Cluster Computing (2023) 26:3089–3112
https://doi.org/10.1007/s10586-023-04069-9

Deep learning for the security of software-defined networks: a review

Roya Taheri[1] · Habib Ahmed[1] · Engin Arslan[1]

Received: 19 January 2023 / Revised: 24 May 2023 / Accepted: 29 May 2023 / Published online: 15 July 2023

Abstract
As the scale and complexity of networks grow rapidly, management, maintenance, and optimization of them are becoming increasingly challenging tasks for network administrators. Software-Defined Networking (SDN) was introduced to facilitate these tasks as it offers logically centralized control, a global view of the network, and software-based traffic analysis, thus, it is widely adopted of SDN to manage large-scale networks. On the other hand, SDN is not immune to cyber attacks. In fact, its centralized architecture makes it more vulnerable to certain types of attacks, such as denial of service. Various attack mitigation strategies are proposed to strengthen the security of SDNs including statistical, threshold-based, and Machine Learning (ML) methods. Among them, Deep Learning (DL)-based models attained the best results as they were able to extract the complex relationship between input parameters and output that could not be achieved with other solutions. Hence, this paper presents a comprehensive survey of the literature on the utilization of different DL algorithms for the security of SDN. We first explain the types of attacks that SDNs are exposed to, then present papers that applied DL to detect and/or mitigate these attacks. We further discuss the public datasets used to train DL models and evaluate their advantages and disadvantages. Finally, we share insights into future research directions to improve the efficiency of DL methods for SDN security.

Keywords Software-defined networks · Network security · SDN security · Deep learning

1 Introduction

Software Defined Networking (SDN) has been introduced to allow forwarding decisions to be made by a central controller, which can calculate optimal routes based on global network view and unique application demands [1]. Unlike traditional routing solutions that are rigid and mostly application oblivious, SDN offers an agile network management approach, which led to its wide adoption. For example, Google uses SDN to manage its wide-area network traffic [2]. On the other hand, SDN introduced a plethora of architectural vulnerabilities that pose significant risks to the safety of networks [3-10].

Researchers proposed many heuristics and statistical methods to overcome the SDN's security issues, but Machine Learning (ML) models are found to be much more effective as they can extract the complex relationship between input and output that cannot be easily realized with other approaches [1, 11-15]. In particular, a subfield of ML, Deep Learning, offers a significant advantage over statistical models and traditional ML methods when dealing with large-scale datasets. Thus, DL has been applied to many fields such as image processing [16, 17], biomedical imaging [18-21], automated inspection of civil infrastructure [22-26], and robotics [27-31] to solve complex problems. Researchers, therefore, proposed DL-based methods to detect and/or mitigate security issues in SDN-based networks.

Although there exist several surveys papers on the application of ML-based techniques for SDN security [32-41], they are either outdated (published several years ago thus they do not reflect the current status) or limited in scope (i.e., focus on one particular component of SDN or one type of attack). As a result, there is a need to provide an up-to-date, systematic, and comprehensive review of research on the application of DL for SDN security.

Figure 1 highlights the main areas along with some of the sub-topics that are discussed in this study. We first

✉ Engin Arslan
earslan@unr.edu

[1] Department of Computer Science and Engineering, University of Nevada, Reno, USA

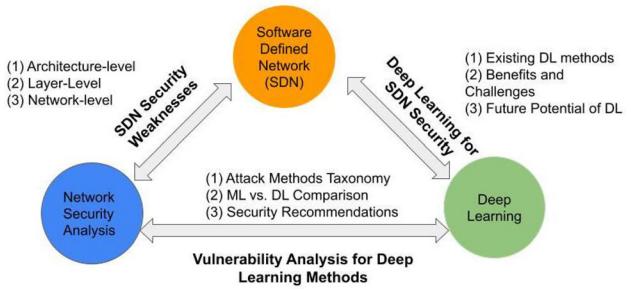

Fig. 1 Visual outline of the different topics that will be covered in this review in relation to the three main research areas, namely SDNs, network security, and Deep Learning

provide a brief overview of SDN and DL. Next, we highlight the security weaknesses of the different parts of the SDN architecture and review DL-based methods proposed to address these weaknesses. Then, we analyze performance metrics used to evaluate the effectiveness of DL methods and we provide a comprehensive overview of different publicly-available datasets used by researchers to train DL models and discuss their benefits and limitations. Finally, we present future opportunities and research directions that can be to facilitate the use of DL for SDN security issues.

The remainder of this survey paper is organized as follows: Section II reviews the main focus and limitations of some existing surveys. Section III presents the background for SDN and DL. Section IV discusses security vulnerabilities for different SDN components and existing challenges and associated opportunities in the use of DL-based methods for SDN security. Section IV highlights three major aspects of DL-based methods and SDN, namely public datasets for enhancing SDN security, performance evaluation metrics used in studies, and classification of different security applications for SDN as discussed in the literature. Section VI discusses the remaining challenges that the SDNs face despite and. Section VII summarizes the paper and highlights some future directions for researchers.

Most of previous survey studies on SDN focused on routing solutions [33, 42]), architectural deployment scenarios (e.g., distributed SDN [43] and Hybrid SDN [41, 44, 45]), control plane implementations [40], and application areas (e.g., SDN-VANETs [46], SDN for IoT [47, 48], SDN for data centers [49], edge-based SDN [50])). Although there are many survey papers published in the area of SDN security, many of them (e.g. [51-58])) were published more than 5 years ago, thus they do not necessarily reflect the current state of the research anymore. Papers that are published within the last 5 years are listed in Table 1 with their publication date, focus area, main topics, as well as their limitation compared to this work. Specifically, they either focus on one aspect of SDN security or do not cover ML/DL solutions with sufficient details. For example, [39, 59] focuses on DDoS attacks and relevant mitigation strategies, [38] concentrates on information security, and [60, 61] discuss Intrusion Detection Systems (IDS) and [62] discusses Attack Detection systems (ADS). In contrast, this survey provides an extensive evaluation of a broad range of security issues in all layers of SDN architecture. We also cover a wide range of security applications. On the other hand, although [34-37] cover SDN security in a broad scope, they do not sufficiently highlight ML/DL-based solutions. For example, [34] covers attacks in different SDN layers but mentions only five ML-based solutions. Similarly, [35] details security frameworks implemented for different layers of SDN architecture but presents only four ML/DL-based solutions out of many available, which indicates that ML/DL is not the primary focus. [37] focuses on security vulnerabilities in SDN architecture and covers only five ML/DL approaches. Although [32, 63] cover ML/DL methods SDN, security is not the only focus of these work (i.e., they cover a wide range of topics including traffic classification, routing optimization, quality of service prediction, and resource management), thus they fall short to provide a sufficient examination of previous work in SDN security. *Consequently, this is the first study that provides a comprehensive overview of ML/DL solutions in the focus area of SDN security*. The abbreviations used in this paper are listed in Table 2.

2 Background

This section provides a brief background on Deep Learning (DL) and Software Defined Networking (SDN) to explain key concepts that are referred to in the following sections.

2.1 Deep learning (DL)

A Neural Network (NN) comprises several connected processing units or nodes that operate parallel to update link weights using nonlinear computations to minimize error [65]. These nodes use activation functions to perform nonlinear analyses. The most basic NN (aka shallow NN) consists of three layers the input layer, hidden layer, and output layer. Deep neural networks (aka Deep Learning) is a term used for NNs with more than three layers, with several hidden layers and neurons needed to process high-dimensional data and learn increasingly complex models. In Deep Learning (DL), neurons train a feature representation based on the previous layer's output. Thus, they perform better than traditional ML methods when handling large-scale high-dimensional datasets [65, 66]. However, DL methods require greater computational power like

Table 1 Different review-based studies and their contribution to the research areas intersecting between SDNs, SDN Security and ML/DL for SDN Security

