contrast, the Stacked Autoencoder (SAE) and SIDS are deployed in the control plane to optimize detection accuracy. Finally, to overcome the challenge of detecting adversarial attacks caused by specific perturbations, a system for detecting DDoS attacks was proposed by [32], which utilizes a Generative Adversarial Network (GAN) and applies adversarial training to make the system less sensitive to malicious attacks. The system can activate the detection system and respond in real-time by continuously monitoring traffic through IP flow analysis. The system constantly monitors traffic using IP flow analysis and activates the detection system to act in real-time. The GAN module detects the attack, and the mitigation module is activated automatically, taking countermeasures to minimize the attack's effect. Their mitigation approach is based on Event-Condition Action (ECA), where the Event is associated with a set of specific rules for the anomaly, the Condition describes the rules where a particular event occurs, and the Action comprises countermeasures taken against the anomaly event. Their method was evaluated using the CISDoS 2019 dataset.

A method incorporating a generalized entropy approach, Particle Swarm Optimization (PSO), and Back Propagation Neural Network (BPNN) were proposed by [33]. The generalized entropy method is initially employed on the switch to identify attacks and categorize traffic into normal and abnormal flows. This triggers the switch to issue abnormal alerts while the PSO-BPNN installed in the controller extracts features from the abnormal traffic flow to predict potential DDoS attacks. The extracted features include average packets per flow, bytes per packet, percentage of pair flow, rate of flow entries, the entropy of source IP address, and average duration per flow. The effectiveness of this approach was validated using the Floodlight controller in the Mininet emulator. In a similar vein, [34] proposed a DDoS detection method that combines Information Entropy (IE) and Convolutional Neural Networks (CNNs). Their approach utilizes the IE to examine suspicious traffic flows, and then the CNN performs fine-grained packet-based detection to distinguish between normal and abnormal traffic. They used the CICIDS2017 dataset to train the CNN, and the method's performance was evaluated in real-time using the POX controller in the Mininet emulator.

Furthermore, the authors employed edge computing to offload the detection task from the control plane to the data plane, reducing the southbound communication overhead and controller burden for attack detection. Using cloud-edge collaboration, [35] designed a DDoS detection system based on Entropy-Measuring (EM), SOM, and K-Dimensional tree (EMSOM-KD). The authors selected Ideal SOM maps using EM and identified most traffic flows directly by the EMSOM. In [36], the authors proposed a hybrid approach for detecting DDoS attacks, where an Information Entropy (IE) based module performs initial detection for anomalous traffic, followed by a second detection module based on Stacked Sparse Autoencoder (SSA)-SVM that confirms the suspected abnormal traffic. After successfully detecting an attack, their defense module issues a new flow table to restore the normal communication of the network timely. A Spatial-Temporal Graph Convolutional Network (ST-GCN) based detection method was proposed in a different study [37], which maps the network into a graph and uses IN-band Network Telemetry (INT) to sense the state of switches with samples. An ensemble model based on optimized weighted voting for DDoS attack detection in SDN was introduced in another study [38], which used different hyper-parameter values of six base classifiers to

build the ensemble model. ML-based solutions are discussed in the section, and their pros and cons are compared in Table 1.

Table 1: ML-based solutions for DDoS detection and mitigation in SDN

Study	Controller type	Protocols	Classifier type	Data set	Evaluation	Discussion
[14]	NOX	TCP, UDP, and ICMP	Self-organizing map (SOM)	Custom developed	Detection rate = 99.11%	The limitation of this work is that it cannot identify the ports of the OF switches from where the attack is launched.
[17]	ONOS	Not mentioned	KNN and K-Means	NSL-KDD	Accuracy = 98.85%	This work is not dealing with when the controller is under large-scale network traffic.
[18]	Floodlight	Not mentioned	Hidden Markov Models (HMMs)	Custom developed	Accuracy = 96%	Using feature vectors along with HMM improves the detection power of the proposed approach.
[19]	POX	Not mentioned	Sequential minimal optimization (SMO)	NSL-KDD	Accuracy = 99.40%	This work is not efficient in detecting unknown attacks.
[22]	POX	TCP, UDP and ICMP	SVM, KNN, naïve bayes (NB), and self-organizing maps (SOMs)	CAIDA 2016	Accuracy of SVM-SOM = 98.12%	This study used an old dataset version and may be extended by using new datasets.
[23]	Not mentioned	Not mentioned	Back propagation neural network (BPNN)	DARPA 1999	Not mentioned	The achieved accuracy of the method is not mentioned. This method is effective in performing a port recovery after an attack.
[24]	Not mentioned	TCP, UDP, HTTP	Bidirectional recurrent neural network (BRNN)	ISCX2012	Accuracy = 99%	The significant advantage of this study is that it can help to reduce the degree of dependence on the software and hardware environments. It simplifies the updation of detection systems in real-time.