Study	Year	Main focus	Major topics covered	Limitation
[41]	2018	Hybrid SDN Networks (HSDN)	(1) Overview of hybrid SDN frameworks and controllers (2) HSDN testing, verification and traffic management (3) HSDN security and future directions	(1) SDN Scope narrow to HSDNs (2) Minor emphasison Security
[64]	2019	Security threats and mitigation for SDN Controllers	(1) SDN architecture overview (2) Security attacks and protection for SDN controllers (3) Open research issues, challenges and direction	(1) SDN scope limited to Controllers (2) ML/DL discussion limited
[32]	2019	ML for SDNs	(1) Background on SDN and ML (2) ML for SDN, challenges and future direction	(1) Focus on all ML techniques (2) Minor emphasis on security
[63]	2019	ML for SDNs	(1) ML for SDNs (2) SDN applications with ML and future directions	(1) Focus on all ML techniques (2) Minor emphasis on security
[40]	2019	Route optimization using ML	(1) SDN overview (2) ML for route optimization and future directions	(1) Security emphasis missing (2) ML Scope too narrow
[60]	2019	SDN intrusion Detection systems (IDS)	(1) SDN overview (2) SDN-based IDS (3) Research challenges	(1) Limited Scope (IDS only) (2) Limited Discussion of ML techniques
[61]	2021	SDN intrusion Detection systems (IDS)	(1) SDN overview (2) SDN-based IDS (3) ML	(1) Limited Scope (IDS only) DL techniques for IDS
[42]	2019	SDN/NFV	(1) Potential of SDN/NFV (2) Algorithmic challenges of SDN/NFV (3) ML for improved optimization of SDN/NFV	(1) ML discussion limited to Network Optimization (2) SDN Security missing
[44]	2019	Optimization of Hybrid SDNs (HSDNs)	(1) Rationale for HSDNs (2) Control and data plane deployment solutions (3) HSDN deployment, optimization and use cases	(1) Scope limited to HSDNs (2) Security discussion missing
[59]	2019	DDoS attacks in Cloud-based SDN (CSDN)	(1) Background on SDN, cloud computing and DDoS (2) DDoS attacks in CSDN (3) Experimental setup for CSDN and open problems	(1) Security Scope limited to DDoS attacks (2) Scope limited to CSDNs
[47]	2020	SDN/NFV for IoT security	(1) NFV overview (2) SDN and NFV for IoT and future direction	(1) SDN Scope narrow to IoT
[62]	2021	Anomaly detection Systems (ADS)	(1) SDN Overview and security challenges (2) ADS Taxonomy, challenges and future directions	(1) Security Scope limited to ADS
[39]	2020	Flow-based DDoS Attacks in SDN	(1) DDoS attack classification (2) DDoS attack detection & mitigation (3) Challenges and future direction	(1) Security Scope narrow to DDoS attacks
[45]	2021	Hybrid SDN (HSDN)	(1) HSDN security and privacy (2) HSDN network management & deployment tools (3) HSDN open research challenges & areas	(1) Scope limited to HSDN (2) Security discussion limited
[33]	2021	SDN control plane (CP)	(1) Background on SDN CPs (2) Centralized and decentralized CPs (3) Controller, CP challenges and future directions	(1) Scope limited to SDN CP and SDN Controllers
[35]	2021	SDN security Issues and Solutions	(1) SDN security background (2) Security in SDN architecture (3) Discussion, open challenges and future research	(1) ML/DL discussion limited

multiple GPUs, to train DL models in a reasonable time. The advancements of DL model development are the increased availability of GPUs, which allows for significantly faster computation and layers that can be trained

Table 2 The list of abbreviations used in this paper

Abbreviation	Definition
SDN	Software-defined network
NFV	Network function virtualization
DDoS	Distributed denial of service
IDS	Intrusion detection system
ADS	Attack detection system
ML	Machine learning
DL	Deep learning
NN	Neural network
DNN	Deep neural network
CNN	Convolutional neural network
RNN	Recurrent neural network
MLP	Multi-layer perceptron
GRU	Gated recurrent unit
GAN	Generative adversarial network
AE	Auto-encoder
SAE	Stacked auto-encoder
DAE	Deep auto-encoder
CAE	Contractive auto-encoder
VAE	Variational auto-encoder
SOM	Self-organizing map
RBM	Restricted Boltzmann machine
DBN	Deep belief network
DTL	Deep transfer learning
DRL	Deep reinforcement learning
KNN	K-nearest neighbor
SVM	Support vector machine
NB	Naive bayes

independently. Thus, a large model with million parameters can be optimized in small, manageable chunks, requiring significantly fewer resources [67]. DL is shown to perform significantly well in a wide range of security applications, especially when dealing with complex problems in high-dimensional data [67, 68]. In the broader perspective, DL-based methods can be classified into three as supervised, unsupervised, and hybrid methods as shown in Fig. 2.

2.1.1 Supervised (discriminative) deep learning

This category of DL models is trained using labeled datasets. Supervised architectures mainly include Multi-Layer Perceptron (MLP), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN) along with their variants [67].

- *Multi-layer perceptron (MLP)* MLP is a feed-forward artificial neural network model consisting of a fully connected input layer that receives the input data to be processed, an output layer that performs prediction and classification about the input, and one or more hidden layers in between these two layers for doing computational functions. The neurons in the MLP are trained with the backpropagation learning algorithm. Several activation functions are used to determine the output of an MLP network. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron and offers to solve problems that are not linearly separable [67].
- *Convolutional neural network (CNN)* CNN consists of two parts feature extraction and classification. Feature extraction consists of convolution and pooling layers. The convolution layer takes an input and performs a series of convolution operations on that whereas the pooling layer reduces the number of parameters by downsampling along the spatial dimensionality of the input [69]. It also deals with the problem of over-fitting, which may occur in typical neural networks [67]. The classification layer is a fully connected layer that uses the features that were taken from a convolutional layer in the previous steps to calculate the score for each class. The architecture of CNN is depicted in Fig. 3.
- *Recurrent Neural Network (RNN)* RNN is a dynamic feed-forward neural network distinguished by its ability to learn sequential data over timesteps as shown in Fig. 4. In conventional feed-forward neural networks, the output of each unit depends on the current input with no direct dependency between the current input and the previous output of the same unit. However, some applications rely on sequential data, such as speech recognition and time-series data in which consecutive samples are correlated [67]. Long Short-Term Memory (LSTM) and A Gated Recurrent Unit (GRU) are popular variants of RNN. LSTM contains a memory cell to remember previous data and three gates (namely input, output, and forget) to manage the flow of information in and out of the cell. This variant solves the issue of the vanishing and exploding gradients that could happen due to multiplexing many tiny or large derivatives [70, 71]. GRU is another variant of RNN. For avoiding vanishing or exploding gradients, LSTM has a higher memory requirement given multiple memory cells in their architecture. Similar to LSTM, GRU uses gating methods to control and manage information flow between cells in the neural network, without having separate memory cells [72]. GRU structure is similar to LSTM without an output gate and because of this structure, it can capture dependencies from extensive data sequences adaptively without losing information from earlier portions of the sequence [67].

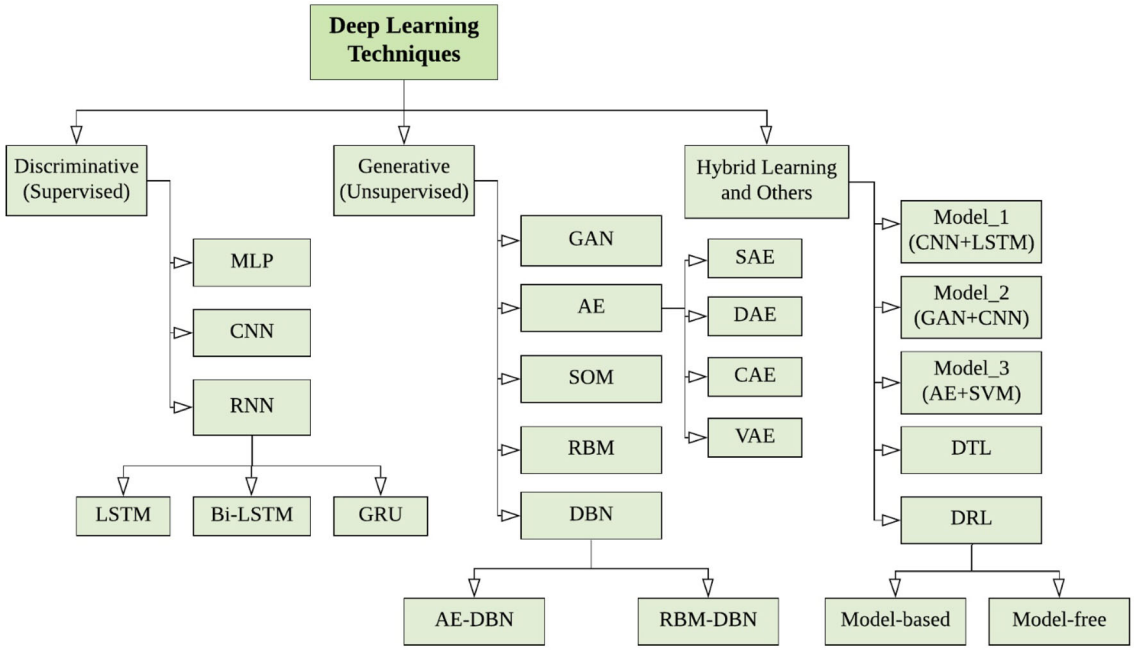

Fig. 2 Basic taxonomy of Deep Learning (DL) methods provided in [67], which classify DL methods into Supervised, Unsupervised and Hybrid methods and their sub-categories