(Continued)

Table 1 (continued)

Study	Controller type	Protocols	Classifier type	Data set	Evaluation	Discussion
[25]	POX	Not mentioned	SOM	CAIDA2015	Detection Rate = 97.28%	In this study, the authors used manual methods for the selection of features, and these methods can be replaced with automatic feature selection methods.

3.2 Botnet Detection and Prevention

Researchers have recently utilized machine learning techniques to develop precise and scalable frameworks for detecting and preventing botnet attacks in SDN. They have employed centralized learning with distributed detection to achieve scalability in detection. Machine-learning techniques for identifying botnets have grown significantly in the past few years. This section discusses the latest developments in machine learning for this type of attack.

A study [39] investigated different types of botnets (P2P, IRC, and HTTP botnets) in SDN controllers. The study found that Decision Tree (DT) effectively detects Peer-to-Peer botnets, while naïve Bayes and SVM detect IRC and HTTP botnets more successfully. Another study [40] used centralized network flow statistics collected by OpenFlow counters for detection, applying decision trees and C4.5 to the collected counters. The proposed method achieved an 80% detection rate for botnets, using a publicly available real-world botnet dataset for the experimental analysis. In another research paper [41], the authors analyzed potentially vulnerable hosts and malicious codes using four different classifiers: NB, DT, Bayesian Networks (BNs), and C4.5. They used historical data for prediction and deployed security rules in the SDN controller to protect potentially compromised hosts and block the entire subnet to restrict the attackers' access. Bayesian Networks achieved a higher precision rate compared to the other classifiers. Several studies have proposed different methods to detect and mitigate botnet attacks in software-defined networking (SDN) using machine learning (ML) algorithms. One way suggested by [42] is using a flow-based approach instead of packet payload inspection to detect botnets in SDN. Their system combines real-time flows with historical context to extract an enriched feature set for classification, which achieved 90% detection accuracy for unknown botnets and 97% for known botnets. Another framework introduced by [43] integrates an ML algorithm into the SDN controller to detect and categorize real-time peer-to-peer (P2P) network traffic. They accurately detected P2P network traffic using a Strom and Zeus botnet dataset for attack traffic and Skype, eMule, and uTorrent network data for normal traffic.

The study proposed by [44] suggested an ML-based framework that uses traffic flow classes to reduce detection complexity and determine high-level policies for the derived flow classes. The K-mean algorithm was used for unsupervised learning to classify NetFlow features, and the DT was used for supervised learning to classify traffic as normal or attack. In the study, [45] proposed a framework that integrates ML with SDN/NFV to detect and mitigate botnet attacks. They suggested a network function that uses network protocols to detect known attacks and collect real-time network traffic as a data set for detecting additional distributed attacks. To detect botnets, a study [46] proposed a method using Multi-Layer Perceptron (MLP) that analyzes malware traffic data collected from an existing network. This method adds a connection block to the external network and implements network

isolation to prevent internal infection. The CTU-13 and ISOT data sets were used to evaluate the method, which achieved a 99.2% accuracy rate. Another approach [47] involved building a system that uses SDN's northbound and southbound API to detect botnets. The switch sends an OpenFlow message containing statistical information to the controller at a fixed time interval within the time window. The controller then sends instructions to the deep learning classifier, which has five hidden layers based on ReLU, to block the traffic flow and isolate the infected host. This system achieved a detection accuracy of 99%. The discussed solutions for botnet attacks are summarized in Table 2.

Table 2: ML-based solutions for botnet detection in SDN

Study	Controller type	Type of botnet	Classifier type	Data set	Evaluation	Discussion
[39]	Open daylight	IRC, HTTP, P2P	Decision tree and C4.5	CTU-13, ISOT	Accuracy = 80%	They analyzed that OpenFlow counters have the potential to identify botnet behavioral patterns and are a suitable candidate for flow-based botnet detection techniques.
[40]	Not mentioned	Not mentioned	NB, DT, bayesian networks (BNs), and C4.5	Long tail	Accuracy = 91.68%	The BN achieved a high precision rate compared to the other classifiers.
[41]	Open daylight	IRC, HTTP	C4.5	ISOT and CTU-13	Accuracy for unknown botnets = 90% and Accuracy for known botnets = 97%	This method needs extensive computations.
[42]	Ryu	Not mentioned	SVM, KNN, and RF	Custom developed	Accuracy = 99.7%	SVM produces good detection results as compared to other classifiers.
[43]	Not mentioned	HTTP	K-mean and SVM	Custom developed		This method did not be implemented for the new types of network traffic.
[44]	Floodlight	IRC, HTTP, P2P	RF	CTU-13	Accuracy = 100%	This work is limited to a few protocols that need more for botnet detection in SDN.
[46]	Ryu	IRC, HTTP, P2P	Multi-layer perceptron (MLP)	CTU-13 and ISOT	Accuracy = 99.2%	The study focused on the controller and did not experiment with the terminals infected by the botnets.

3.3 Saturation Attacks Detection and Mitigation

A saturation attack is a form of adversarial attack that can impact the entire SDN network due to its prolonged duration. When the control plane of the controller is overloaded, the SDN may become unavailable. The attack involves a malicious host generating a large volume of table-miss packets, which can deplete the resources of the control plane.

In SDN-based networks, a saturation attack operates by manipulating the OpenFlow switch, which receives network packets. (i) A miss-table error occurs if a new incoming packet does not match the local flow rules. (ii) A Packet-In message containing the header of the table-miss packet will be generated in case the switch's buffer is not full. (iii) If the switch's buffer is full, the whole table-miss packet is encapsulated in the Packet-In message and sent to the controller. (iv) The controller receives the Packet-In message and processes the table-miss packet. (v) The controller also sends Packet-Mod and Packet-Out messages to install new flow rules in the switch's table. (vi) This reactive packet processing mechanism exposes the OpenFlow network to the attacker. The attacker gets an opportunity to consume the different computation resources, such as CPU, memory, etc., of the switches and controller and saturate the channels of OpenFlow connections which are responsible for delivering the forwarding messages between OpenFlow switches and controller. (vii) Once the attacker gets access to the controller, it can launch different saturation attacks such as TCP-SYN, ICMP, UDP, TCP-SARFU flooding, and IP Spoofing and their combinations (i.e., hybrid saturation attacks) at data-to-control planes. The attacker controls the many hosts (zombie machines) of the SDN network and sends forged packets, making it impossible for the controller to match the new packets with the flow rules of the switch. Thus, the controller starts to receive a large number of Packet-In messages. So, the attack from the data-to-control plane exhausts the computing resources of the controller. (viii) When a data-to-control plane flooding attack occurs, the controller sends Packet-Out and Packet-Mod messages, which cause flooding attacks from the control-to-data plane. Therefore, the flow tables of the targeted switches are filled with fake flow rules. (ix) The switch buffer is consumed and becomes unavailable for the legitimate new packets. (x) Finally, the bandwidth of the OpenFlow channel is exhausted, which disables the delivery of OpenFlow messages between the switches and the controller. The above-discussed scenario is shown in Fig. 2.

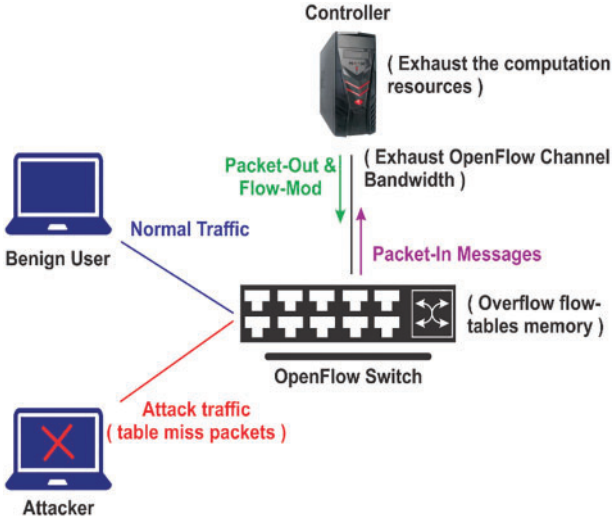

Figure 2: General procedure for a saturation attack in SDN

A study [48] examined how machine learning-based systems detect saturation attacks within an SDN environment. The study found that adversaries could bypass machine learning classifiers by creating adversarial attacks that evade detection. The authors proposed an adversarial testing tool that generated four types of saturation attacks by manipulating different traffic features to address this. They also suggested using various machine learning classifiers to improve detection, but their tests showed that the saturation attacks reduced the detection power of the system.

Another study [49] proposed a machine-learning approach to detecting saturation attacks in SDN using time windows. The authors found that if the window size were too large, the response time of the detection method would be too slow, giving the attacker time to saturate the network. Conversely, if the window size were too small, it would cause frequent false alarms and high-performance overheads for the controller. They investigated the impact of time windows on three different classifiers using OpenFlow traffic data. Reference [50] introduced a method called FlowMerge that used a Convolutional Neural Network to detect different types of saturation attacks, but the approach had some limitations. The authors generated the attack samples based on the machine learning classifier. It was unclear whether the attacks could evade other machine learning classifiers installed in the SDN controller. Reference [51] presented a Deep Neural Network-based Deep-fool algorithm to generate and detect saturation attacks in SDN. They used image inputs to perform classification. Reference [52] proposed a library named Clearhans v0.1 to create attacks in SDN and to help improve the robustness of machine learning classifiers. Finally, a framework called Fast Recovery Saturation Attack Detection and Mitigation (FSDM) was proposed [53]. The FSDM framework used Control Channel Occupation Rate (CCOR) distribution to detect the ports from where the attacker comes and used different policies to block the attack flows. The framework also included a novel function module called Force Checking, which enables the SDN controller to clean up the remaining attack flows and recover quickly.