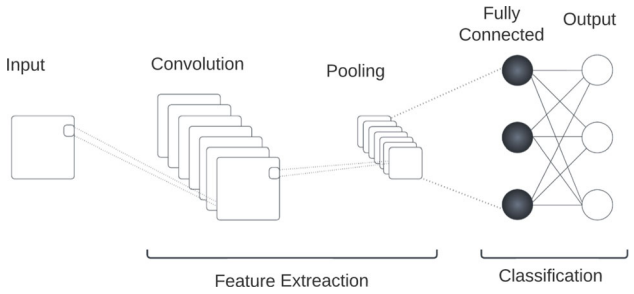

Fig. 3 Architecture of convolutional neural network

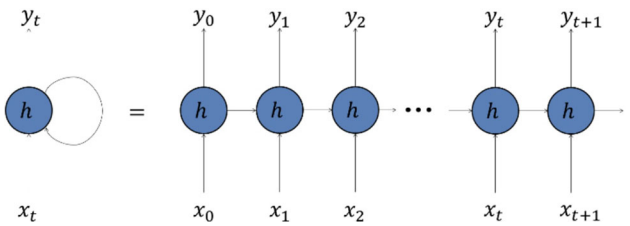

Fig. 4 Representation of recurrent neural networks

2.1.2 Unsupervised (generative) deep learning

This category of DL discovers hidden patterns or data groupings in unlabeled datasets without the need for manual data labeling [67]. Commonly used DNN techniques for unsupervised learning are as follows:

- *Generative adversarial network (GAN)* It learns regularities or patterns in input data so that the model can generate samples similar to the ones in the original dataset. Although GAN is mainly designed for unsupervised learning problems, it is shown to perform well for some semi-supervised and reinforcement learning tasks. Figure 5 illustrates a typical GAN architecture.
- *Auto-Encoder (AE)* Auto-Encoder creates noise-free data representation by reducing the dimensionality of input data as shown in Fig. 6. The encoder reduces the dimension of the data and the decoder increases it back to its original size. There are several hidden layers between the encoder and decoder to learn the relationship between input and output using a backpropagation algorithm.
- *Self-organizing map (som)* SOM is another unsupervised learning technique with a feedforward structure that is used to make a low-dimensional representation of a dataset while preserving the topological structure

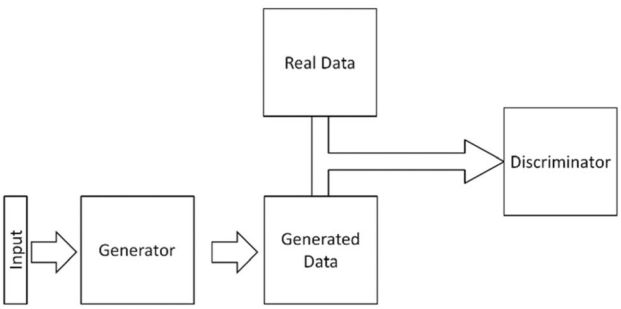

Fig. 5 Architecture of Generative Adversarial Networks (GAN)

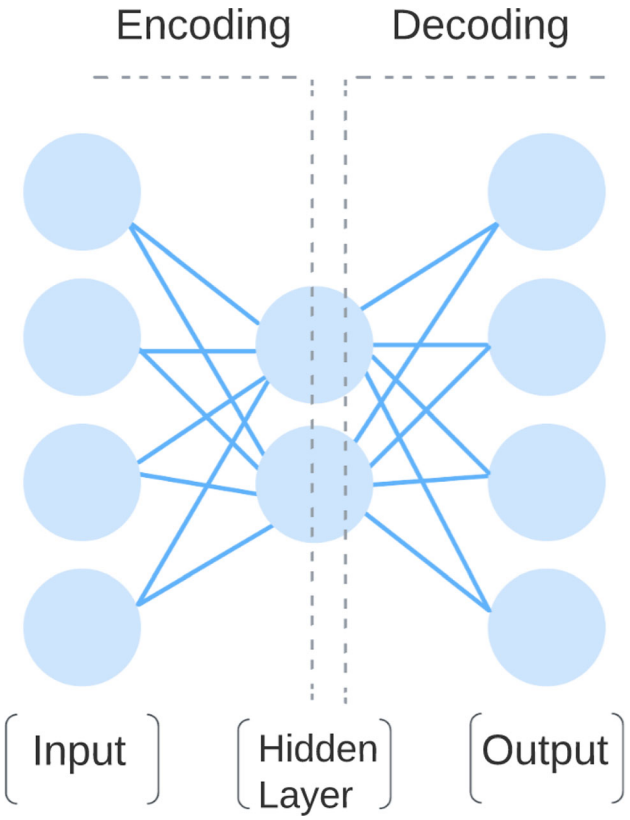

Fig. 6 The architecture of Auto Encoder (AE)

of the data. The SOM has the input and output layers, without any hidden layers. Each neuron is simultaneously given a weight value based on the input space and the weight value is passed to the output without using any activation function in neurons [73].

2.1.3 Hybrid deep learning

Hybrid deep neural networks combine multiple supervised or unsupervised deep neural networks. There are three categories based on the integration of different unsupervised or supervised models as follows:

- An integration of different unsupervised or supervised models to extract more meaningful and robust features such as CNN with LSTM and AE with GAN.
- An integration of an unsupervised model with a supervised model such as DBN+MLP, GAN with CNN, and AE with CNN.
- An integration of an unsupervised or supervised to a non-deep learning classifier such as AE with Supper Vector Machine (SVM) and CNN with SVM.

2.1.3.1 Deep transfer learning (DTL) DTL (illustrated in Fig. 7) aims to eliminate the need for training data and test

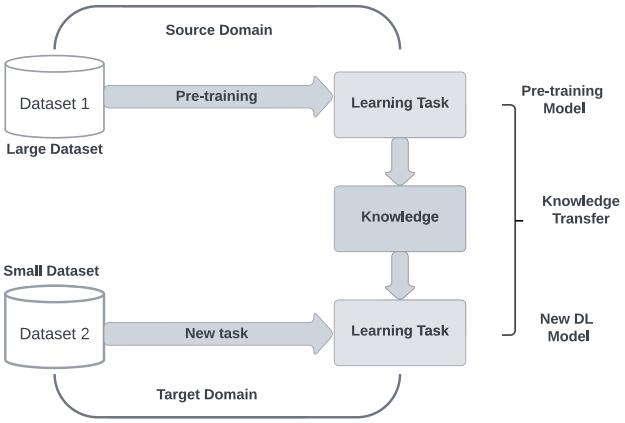

Fig. 7 The demonstration of Transfer Learning (TL)

data to be independent and identically distributed. As a result, there is no need to train the model from scratch in the target domain, which can significantly minimize the need for training data and training time in the target domain. Since DL models typically require significant computational resources, DTL can tackle this issue by obviating the need for training a new model for every task/domain [74]. Yet, it may not be applicable to all domains and may need human expertise and manual feature extraction/transformation to perform well.

2.1.3.2 Deep reinforcement learning (DRL) In Reinforcement Learning (RL), an agent chooses actions using a policy that performs them. These actions result in positive or negative outcomes through which the agent learns optimal actions [75]. When RL uses Deep Neural Network to estimate the policy it is called Deep Reinforcement Learning. Figure 8 shows the conceptual architecture of DRL.

2.2 Software defined networking (SDN)

SDN adopts centralized network management and routing approach to facilitate network management and optimize routing decisions. SDN architecture is composed of three layers (i) application layer, (ii) control layer and (iii) data layer as illustrated in Fig. 9. The data plane (also known as

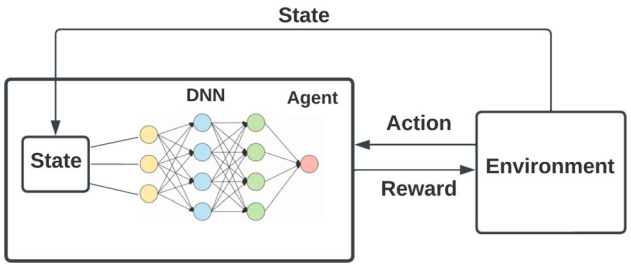

Fig. 8 A representation of Deep Reinforcement Learning (DRL)

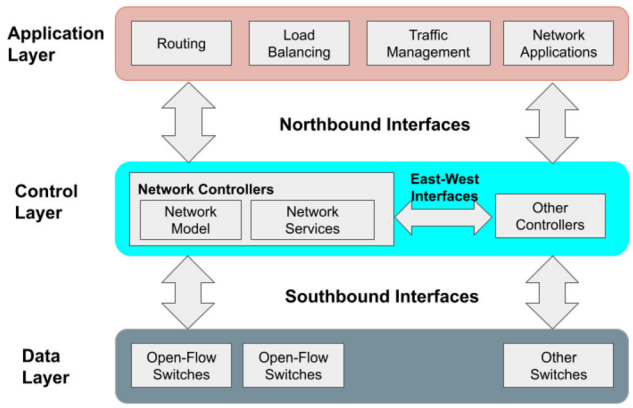

Fig. 9 Basic three-tiered model of Software Defined Networks (SDN). The tiers are called application layer, control layer, and data (infrastructure) layer [43]

the infrastructure plane) is the is comprised of hardware or software-based forwarding devices which can forward, drop, and modify packets depending on policies defined by the control plane. The control plane manages the data plane's processing and forwarding functionalities (i.e., southbound communication), thus it can be considered the "brain" of the SDN architecture. It can accept requests from applications (i.e., northbound communication) and defines forwarding rules to meet the requirements of the requests. It can also gather performance statistics from data layer devices to monitor network traffic. The application layer is composed of applications that can communicate to the control layer to specify networking demands of applications such as bandwidth and delay, such that the control layer configures the data layer in a way that application requirements can be met.

3 SDN security analysis: architectural vulnerabilities and attack types

This section will attempt to create a foundation based on earlier studies and cover aspects not discussed in earlier review-based studies. Specifically, the use of DL methods for SDN security is a relatively recent topic, thus we will focus on presenting attack scenarios as well as proposed DL-based methods to identify and/or overcome these attacks.

Figure 10 outlines some of the major attacks that threaten the different parts of SDN architecture. While previous survey studies analyzed some of these attacks [32, 34, 35, 38, 53], they typically do not cover all layers of SDN. For example, [53] presented a broader perspective on SDN security without detailing architectural breakdown. [35] used the STRIDE (Stride, Tampering, Repudiation, Information Conflict of interest, Denial of Service, Elevation of Privilege) framework to evaluate the different security issues and proposed solutions in the existing literature. This framework makes it challenging to identify which part of SDN architecture is most vulnerable to which types of attacks. Furthermore, the individual metrics within STRIDE are not defined or discussed fully to understand their relationship to security issues and particular attack types. [34, 37, 38] provided a layer-wise assessment of attack vulnerabilities and types as well as related studies. Similarly, this study will attempt to pair each SDN architectural component with associated attack types (e.g., DoS, DDoS, spoofing, and packet injection) and proposed solutions for each attack type to provide a holistic overview of the security vulnerabilities in each component of SDN. Please note that we focus on attack types that researchers applied DL to address them. Hence, not all attack types are presented in the following sections.

3.1 Vulnerability analysis for control plane

This section will focus on highlighting some of the significant security issues that threaten the safety of information and data within the control plane. Table 3 lists the vulnerabilities, proposed solutions as well as their merits.

3.1.1 Denial of service (DoS/DDoS) attacks

DoS/DDoS attacks intend to overwhelm network resources by generating a large volume of illegitimate requests thereby preventing network service delivery to legitimate users. In the control plane context, the purpose of DoS/DDoS attacks is to prevent the control plane function properly by overwhelming it with fake requests. By doing so, legitimate requests from application or data layers are either delayed or completely dropped due to a lack of resources. A considerable amount of effort has been put by researchers to develop effective DoS/DDoS detection and mitigation solutions [5, 6, 8, 9, 92-95]. Likewise, most of the studies in DL for SDN security area are primarily focused on DoS/DDoS detection and mitigation. In this context, the purpose of a DDoS attack is very similar to a DoS attack. Only the nature and intensity of the attack differ from a DDoS attack.

Wang et al. discussed the development of SGuard, a lightweight and efficient security mechanism, against DoS attacks [3]. SGuard has two main components as access control and classification components. The access control component checks authorization information for packet source tracing to perform gate-keeping by dropping packets from spoofed sources. The classification component uses Self-Organizing Map-based ANN to separate normal and malicious traffic flows. Future studies should attempt to reduce system training time and practical assessment of

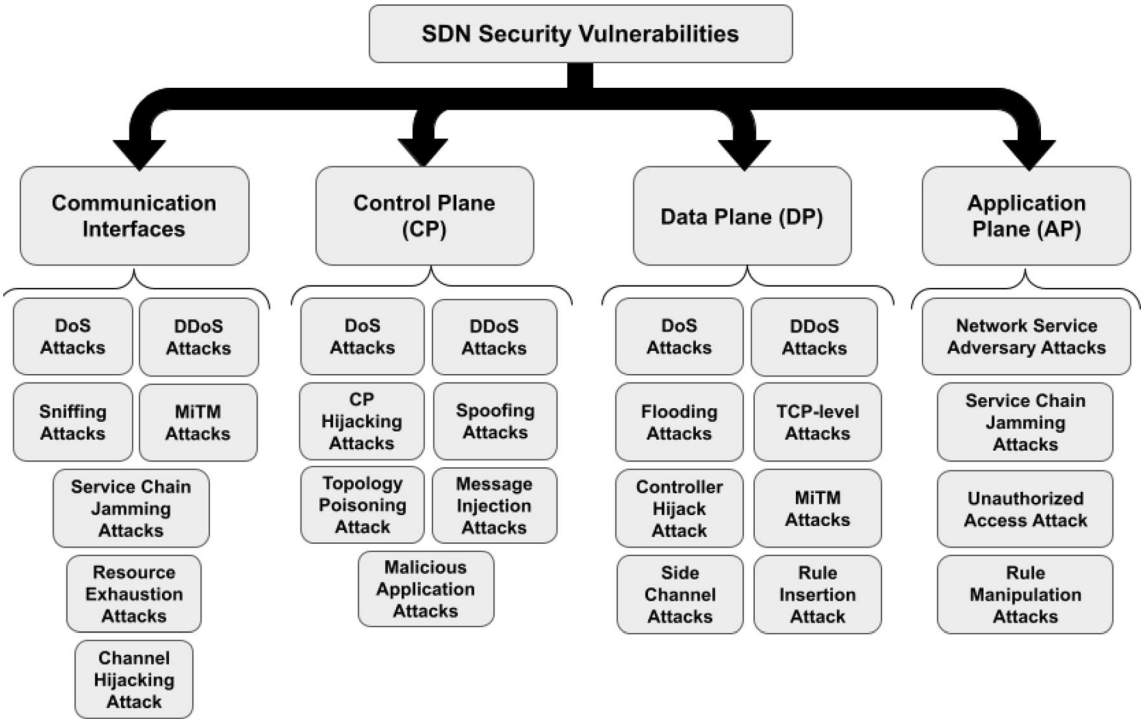

Fig. 10 Some of the major security vulnerabilities in Software Defined Networks

SGuard on complex, functional network topologies. Jia et. al. combine LSTM with SVM to process time-series flow characteristics for DDoS attack detection in a Space-based network with SDN [77]. SVM is used to solve the misclassification problem caused by the sensitivity of the LSTM model during the network startup phase. The proposed solution is shown to reduce the attack detection time as well as the system overhead.

Alanazi et al. proposes a DL-based ensemble solution for DDoS attack detection by adopting three ensemble techniques and different DL architectures such as CNN, LSTM, and GRU. Experiments are conducted using flow-based dataset CIC-IDS2017. The results showed high attack detection accuracy (99.77%) using a small number of flow-based features [78].

Performance and Features (*P and F*) is a lightweight, real-time framework to detect and mitigate Low-rate DoS (LDoS) attacks [7]. It uses Gradient Boosting Decision Tree-based ML method to analyze the performance of normal traffic under attack state and then to detect the presence of LDoS attacks. Moreover, *P and F* can also locate attack sources and victims according to flow features (F) of LDoS attacks based on time-frequency analysis. Shin et al. proposed AVANT-GUARD to reduce southbound (between the controller and network devices) communication requirement in OpenFlow protocol, thereby minimizing DoS attack surface [4]. Results reveal the effectiveness of AVANT-GUARD against TCP SYN flooding and network scanning attacks. However, it is unable to protect other parts of SDN against DoS attacks. A probabilistic model for proxying and blocklisting network traffic is used in LineSwitch, a method to prevent DoS attacks from affecting the control plane and preserving network functionality [7]. Results from experiments under different traffic and attack scenarios reveal the considerable promise of LineSwitch in comparison with AVANT-GUARD. However, there are some practical issues with LineSwitch activation, such as jitter, which can cause problems in the delivery of high-speed services (such as Voice-over-IP (VoIP)).