3.4 Ransomware Detection and Prevention

Ransomware is malware that encrypts and locks a user's files and demands a ransom to release them. To spread the attack, the perpetrator seeks control of the SDN controller and employs HTTPS to deliver the malware, making extracting and identifying features through deep packet inspection difficult. However, detection and mitigation techniques are available to safeguard the SDN controller from ransomware attacks. One such technique is machine learning, and this section focuses on its development for ransomware detection and prevention in SDN.

A study [54] proposed a K-Nearest Neighbor (KNN)-based prediction system that identifies ransomware traffic packets and integrates a dynamic isolation method in SDN. The system achieves 97.7% prediction precision for ransomware. In another study [55], a two-phase stream processing and classification approach is introduced. In the stream processing phase, the system reads a flow, manages a custom flow table, and extracts flow features. In the classification phase, the Random Forest (RF) classifier trains on the extracted features to distinguish normal traffic from ransomware traffic. In yet another proposal [56], a federated learning-based anti-ransomware learning mechanism is suggested for detecting and mitigating four types of ransomware attacks: Petya, PowerGhost, BadRabbit, and WannaCry. During the defense phase, the trained federated learning classifier is installed in the SDN controller, which detects ransomware attacks and blocks traffic from the victim device.

4 Vulnerability Analysis of Solution Space from the Literature

The following section discusses the limitations of the suggested solutions and sheds light on the current research challenges in the field. Existing literature explores advanced machine learning

techniques to identify and address these attacks in an SDN controller, as presented in the solution space. To evaluate the effectiveness of machine learning algorithms, the Confusion Matrix is utilized, as depicted in Fig. 3a. Thus, the research gaps in the literature are outlined below.

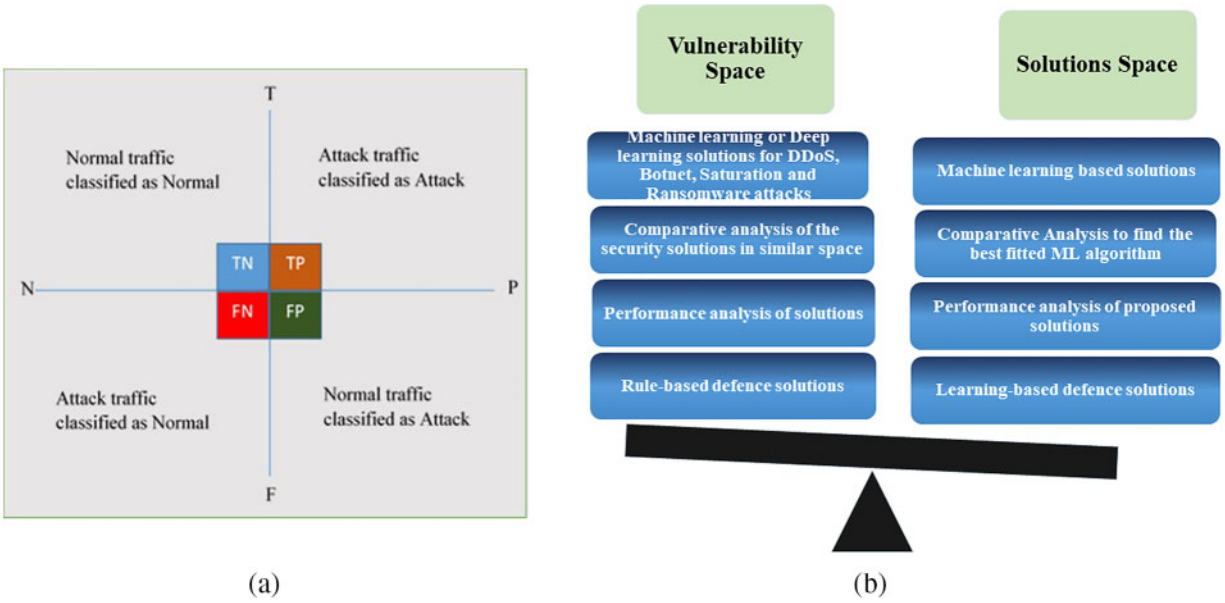

Figure 3: (a) Confusion matrix for performance evaluation of machine learning algorithms. (b) Classification of vulnerability analysis

4.1 Lack of Discussion on Every Single Performance Evaluation Measure

Although numerous machine learning-based solutions are available for detecting and mitigating different types of cyberattacks in the SDN realm, such as DDoS, botnet, saturation, and ransomware, a significant issue with these solutions is that the performance metric used for evaluating them may need to be revised. Accuracy, comprehensiveness, and performance are essential for assessing machine learning-based security solutions. Proving the exactitude of these solutions can be both costly and challenging. Many research papers need to validate all the critical parameters in the performance metrics. For instance, in some studies, the solution's performance is based on the correct and incorrect classification of attack traffic, as shown in Fig. 3a. Therefore, the performance metric for these attacks should include essential parameters such as True Negative (TN), True Positive (TP), False Negative (FN), False Positive (FP), Accuracy, Reliability, Specificity, False Positive Rate (FPR), False Negative Rate (FNR), F1-Score, Recall, and Precision.

4.2 Vulnerability Analysis

Fig. 3b presents a classification of the vulnerability analysis obtained from the literature survey. The figure's vulnerability space highlights the research gaps identified in the literature, while the solution space indicates the technological approaches required to address those gaps. In other words, the figure illustrates the areas where existing solutions are lacking and suggests potential avenues for future research.

4.3 Research Challenges and Future Directions

Although SDN has changed the way for future networks, it has brought different security challenges that need the immediate attention of the research community. These challenges hinder the development of more efficient and flexible Network Intrusion Detection Systems (NIDS's) using ML and DL for SDN-based networks [5]. The open research challenges are listed below:

- **Availability of benchmark dataset:** The existing solutions rely on old datasets or simulated traffic streams [30–32]. They are inaccurate for further academic research because they need proper data classification. Most network researchers are using synthetic datasets to detect different attacks in SDN due to a lack of realistic and better datasets. Several datasets are available for testing and evaluating machine learning-based detection systems. The most widely used datasets are NSL-KDD, CSE-CIC-IDS2018, ISCX, etc. However, all these datasets have been generated and collected over a conventional network. These are not suitable for SDN-based networks because the SDN networks work on flow-based mechanisms. So, creating standard datasets (labeled or unlabeled) that are consistent, accurate, and flow-based for detecting attacks in an SDN controller.
- **Selection of appropriate features:** Another predominant challenge is to select and use the most suitable feature selection methods that can precisely explore the relevant features for detecting these attacks and also need to determine the redundancy between these features. Therefore, improving computational realism and determining an optimal number of model parameters is challenging in ML/DL.
- **Machine learning-based mitigation techniques:** The issue of generating different attacks such as DDoS, botnet, saturation, and ransomware against the SDN controller is entirely new, and only some initial work exists to mitigate these attacks.
- **Need to handle the colossal network traffic in SDN:** Due to the centralized nature of SDN, the conventional detection and mitigation techniques cannot be used because these are insufficient to provide an efficient security solution to the SDN controller. The SDN aims to handle heterogeneous networks, where many network devices, on a large scale, are geographically distributed. Moreover, the machine learning-based solutions do not consider the large volume of real-time traffic data. So, this issue also needs the care of researchers in the future.
- **Need to evaluate the solutions on simulated experimental testbeds:** Although the machine learning research community has proposed many solutions, the exponential rise in frequency and attack size shows the ineffectualness of the proposed solutions [30]. Researchers try to detect and mitigate these attacks in different studies in simulated environments. Hence, there is a keen need to devote considerable effort to developing real-time experimental testbeds.
- **Collectively detect and mitigate the attacks against the SDN controller:** The SDN controller consists of three layers: application, control, and data. Most researchers focus on providing machine learning solutions for individually detecting and mitigating attacks at three layers. But, there is a need to collectively detect and mitigate the attacks at all layers to save the whole SDN-based network architecture from these attacks.
- **Need to focus on distributed multi-controller architectures:** Network Scalability is another critical issue in SDN-based networks. In most research studies [16,22], the researcher deployed a single controller in the control plane, used machine learning techniques to detect and mitigate attacks, and achieved good results. But, with the increase in the number of flows and network size in a single controller architecture, the controller starts to face scalability issues, which limits its computational power. In some studies, the researcher used distributed multi-controller platforms to solve the scalability and single-point failure issues. Generally, a distributed multi-controller

platform consists of a central root controller and several local controllers. The root controller serves as a logical center and has complete access to all the other controllers and switches in the network. It can detect and mitigate attacks by regularly deploying trained machine learning (ML) models on the local controllers. These models enable the local controllers to detect attacks directly within the traffic flows. Furthermore, the trained models reduce the workload of the single controller and improve the scalability issues in the SDN-based networks. So, it is recommended that the researchers consider a distributed multi-controller platform instead of a single controller for future research. They must try to deploy machine learning models in a distributed multi-controller platform to detect better and mitigate different attacks.

- **Need to handle large traffic flows:** Another big challenge is handling packet processing flows effectively while implementing the ML/DL approach with a large volume of data. There are difficulties in finding forged traffic in normal flows. Also, the controllers can face a performance bottleneck due to a large amount of forwarding and incoming data. So, we must introduce the ML/DL techniques that immediately and intelligently detect the forged traffic and help reduce the controller bottleneck.

5 Conclusion

In this paper, we have studied and examined the current and state-of-the-art ML techniques for securing SDN. We began our discussion by briefly introducing SDN and machine learning. Then, we analyzed and highlighted the security issues and the most devastating four attacks on the SDN controller. After that, we discussed the recent machine-learning developments for detecting and mitigating these attacks in the SDN controllers. Finally, we conducted a vulnerability and gap analysis of the existing solutions. We concluded with general vulnerabilities or gaps in the ML-based solutions to secure the SDN for future research in this field.