Yuhua et al. proposed an unsupervised learning-based method using K-means++ and Fast K-Nearest Neighbors (K-FKNN) for DDoS detection using NSL-KDD dataset [96]. In another study, a Gated-Recurrent Unit (GRU) based Deep Learning (GRU-DL) model is combined with fuzzy logic to detect and mitigate a variety of DDoS attacks by exploiting commonly used applications and services (e.g., MSSQL, NetBios, SSDP, and UDP). This system was tested by a real-world dataset and an emulated traffic and the results were compared with DNN, CNN, and LSTM methods and show GRU has similar results to CNN and LSTM but better than DNN [80]. A back-propagation neural network-based deep reinforcement learning DDoS mitigation strategy was developed in [76], which leveraged multiple controllers and two modules (namely anomaly detection and dynamic defense). The anomaly detection

Table 3 Security vulnerabilities and proposed solutions for SDN control plane

Attack type	Attack objective	Proposed solution	Advantage	DL/ML method
DoS/DDoS	Exhaust resources	Safeguard scheme [76]	Uses multiple controllers	Back propagation neural network
		SNB-SDN [77]	Reducing attack detection time	Long short term memory
			Using small number of	Long short term memory
		Ensemble DL [78]	flow-based features	Convolutional neural network
				Gated recurrent unit
		Deep CNN-Ensemble [79]	Efficient/Scalable framework	Convolutional neural network
		GRU-DL [80]	Mitigates multiple DDoS attacks	Gated recurrent unit
		sF-APS-DL [81]	Effective performance	Stacked autoencoder
		LSTM-CSSD [82]	High performance framework	Long short term memory
	Limiting access to		Enhanced performance using	Long short term
	network services	DO-IDS [12]	unbalanced datasets and optimized	Memory-based autoencode
			one-class detection algorithm	
		GRU-DL [80]	Mitigates multiple DDoS attacks	Gated recurrent unit
		RNN-LSTM [83]	Good trade-off betweenprecision and recall metrics	Long short term memory
		CIDS [84]	Detect abnormal network behaviors at entire VANET	Generative adversarial network
		LEDEM [85]	Prevent saturation issues	Extreme learning machine
		Rl-shield [86]	Mitigate attack by persistent re-routing	Deep reinforcement learning
		CyberPulse [87]	Classify malicious flows with high accuracy	Multi-layer perceptron
		CNN ensemble Framework [79]	Scalability and Cost-effectiveness	Convolutional neural network
		DLSDN [88]	High accuracy using various DL methods	Stacked auto-encoder multi-layer perceptron
		SD-IIoT [89]	Generating real-like network traffic	Deep reinforcement learning
Topology Poisoning	Limiting network visibility	MLLG [90]	Prevent Link Fabrication	Multi-layer perceptron
		DRL-ATS [91]	Effective recovery mechanism	Deep reinforcement learning
Packet Injection	Network service disruption	SD-IIoT [89]	Generating real-like network traffic	Deep reinforcement learning

SNB-SDN space-based network based software-defined network, *CNN-EF* CNN-ensemble framework, *GRU-DL* gated recurrent unit deep learning, *sf-aps-dl* sFlow and adaptive polling sampling for deep learning, *LSTM-CSSD* long short-term memory-based collaborative source-side DDoS, *DO-IDS* deep learning-based one-class intrusion detection scheme, *CIDS* collaborative intrusion detection system, *LEDEM* learning-driven detection mitigation, *DLSDN* deep learning for SDN, *MLLG* machine learning-based link guard, *DRL-ATS* deep reinforcement learning-based attack tolerance scheme

module focuses on switches in the data plane for traffic flow classification. The controller dynamic defense module mitigates DDoS attacks by remapping the controller and sending the access control message to switches. Another study used sFlow and adaptive polling sampling with Snort IDS to identify the network traffic, and then classify the traffic using Stacked AutoEncoder (SAE) as benign and malicious traffic in IoT networks [81]. In the data plane, sFlow and adaptive polling-based sampling are leveraged to capture network statistics. In the control plane, Snort IDS with Stacked Auto-Encoders (SAE) Deep Learning model is used to detect DDoS attacks. The proposed solution attains 95% true positive and 4% false positive rates, but incurs has high computational overhead.

Hu et al. proposed Deep One-Class Intrusion Detection System for DoS detection in industrial SDNs using an LSTM autoencoder. The authors first collected network flow features into vector sequences and inputted them into an encoder for the encoding process. The encoding result is sent to a decoder, and a scorer, respectively. The decoder outputs the reconstructed feature vector sequence. The scorer calculates and returns a novelty score for encoding which is compared to a predefined detection threshold to detect the presence of abnormal traffic. This system is

evaluated using two public datasets, KDD99 (data with 14 attack types) and NSL-KDD (data with 16 attack types). The results of this system show high detection speed and enhanced accuracy [12]. A recent study developed an early warning system for intrusion detection in SDN using the Deep CNN network that detects DDoS attacks. Using 11 features from the dataset InSDN, the proposed method was able to provide an accuracy of 100% [13]. A comparison of different ML and DL frameworks was presented for intrusion detection systems in SDNs. The LSTM-FUZZY system presented in this work has three distinct phases: characterization, anomaly detection, and mitigation. Out of the different ML and DL systems, the LSTM-based network was able to provide the highest performance (accuracy greater than 98.0%) on two different flow datasets (e.g., CICDoS 2017 and CICDDoS 2019) [97]. For SDN-IoT applications, a novel Deep Learning-based three-tier intrusion detection and mitigation system was developed in another research. The proposed approach was evaluated based on different network simulation scenarios, which provided detection accuracy results of 97.7% [98].

In another study, LSTM and CNN models are used for DDoS (TCP, UDP, and ICMP flooding) detection and mitigation that can be launched against SDN controllers. The training dataset comprising normal and malicious (DDoS) traffic is collected using Mininet. The performance of the deep learning models is compared against classical machine learning models (KNN, NB, and SVM) to show that the LSTM model produced higher performance than ML methods [83]. For distributed SDN-based VANETs, a GAN (Generative Adversarial Networks)-based solution for DDoS detection has been developed in [84]. It enables multiple SDN controllers to collectively train a global IDS for the entire network without directly exchanging sub-network flows. A semi-supervised Deep Extreme Learning Machine (SDELM) was developed in [85] for DDoS detection and mitigation for SDN-managed IoT networks. The performance of the proposed method is compared against other learning-based methods (e.g., AdaBoost, SVM, Deep Belief Network, Naive Bayes) to demonstrate its superior performance.

Rezapour et al. proposed a novel Deep Reinforcement Learning-based approach, RL-Shield, for Link Flooding Attack (LFA) detection and mitigation [86]. Using different attack strategies (e.g., persistent attack, strategic attack, and collusion attack) on four separate real network topologies (e.g., ARPANet, NSFNet, Evolink, and GTS-CS), authors demonstrated that RL-Shield outperforms other routing-based approaches such as GKLD [99]. CyberPulse is a DL-based method for flooding attack mitigation on SDN control channel [87]. A comprehensive review has been conducted to assess the performance of deep learning classification, and a side-by-side comparison has been made against other solutions to show that CyberPulse can classify malicious flows with high accuracy and mitigate them effectively.

Ahuja et al. applied DL models to detect DDOS attacks such as TCP-SYN, UDP flooding, and ICMP flooding attacks. Authors generated a custom dataset to train the models [100]. The results show 99.75% accuracy by applying a Stacked Auto-Encoder Multi-layer Perceptron (SAE-MLP) that is comprised of several autoencoders tied together where the output of one autoencoder is fed to the input of another, and a SoftMax classifier also is used for feature extraction [88]. Wang et al. proposed a deep reinforcement learning (DRL)-based attack tolerance scheme to let regular traffic bypass forwarding nodes and controllers in software-defined IIoT (Industrial Internet of Things) that have been targeted by DDoS attacks (e.g., packet flooding, packet injection, link fabrication, etc.). The proposed approach uses a generative adversarial network (GAN) to flexibly generate real-like network traffic for more sufficient and effective experimental verification of the attack tolerance scheme. Experimental results show that the proposed scheme can significantly improve the successful arrival rate of IIoT traffic and achieve near-optimal results [89].

3.1.2 Topology poisoning attacks

In this attack type, the attacker hijacks the control plane resources and prevents the controller from accurately assessing the network topology. Due to the lack of effective authentication mechanisms in the SDN control plane, the legitimacy of topology information, including switches, hosts, and internal links cannot be verified reliably [44]. Furthermore, the existing mechanisms are simple and predictable thus attackers can easily hack the system and gain unauthorized access. In order to provide reliable protection from these attacks, a number of studies have proposed reliable and robust security mechanisms. An attacker can forge link layer discovery protocol (LLDP) packets to create a fake link (link fabricating attack) [89]. If the controller updates the network topology according to forged LLDP packets, it will have a false network view and make wrong decisions based on it. Jiadai et al. introduced a deep reinforcement learning (DRL)-based attack tolerance scheme to separate actual traffic from attack traffic such as link fabrication that targets forwarding nodes and controllers in software-defined IIoT (Industrial Internet of Things) networks. TopoGuard [101] avoids poisoning attacks by assigning a special attribute to active ports of all switches, which enables the controller to identify the origin of the traffic. It uses an encryption-based authentication mechanism to avoid spoofing attacks. One of the limitations of TopoGuard is that it is unable to protect against

relay-type link fabrication attacks. Consequently, TopoGuard+ was developed after two major vulnerabilities of TopoGuard against new topology attacks were discovered, namely port amnesia and port probing [102]. TopoGuard+ checks for suspicious port reset events and tracks the latency of inter-switch links to detect link fabrication attacks using topology packets leveraging out-of-band channels.

In another study, the Link Latency Attack (LLA) is introduced in which an adversary can add a fake link into the network and corrupt the controller's network topology view [90]. This attack can compromise the end hosts without the need to attack the SDN-enabled switches. The authors developed an ML-based Link Guard (MLLG) framework to defend against LLAs. Results show an accuracy of 98.22% in detecting the attack. MLLG outperformed TopoGuard+ by 16% in accuracy. A Deep Reinforcement Learning-based Attack Tolerance Scheme (DRL-ATS) for SDN-based Internet-of-Vehicles (IoVs) has been discussed in [91]. The scheme can tolerate topology poisoning attacks by adaptively adjusting the service deployment in the SDN-enabled vehicular edge network.