In summary, applying machine learning algorithms to secure software-defined network (SDN)-based networks is an expansive area of research, with numerous challenges yet to be addressed. This review aims to provide an overview of vulnerability analysis in order to aid in developing SDN technologies and creating solutions for detecting and mitigating attacks. Given the significance of these challenges, both the network and machine learning communities must address them and advance in this field. Additionally, this article briefly examines the appropriate use of machine learning algorithms in resolving security issues.

Funding Statement: The authors received no funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] W. I. Khedr, A. E. Gouda and E. R. Mohamed, "FMDADM: A multi-layer DDoS attack detection and mitigation framework using machine learning for stateful SDN-based IoT networks," *IEEE Access*, vol. 11, pp. 28934–28954, 2023.

[2] A. Iranmanesh and H. R. Naji, "A protocol for cluster confirmations of SDN controllers against DDoS attacks," *Computers & Electrical Engineering*, vol. 93, pp. 107265, 2021.

[3] T. E. Ali, Y. W. Chong and S. Manickam, "Machine learning techniques to detect a DDoS attack in SDN: A systematic review," *Applied Sciences*, vol. 13, no. 5, pp. 3183–3209, 2023.

[4] M. Revathi, V. V. Ramalingam and B. Amutha, "Rmcartam for DDoS attack mitigation in SDN using machine learning," *Computer Systems Science and Engineering*, vol. 45, no. 3, pp. 3023–3036, 2023.

[5] M. W. Nadeem, H. G. Goh, V. Ponnusamy and Y. Aun, "DDoS detection in SDN using machine learning techniques," *Computer Material Continua*, vol. 71, no. 1, 2022. https://doi.org/10.32604/cmc.2022.021669

[6] Y. Maleh, Y. Qasmaoui, K. El Gholami, Y. Sadqi and S. Mounir, "A comprehensive survey on SDN security: Threats, mitigations, and future directions," *Journal of Reliable Intelligent Environments*, vol. 9, pp. 201–239, 2022.

[7] M. A. Bouke, A. Abdullah, S. H. ALshatebi, M. T. Abdullah and H. El Atigh, "An intelligent DDoS attack detection tree-based model using gini index feature selection method," *Microprocessors and Microsystems*, vol. 98, pp. 104823, 2023.

[8] G. Somani, M. S. Gaur, D. Sanghi, M. Conti and R. Buyya, "DDoS attacks in cloud computing: Issues, taxonomy, and future directions," *Computer Communications*, vol. 107, pp. 30–48, 2017.

[9] K. S. Sahoo, S. K. Panda, S. Sahoo, B. Sahoo and R. Dash, "Toward secure software-defined networks against distributed denial of service attack," *Journal of Supercomputers*, vol. 75, no. 8, pp. 4829–4874, 2019.

[10] N. Sultana, N. Chilamkurti, W. Peng and R. Alhadad, "Survey on SDN based network intrusion detection system using machine learning approaches," *Peer-to-Peer Network Applications*, vol. 12, no. 2, pp. 493–501, 2019.

[11] K. Kalkan, L. Altay, G. Gür and F. Alagöz, "JESS: Joint entropy-based DDoS defense scheme in SDN," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 10, pp. 2358–2372, 2018.

[12] N. A. S. Lima and M. P. Fernandez, "Towards an efficient DDoS detection scheme for software-defined networks," *IEEE Latin America Transactions*, vol. 16, no. 8, pp. 2296–2301, 2018.

[13] P. Kumar, M. Tripathi, A. Nehra, M. Conti and C. Lal, "SAFETY: Early detection and mitigation of TCP SYN flood utilizing entropy in SDN," *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1545–1559, 2018.

[14] R. Braga, E. Mota and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in *IEEE Local Computer Network Conf.*, Denver, CO, USA, pp. 408–415, 2010.

[15] X. D. Zang, J. Gong and X. Y. Hu, "An adaptive profile-based approach for detecting anomalous traffic in backbone," *IEEE Access*, vol. 7, pp. 56920–56934, 2019.

[16] Y. Xu, H. Sun, F. Xiang and Z. Sun, "Efficient DDoS detection based on K-FKNN in software defined networks," *IEEE Access*, vol. 7, pp. 160536–160545, 2019.

[17] L. Tan, Y. Pan, J. Wu, J. Zhou, H. Jiang *et al.,* "A new framework for DDoS attack detection and defense in SDN environment," *IEEE Access*, vol. 8, pp. 161908–161919, 2020.

[18] T. Hurley, J. E. Perdomo and A. Perez-Pons, "HMM-based intrusion detection system for software defined networking," in *15th IEEE Int. Conf. on Machine Learning and Applications (ICMLA)*, Anaheim, CA, USA, pp. 617–621, 2016.