3.1.3 Packet injection attacks

This is a relatively recent vulnerability for SDN thus received little attention in the literature. Yet, it can have a severe impact on different parts of SDN architecture by injecting manipulated packets into SDN. It can lead to service disruption in the control plane along with an increase in resource consumption in the data plane [103]. Wang et al. uses the SDN controller's improper exception handling to disconnect the control channel. Specifically, it tampers the packet-in message's length field and sends it to the controller. The authors propose a DRL-based attack tolerance scheme to distinguish real traffic from fake ones [89].

3.2 Vulnerability analysis of data plane

Most common data plane attack types include DoS/DDoS, unauthorized access, side-channel attacks, fraudulent rule insertion, flow-rule flooding, and ARP poisoning [34, 37, 38]. In this study, we present Denial-of-Service (DoS/DDoS), and side-channel attacks based on the availability of DL-based methods.

3.2.1 Denial of service (DoS/DDoS) attacks

In DoS/DDoS attacks, the attacker generates traffic flow that overwhelms the links between switches in the data plane and creates a bottleneck to prevent network service delivery to legitimate users and their generated traffic. It is for this reason, DoS/DDoS attacks affect multiple parts of SDN architecture and cause considerable damage in terms of service delivery. Table 4 highlights different DL-based solutions that target data plane attacks.

According to [112], DoS attacks targeting SDN data planes can be classified into two groups, M-DoS and S-DoS. Multiple flow entries (M-DoS)-based attacks intend to exhaust the Ternary Content-Addressable Memory (TCAM) resource of the switch. Single entry (S-DoS) attacks, on the other hand, target a link between the network device and the SDN controller. In this respect, the authors proposed a Back-Propagation Multi-feature-based Neural Network (BP-MFDD). Results reveal that BP-MFDD is able to effectively detect both types of attacks. However, the computational time of the proposed mechanism increases with an increase in flow rate, which is one limitation that future work can attempt to rectify.

Razib et al. developed a DNN-LSTM-based method for intrusion detection system in SD-IoT networks [105]. Three different DNN-LSTM variations (namely Cu-DNNLSTM, Cu-DNNGRU, and Cu-BLSTM) are implemented and validated using CICIDS 2018 dataset. Their performance is also compared to state-of-the-art solutions that utilize GRU-LSTM, GRU-RNN, and Generalized Suffix Tree models. However, the authors fail to provide an attack-level performance evaluation for this system. A Double Deep Q-Network-based intrusion response algorithm developed in [104] to provide rapid response against TCP SYN Flooding and Link Layer Flooding attacks. The performance of the proposed method for detecting the attacks is evaluated using metrics such as the ratio of malicious packets dropped and the number of flow rules. Ali et al. analyzed current strategies for malicious traffic identification and listed the shortcomings of the classical machine learning method. The authors showed that DNN attains higher accuracy. The presented models were evaluated using KDD-CUP99 and NSL-KDD datasets and obtained 99.6% accuracy on attack detection [113].

Thu et al. proposed Deep Deterministic Policy Gradient (DDPG)-based Deep Reinforcement Learning approach for DDoS detection and mitigation system in Software-defined satellite networks (SDSNs) [106]. The proposed model is composed of an actor-network and a critic network. The actor-network adjusts policy parameters, and the critic network evaluates the policy function of the actor-network using a time-difference error. The performance of the proposed DDoS mitigation system is compared against different types of DL models such as LSTM, Densely Connected Network, and Gated Recurrent Unit in terms of power consumption for normal, abnormal, and total traffic. A semi-supervised Deep Extreme Learning Machine (SDELM) was developed in [85] for DDoS detection and

Table 4 Security vulnerabilities and proposed solutions for SDN data plane

Attack type	Attack objective	Proposed solutions	Advantage	DL method
		Deepair [104]	Superior compared to the Q-learning based approach	Deep reinforcement learning
		DNNLSTM [105]	Efficient threat detection, high accuracy and low resource consumption	Deep neural network and long short term memory
		SDSN [106]	Reduce energy consumption	Deep reinforcement learning
		DNN-IDS [107]	High efficiency	Deep neural network
		LEDEM [85]	Successful working in a real hardware network	Deep belief network
DoS/ DDoS	Network service disruption	Hybrid framework [108]	High bandwidth and low latency	Restricted Boltzmann machine
		LSTM-Autoencoder-based [109]	High performance of controller	Long short term memory
				Convolutional Neural Networks
		DDoS defender [110]	Easy real-time update of detection	Long short term memory and recurrent neural networks
		DCNN [79]	Minimal computational complexity	Convolutional neural networks
		Distributed SDIN [50]	Distributed control plane	Deep reinforcement learning
		LSTM-FUZZ [97]	Automatic execution of activities	Long short term memory
		DeepGuard [1]	Enhanced traffic flow monitoring capability	Deep reinforcement learning
		IDPS [111]	High performance of controller	Recurrent neural networks
		DLSDN [88]	High accuracy using various DL methods	Multi-layer perceptron

mitigation for IoT using SDN-Cloud architecture. The performance of the proposed method was compared against other ML/DL methods (e.g., AdaBoost, SVM, Deep Belief Network, Naive Bayes) to demonstrate superior performance.

A hybrid Deep Learning framework with Restricted Boltzmann Machines (RBM) and SVM was utilized for anomaly detection in multimedia applications in SDNs [108]. The proposed system was evaluated on different datasets (e.g., KDD99 and custom dataset) and it is shown to attain a detection accuracy rate of 99.0% for KDD99. It also obtained more than 90% accuracy for the custom-generated dataset in detecting stealthy DDoS and volumetric DDoS attacks. ElSayed et al. proposed an LSTM-Autoencoder-based model for DDoS attack detection and mitigation. This paper uses two popular feature selection methods as Information Gain (IG) and Random Forest (RF) to find the most relevant features for each dataset (InSDN, CICIDS2017 and CICIDS2018). The comparison against other ML/DL models (e.g., SVM, Naive Bayes, Decision Trees, Random Forest, Logistic Regression, and DDoSNet [114]) showed the proposed approach provides a high attack detection rate, more efficient training time, and lower overhead on network performance [115]. Scaranti et al. developed anomaly detection models using Using Artificial Immune System (AIS) and Fuzzy Logic models [116]. The proposed solution is tested with experimental and open-source datasets (i.e., CiCDDoS2019) and compared against state-of-the-art solutions. The results indicate that AIS was able to surpass other ML-based techniques (e.g., Naive Bayes, Random Forest, K-NN, and Local Outlier Factor) for most of the evaluation metrics. Another study compared the performance of ANN with various classic ML algorithms (e.g., Logistic Regression, Support Vector Classifier, KNN, Random Forest, Ensemble Classifier, and Support Vector Classifier with Random Forest) for DDoS detection. Results show that the hybrid model of the Support Vector classifier with Random Forest has the highest accuracy [117]. Tang et al. introduced a Gated Recurrent Unit (GRU)-based approach for DDoS attack detection in SDNs [11]. The proposed model achieved higher accuracy compared to other ML-based approaches (e.g., Naive Bayes, SVM, and DNN) using six features from the NSL-KDD dataset. Another study [110] built an efficient defense mechanism using three different DL algorithms against DDoS attacks in SDNs. The proposed models are evaluated on the ISCX dataset and they were found to attain 98% accuracy. Novaes et al. used Generative Adversarial Network (GAN) framework to alleviate the impact of DDoS attacks in SDNs. The authors compared the obtained results from the GAN framework with different DL algorithms (e.g., LSTM, CNN, MLP) to

highlight superior performance using the public dataset such as CiC-DDoS2019) [118].

To classify DDoS attacks based on IP flow traffic, [79] utilized an ensemble Deep CNN framework. The performance of the proposed network was compared against other state-of-the-art solutions using the CICIDS2017 dataset. In various performance metrics, the proposed solution was able to obtain 99.4% detection accuracy. For industrial SDNs, an attack mitigation scheme using Deep Reinforcement Learning and distributed control plane has been proposed in [50]. The performance of the proposed system is tested against different types of attacks (e.g., topology forgery and packet in flooding attacks scenarios) and performance metrics (e.g., detection time). The proposed system can isolate the attack source by adaptively adjusting the switch takeover and can achieve near-optimal performance. The use of a hybrid deep learning framework was discussed in another study [97]. The framework combined LSTM memory cells and Fuzzy Logic networks to prevent different types of attacks including DDoS and port scan attacks. The framework was able to provide a high level of performance (Recall = 93.13% for one of the attack scenarios) on CICDDoS 2019 dataset. A Double P-value of Transductive Confidence Machines for the K-Nearest Neighbors algorithm (DPTCM-KNN) was proposed for detecting anomalous flows in SDNs [119]. Although the attack-level analysis is missing, the system-level analysis and comparison with other methods (e.g., TCM-KNN, KNN-ACO, and ABT-SVM) using different metrics (e.g., accuracy, FPR, TPR) returned promising results.