[19] A. Alshamrani, A. Chowdhary, S. Pisharody, D. Lu and D. Huang, "A defense system for defeating DDoS attacks in SDN based networks," in *Proc. of the 15th ACM Int. Symp. on Mobility Management and Wireless Access*, Miami, Florida, USA, pp. 83–92, 2017.

[20] D. Hu, P. Hong and Y. Chen, "FADM: DDoS flooding attack detection and mitigation system in software-defined networking," in *GLOBECOM 2017–2017 IEEE Global Communications Conf.*, Singapore, pp. 1–7, 2017.

[21] S. U. N. Guozi, W. Jiang, G. U. Yu, R. E. N. Danni and L. I. Huakang, "DDoS attacks and flash event detection based on flow characteristics in SDN," in *15th IEEE Int. Conf. on Advanced Video and Signal Based Surveillance (AVSS)*, Auckland, New Zealand, pp. 1–6, 2018.

[22] V. Deepa, K. M. Sudar and P. Deepalakshmi, "Design of ensemble learning methods for DDoS detection in SDN environment," in *Int. Conf. on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*, Vellore, India, pp. 1–6, 2019.

[23] J. Cui, J. He, Y. Xu and H. Zhong, "TDDAD: Time-based detection and defense scheme against DDoS attack on SDN controller," in *Australasian Conf. on Information Security and Privacy*, Wollongong, NSW, Australia, pp. 649–665, 2018.

[24] C. Li, Y. Wu, X. Yuan, Z. Sun, W. Wang *et al.,* "Detection and defense of DDoS attack–based on deep learning in OpenFlow-based SDN," *International Journal of Communication Systems*, vol. 31, no. 5, pp. e3497, 2018.

[25] T. M. Nam, P. H. Phong, T. D. Khoa, T. T. Huong, P. N. Nam *et al.,* "Self-organizing map-based approaches in DDoS flooding detection using SDN," in *2018 Int. Conf. on Information Networking (ICOIN)*, Chiang Mai, Thailand, pp. 249–254, 2018.

[26] M. P. Novaes, L. F. Carvalho, J. Lloret and M. L. Proença, "Long short-term memory and fuzzy logic for anomaly detection and mitigation in software-defined network environment," *IEEE Access*, vol. 8, pp. 83765–83781, 2020.

[27] K. S. Sahoo, B. K. Tripathy, K. Naik, S. Ramasubbareddy, B. Balusamy *et al.,* "An evolutionary SVM model for DDOS attack detection in software defined networks," *IEEE Access*, vol. 8, pp. 132502–132513, 2020.

[28] M. Revathi, V. V. Ramalingam and B. Amutha, "A machine learning based detection and mitigation of the DDOS attack by using SDN controller framework," *Wireless Personal Communications*, vol. 127, pp. 2417–2441, 2021.

[29] N. M. Yungaicela-Naula, C. Vargas-Rosales and J. A. Perez-Diaz, "SDN-based architecture for transport and application layer DDoS attack detection by using machine and deep learning," *IEEE Access*, vol. 9, pp. 108495–108512, 2021.

[30] R. T. Kokila, S. T. Selvi and K. Govindarajan, "DDoS detection and analysis in SDN-based environment using support vector machine classifier," in *Sixth Int. Conf. on Advanced Computing (ICoAC)*, Chennai, India, pp. 205–210, 2014.

[31] R. M. A. Ujjan, Z. Pervez, K. Dahal, A. K. Bashir, R. Mumtaz *et al.,* "Towards sFlow and adaptive polling sampling for deep learning based DDoS detection in SDN," *Future Generation Computer Systems*, vol. 111, pp. 763–779, 2020.

[32] M. P. Novaes, L. F. Carvalho, J. Lloret and M. L. Proença Jr, "Adversarial deep learning approach detection and defense against DDoS attacks in SDN environments," *Future Generation Computer Systems*, vol. 125, pp. 156–167, 2021.

[33] Z. Liu, Y. He, W. Wang and B. Zhang, "DDoS attack detection scheme based on entropy and PSO-BP neural network in SDN," *China Communications*, vol. 16, no. 7, pp. 144–155, 2019.

[34] L. Wang and Y. Liu, "A DDoS attack detection method based on information entropy and deep learning in SDN," in *IEEE 4th Information Technology, Networking, Electronic and Automation Control Conf. (ITNEC)*, Chongqing, China, vol. 1, pp. 1084–1088, 2020.

[35] Y. Xu, Y. Yu, H. Hong and Z. Sun, "DDoS detection using a cloud-edge collaboration method based on entropy-measuring SOM and KD-tree in SDN," *Secure Communication Networks*, vol. 2021, pp. 1–16, 2021.

[36] Z. Long and W. Jinsong, "A hybrid method of entropy and SSAE-SVM based DDoS detection and mitigation mechanism in SDN," *Computer Security*, vol. 115, pp. 102604, 2022.

[37] Y. Cao, H. Jiang, Y. Deng, J. Wu, P. Zhou *et al.,* "Detecting and mitigating DDoS attacks in SDN using spatial-temporal graph convolutional network," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 6, pp. 3855–3872, 2021.