A Sampled Density Peak-based Clustering algorithm is leveraged to develop Anomaly Detection System (ADS) in [120]. The performance of the proposed system is compared to other clustering-based approaches (e.g., DBScan, Birch, K-means, MeanShift) to reveal higher detection accuracy. FADE [121] is an ADS that analyzes the network topology and flow rules and computes a minimal set of flows that covers all forwarding rules. It then calculates an optimal number of monitoring positions on its path and installs dedicated rules for flow statistics collection. FADE also manages the expiration of existing rules and the installation of updated rules to ensure the accuracy of collected statistics. The authors also proposed a scalable version of FADE, called iFADE, that can reduce the number of rules by 40% when compared to FADE. FADE and iFADE are lightweight ADS models, as they reduce the overhead of control messages by about 50%–90% compared to state-of-the-art solutions such as SPHINX [122]). Another study utilized a Deep Reinforcement Learning-based system (DEEPGUARD with Double Deep Q-Network (DDQN)) for anomaly detection and effective traffic flow management in SDNs. The learned optimal traffic flow matching control policy allows the extraction of useful traffic information that improves anomaly detection [1]. Shafi et al. developed an ADS for Fog-Assisted DDoS section framework for IoT networks, E3ML, using Recursive Neural Networks (RNN), Multi-Layer Perceptron (MLP), and Alternate Decision Trees (ADT) to detect DoS attacks [111]. The models are tested using UNSW-NB15 dataset and the results show that ADT model yields the best performance (near 100% precision and recall) when the attack detection interval is set to 60 seconds or more. On the other hand, the proposed hybrid model, E3ML, attains better results at shorter detection intervals and comparable results when using higher detection intervals.

Musumeci et al. investigates how data plane programmability, enabled by P4 language, can be used to detect DDoS attacks directly on network switches with the need to involve SDN controllers. It investigates the potential of using ML models to perform automated DDoS detection such as SYN flooding attack detection. The results show that all ML algorithms achieved more than 98% in accuracy, precision, recall, and F1-scores. Classification time remained in the order of a few hundred microseconds, indicating a significant reduction in detection latency when models are evaluated in the data plane [123]. pHeavy is another ML-based method for P4-enabled devices to detect elephant flows which can be used for DoS attack prediction in the data plane [124]. Thus, the results show that pHeavy can predict heavy flows accurately. Euclid is also a p4-based method that utilizes information-theoretic and statistical analysis for DDoS attack detection and packet classification [125]. This paper uses the CAIDA [126, 127] dataset to evaluate the proposed mechanisms and the results show that Euclid can detect attacks with high accuracy (98.2%) and low delay (around 250 ms).

3.2.2 Side-channel attacks

In side-channel attacks, attackers use various methods (analyze the time gap or the flow configuration delay in the flow tables) to access confidential information such as network configuration [38]. Previous research analyzed the intricacies of the attacks and the level and extent to which these attack types can cause SDN security vulnerabilities. For example, different fingerprinting-based techniques have been developed to explore and exploit the different weaknesses such as the time gap in flow table update messages. In this respect, a time-based fingerprinting method was developed in [128]. Cui et al. used dispersion in packet pairs sent between controllers to switches to deduce flow rules, network configuration, and controller type [129-131].

Interestingly, most studies on data plane side-channel attacks are focused on finding new attack surfaces in the existing SDN implementations. FlowKeeper is one of the very few studies that attempt to robustify the SDN data plane to mitigate attacks including side-channel attacks [132]. It introduces a dynamic delay adjustment method for communication between switches and a controller to prevent side-channel attacks without hampering the performance of benign traffic considerably. Another research proposed an integrated SDN-IoT architecture that can detect side channel attacks in IoT networks [133]. In this architecture, a Fuzzy Neural Network (FNN) based attack detection system is developed using NSL-KDD datasets. The proposed model uses the value of signal noise ratio and sudden lack of memory as features to detect the attacks. The confusion matrix analysis reveals that the proposed FNN-based attack detection system can detect the attacks with an accuracy of 83%.

3.3 Vulnerability analysis of application plane

Application plane is susceptible to different attacks including service neutralization, application termination, and communication disruption [34]. Singh et al. mentioned fraudulent rule manipulation, network service adversary, and unauthorized access into the list [38]. Rahouti et al. added malicious application-based attacks and authorization-related attacks to the list of vulnerabilities [37]. Hence, it can be said that control and data plane security vulnerabilities and proposed solutions have been given considerable attention in the literature. In comparison, application layer and communication interface have not been studied sufficiently. This is mainly because SDN controllers (control plane) and switches (data plane) are considered more critical components that regulate and manage flows. Consequently, Table 5 merges the vulnerabilities for application plane and communication interfaces.

Application layer flooding attacks targeting higher layers of the protocol stack such as DNS, NTP, SNMP, and HTTP are widespread. VARMAN (adVanced multi-plAne secuRity fraMework for softwAre defined Networks) is an Intrusion detection system that used Poisson distribution for the traffic generator and replayed the collected traffic trace files using the fprobe tool to target HTTP services. The proposed ML workflow for attack detection and mitigation in the SDN controller consists of two main types of traffic flow classifiers, namely Non-Symmetric, Stacked, Deep Autoencoder Learning, and Random Forest Classifier. VARMAN detects the application layer attack traffic patterns by matching the request/response packets and identifying malicious behavior in the protocol semantics [134].

Griffin is an unsupervised learning-based (Deep Autoencoders) IDS for the detection of known and zero-day intrusion attacks in real time [14]. Griffin employs feature extraction, cluster analysis for dimensionality reduction, and an ensemble autoencoder to further extract features with low complexity and high precision before training a model. The authors state that the application plane of SDN is vulnerable to reconnaissance attacks in which malware finds devices that use the default username and password combinations. Griffin has been evaluated with the open datasets (e.g., Mirai, Active Wiretap) to demonstrate its detection effectiveness. The results show that Griffin's complexity is about 40% lower, and its accuracy is up to 19% higher than existing NIDSs (e.g., flow-level clustering, SVM, Random Forest, Suricata). INDAGO [136] is a proactive mechanism that statistically analyzes the application behavior using an unsupervised learning-based K-means clustering algorithm and automatically detects malicious activities in applications. They implement Security Sensitive Behavior Graph (SSBG) used for extracting the control plane semantic features and behavior profiling [136]. However, one of the limitations of INDAGO [136] is that it does not provide real-time malicious application detection.

Table 5 Security vulnerabilities and proposed solutions for application plane and communication interface in SDN architecture

Attack type	Attack objective	Proposed solutions	Advantage	DL method
Application plane				
Flooding attack	Higher layers of protocol stack	VARMAN [134]	Real-time detection	Deep Autoencoder Learning
Reconnaissance Attack	Unauthorized Access	Griffin [14]	Real-time and high accuracy	Deep Autoencoder Learning
Communication interfaces				
MiTM	Data sniffing	Griffin [14]	Real-time and high accuracy	Deep autoencoder Learning
		CNN-LSTM [135]	High detection speed	CNN and LSTM

3.4 Vulnerability analysis of communication interfaces

Some of the attack types discussed by [38] for the different communication interfaces include man-in-the-middle, sniffing, resource saturation, and service chain jamming. Some of the cross-layer attacks mentioned in [37] include cross-path attacks, teleportation attacks, rootkit attacks, and controller placement-related attacks. Cross-path attack is another type of attack that the communication interfaces are vulnerable to. In a cross-path attack, the attacker's objective is to disrupt the communication between data plane devices and controller [137]. The attacker can achieve this goal by launching a saturation attack that sends a vast number of spoofed packets to reduce the possibility of matching any of the existing flow entries on the targeted switch [138]. Such a data-to-control plane flooding attack may exhaust the controller's computation resources thereby disrupting the effective operation of OpenFlow switches [138]. To tackle this problem, authors in [138] investigate the impact of measurement time-window on attack detection performance for SVM, Naive Bayes (NB), and K-NN classifiers. The NB model obtained the highest performance with 85% precision, 96%, 91% F-1 scores using one minute as an attack detection time window.

Man-in-the-Middle (MiTM) attacks can threaten one or more of the communication channels and interfaces between the different layers of the SDN architecture. In MiTM, an intruder injects a host between source and destination communication to sniff the data during communication [38]. Griffin [14], a Network Intrusion Detection System, uses an ensemble of autoencoders to detect both known and zero-day intrusion attacks in real-time with high accuracy. Specifically, Griffin uses an efficient feature extraction framework to capture the sequential features of the traffic packets. It shows higher accuracy in detecting MiTM attacks compared to Support Vector Machines and Random Forest models.