[38] A. Maheshwari, B. Mehraj, M. S. Khan and M. S. Idrisi, "An optimized weighted voting based ensemble model for DDoS attack detection and mitigation in SDN environment," *Microprocess Microsystems*, vol. 89, pp. 104412, 2022.

[39] U. Wijesinghe, U. Tupakula and V. Varadharajan, "Botnet detection using software defined networking," in *22nd Int. Conf. on Telecommunications (ICT)*, Sydney, NSW, Australia, pp. 219–224, 2015.

[40] F. Tariq and S. Baig, "Botnet classification using centralized collection of network flow counters in software defined networks," *International Journal of Computer Science and Information Security*, vol. 14, no. 8, pp. 1075, 2016.

[41] S. Nanda, F. Zafari, C. DeCusatis, E. Wedaa and B. Yang, "Predicting network attack patterns in SDN using machine learning approach," in *IEEE Conf. on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Palo Alto, CA, USA, pp. 167–172, 2016.

[42] F. Tariq and S. Baig, "Machine learning based botnet detection in software defined networks," *International Journal of Secure Applications*, vol. 11, no. 11, pp. 1–12, 2017.

[43] S. C. Su, Y. R. Chen, S. C. Tsai and Y. B. Lin, "Detecting p2p botnet in software defined networks," *Secure Communication Networks*, vol. 2018, pp. 1–13, 2018.

[44] D. Comaneci and C. Dobre, "Securing networks using SDN and machine learning," in *IEEE Int. Conf. on Computational Science and Engineering (CSE)*, Bucharest, Romania, pp. 194–200, 2018.

[45] Y. Park, N. V. Kengalahalli and S. -Y. Chang, "Distributed security network functions against botnet attacks in software-defined networks," in *IEEE Conf. on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Verona, Italy, pp. 1–7, 2018.

[46] S. Maeda, A. Kanai, S. Tanimoto, T. Hatashima and K. Ohkubo, "A botnet detection method on SDN using deep learning," in *IEEE Int. Conf. on Consumer Electronics (ICCE)*, Las Vegas, NV, USA, pp. 1–6, 2019.

[47] I. Letteri, M. Del Rosso, P. Caianiello and D. Cassioli, "Performance of botnet detection by neural networks in software-defined networks," in *The Second Italian Conf. on Cyber Security (ITASEC18)*, Milan, Italy, pp. 1–10, 2018.

[48] S. Y. Khamaiseh, I. Alsmadi and A. Al-Alaj, "Deceiving machine learning-based saturation attack detection systems in SDN," in *IEEE Conf. on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Leganes, Spain, pp. 44–50, 2020.

[49] S. Khamaiseh, E. Serra, Z. Li and D. Xu, "Detecting saturation attacks in SDN via machine learning," in *4th Int. Conf. on Computing, Communications and Security (ICCCS)*, Rome, Italy, pp. 1–8, 2019.

[50] A. Abusnaina, A. Khormali, D. Nyang, M. Yuksel and A. Mohaisen, "Examining the robustness of learning-based DDoS detection in software defined networks," in *IEEE Conf. on Dependable and Secure Computing (DSC)*, Hangzhou, China, pp. 1–8, 2019.

[51] S. -M. Moosavi-Dezfooli, A. Fawzi and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 2574–2582, 2016.

[52] N. Papernot, F. Faghri, N. Carlini, I. Goodfellow, R. Feinman *et al.,* "Technical report on the cleverhans v2. 1.0 adversarial examples library," arXiv Prepr. arXiv1610.00768, 2016.

[53] X. Huang, K. Xue, Y. Xing, D. Hu, R. Li *et al.,* "FSDM: Fast recovery saturation attack detection and mitigation framework in SDN," in *IEEE 17th Int. Conf. on Mobile Ad Hoc and Sensor Systems (MASS)*, Delhi, India, pp. 329–337, 2020.

[54] H. Y. Chang, T. L. Lin, T. F. Hsu, Y. S. Shen and G. R. Li, "Implementation of ransomware prediction system based on weighted-KNN and real-time isolation architecture on SDN networks," in *IEEE Int. Conf. on Consumer Electronics-Taiwan (ICCE-TW)*, Yilan, Taiwan, pp. 1–2, 2019.

[55] G. Cusack, O. Michel and E. Keller, "Machine learning-based detection of ransomware using SDN," in *Proc. of the ACM Int. Workshop on Security in Software Defined Networks & Network Function Virtualization*, New York, US, pp. 1–6, 2018.

[56] C. Thapa, K. K. Karmakar, A. H. Celdran, S. Camtepe, V. Varadharajan *et al.,* "FedDICE: A ransomware spread detection in a distributed integrated clinical environment using federated learning and SDN based mitigation," in *Int. Conf. on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, Melbourne, Australia, pp. 3–24, 2021.