Ahuja et al. developed ML models to detect ARP Poisoning and ARP Flooding attacks [135]. The CNN-LSTM model outperformed the other machine and deep learning algorithms tested including LR, NB, SVC, DT, RF, EC, ANN, CNN, LSTM, and CNN-LSTM, with an accuracy score of 99.73% and a false positive rate of 0.017%. Another research proposed to detect side channel, man-in-the-middle, malicious code, and DDoS attacks using Fuzzy Neural Network (FNN) [133]. The FNN is trained and tested using NSL-KDD datasets. SSH alarms for redirecting SSH/HTTPS sessions, security certificate error messages, and phishing emails are being used to detect MiTM attacks. The confusion matrix analysis reveals that the proposed FNN-based attack detection system can detect the attacks with an accuracy of 83%.

4 Training datasets and performance evaluation metrics

In this section, three main aspects at the intersection of Deep Learning and SDN will be highlighted in three separate sub-sections. Due to the importance of data quality and quantity for training and validation of DL methods, we start with the review of some of the existing publicly-available datasets to revisit the claims that data limitation is a major challenge in the adoption of DL methods for SDN security [32]. In the second subsection, we outline the performance evaluation metrics developed to examine the performance of various security enhancement models. In the third subsection, we discuss the types of DL-based SDN security applications and provide a taxonomy based on the scope of the work.

Another interesting insight from examining the performance metrics outlined in different studies is that they do not provide an attack-by-attack performance, rather they provide an overview of the overall performance of the proposed systems. However, providing performance in this manner would allow the authors as well as other researchers to have a better appreciation of the strengths as well as weaknesses, and deficiencies of the different SDN security systems against different attack types. Similarly, most of the studies in this field do not provide a link between earlier studies and their study when providing relevant metrics to gauge the performance of their SDN security systems. This disconnect and lack of validation with respect to utilizing different statistical measures make it difficult to provide a reliable cross-examination between different studies outlining different SDN security systems. This apparent disconnect can be reduced by clearly highlighting which measures they are using (this part is present) and providing an effective rationale regarding why they are using those specific measures by mentioning relevant and credible studies that have suggested the usage of those metrics (this part is currently lacking).

4.1 Dataset for deep learning-based SDN security methods

For any learning-based method, the quality and quantity of data is the key to developing a successful solution. This is even more critical for DL-based methods as they, by design, require a considerable amount of data to ensure that the developed model is accurate and performs well with new data. Yet, earlier SDN security studies did not investigate this aspect in sufficient detail. Thus this study

provides the first in-depth analysis of the dataset used in SDN security papers. Table 6 lists the major datasets used for SDN security. Luckily, many of the public datasets [139-145] contain more than a million entries, which is a promising sign, especially for DL models that are highly dependent on quality and quantity of data. Feature set size for each dataset ranges between 14 and 83. It is important to note that feature set size is not necessarily an indicator of performance since most studies apply feature extraction to remove redundant or irrelevant attributes before training DL models. Moreover, some of the datasets provide metadata [141-147] along with raw data, while others [139, 140, 148] do not provide such accompanying information.

Existing dataset are either generated using real-world networks [141-148] or simulations [139, 140, 149]. While the dataset captured from real-world networks offer more realistic results, they typically suffer from class imbalance as many types of attacks are either detected rarely or not detected at all. On the other hand, synthetic datasets captured in simulated networks can be misleading due to failure to reflect real network behavior in terms of scale, topology, and traffic characteristics. Moreover, it is nearly impossible to assess the reliability and credibility of the generated data.

Another issue with the most available dataset is that they are collected from small, non-SDN networks; thus, trained models may not work well in large-scale production SDN networks due to differences in traffic scale and behavior. InSDN [149] is the only dataset that was generated in a simulated SDN network, but it also has similar issues due to being generated in a simulated environment using synthetic traffic. Consequently, in the absence of an open-source dataset generated from actual, physical SDNs, it is nearly impossible to establish a benchmark to test the feasibility of proposed mechanisms for real-world networks.

4.2 Performance evaluation metrics for SDN security methods

An essential aspect of evaluating a solution is presenting evaluation results using well-known performance metrics. Hence, this section presents commonly used performance metrics in existing studies. Table 7 presents performance metrics used in previous SDN security papers. True

Table 6 Open-source datasets used to train and evaluate Deep Learning models

Study	Dataset name	Year	Attack classes	Attack types	Sample size	Features	Limitations
[139]	KDD99	1998	22 train/ 15 test	DoS, Probe, R2L, U2R	5.0 M	41	Data duplication very high
[140]	Kyoto	2006	N/A	N/A	93 M sessions	24	Imbalanced class distribution, attack types not known, synthetic dataset
[148]	NSL-KDD	2009	N/A	DoS, Probe, R2L, U2R	150,000	41	Synthetic dataset
[141]	ISCX-2012	2012	4 train/ test	HTTP DDoS, SSH-BF, Infiltration	2.0M packets	N/A	Only contains HTTP traffic, limited features for ML/DL training
[142]	UNSW-NB15	2015	9 train/ test	DoS, Fuzzer, Backdoor, Shellcode, Worm	2.54 M	49	Imbalanced train/test distribution
[143]	AWID	2016	3 test/train	Flooding, Injection,Impersonation	2.4M	58	Highly unbalanced dataset
[144]	CICIDS-2017	2017	14 test/train	DoS, DDoS, Botnet, CSS, Portscan, SSH-HB, Infiltration, SQL-I	2.83M	83	Redundant records, missing class labels
[146]	CIDDS-001	2017	4 train/ test	Portscan, Pingscan,DoS, BF	325,865	14	Imbalanced dataset
[145]	CSE-CIC-IDS-2018	2018	7 test/train	Botnet, Slowloris, DoS, DDoS, CSS, SSH-BF, SSH-HB, Portscan, SQL-I	16.2M instances	83	Synthetic dataset, high class imbalance
[147]	CiC-DDoS 2019	2019	12 train/6 test	DDoS	674,463	87	Slightly Unbalanced dataset
[149]	InSDN	2020	7 train/ test	Botnet, DoS, DDoS, Web attacks, Password, BF, Probe, Exploitation	343,939 instances	83	High class imbalance, synthetic dataset

CSS cross-site scripting, *BF* brute force, *SQL-I* sequential query language injection, *HB* heartbleed, *AWID* aegean wi-fi intrusion dataset, *CIDDS* coburg network intrusion detection dataset

Table 7 Performance metrics used in different studies covered in this review-based study in relation to ML/DL-based studies

DL and ML metrics										
Study	Year	Accuracy	Precision	Recall	F-score	DR	TNR	TPR	FPR	FNR
[115]	2022	X	X	X		X				
[150]	2021	X					X	X		X
[151]	2021	X	X	X				X		X
[121]	2021	X					X	X		
[111]	2018	X	X	X	X		X	X	X	X
[97]	2020	X	X	X		X	X	X		
[120]	2017	X								
[152]	2018									
[79]	2020	X	X	X	X				X	X
[153]	2017	X	X	X						
[6]	2020			X		X			X	
[82]	2022	X				X			X	
[7]	2022	X		X					X	
[116]	2020	X	X	X	X			X	X	
[154]	2020		X	X	X				X	
[86]	2022		X	X						
[155]	2018							X	X	
[92]	2017							X	X	
[119]	2018	X						X	X	
[156]	2021	X								
[157]	2019	X	X	X	X				X	
[95]	2021	X			X		X	X		X
[158]	2022					X				
[84]	2021	X	X	X	X					
[13]	2020	X	X	X	X					
[105]	2022	X	X	X	X		X	X	X	X
[14]	2018					X				
[98]	2020	X	X			X				
[159]	2022	X	X	X	X					
[15]	2022	X	X	X	X		X	X	X	X
[160]	2020		X			X			X	
[89]	2022									
[161]	2021	X	X	X	X					
[162]	2018	X	X	X	X		X	X	X	X
[134]	2019	X	X	X	X	X		X	X	X

FPR false positive rate, *TPR* true positive rate, *DR* detection rate, *TNR* true negative rate, *FNR* false negative rate

Positive (TP) signifies how many positive class samples a model predicted correctly. True Negative (TN) refers to how many negative class samples a model predicted correctly. False Positive (FP) and False Negative (FN), on the other hand, indicates incorrect positive or negative predictions by a model, respectively. Accuracy is the simplest metric to implement and is defined as the number of correct predictions made as a ratio of all predictions made. It works well only if there are an equal number of samples belonging to each class and can be calculated with the help of the following formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Precision explains the percentage of correctly identified positive samples to the total number of positive predictions. It is calculated as follows:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Recall, on the other hand, refers to the ratio of correctly predicted samples to all positive samples in the dataset.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

Finally, F-Score is the harmonic mean of precision and recall values and shows how precise and robust a model is.

$$F - Score = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{4}$$

Accuracy, precision, recall, and F-score value are all relevant and important metrics when evaluating the performance of ML models. Yet, many papers do not report these metrics as shown in Fig. 11. For example, only 30–40% of all studies did not include precision, recall, or F-Score metrics. Although some studies presented a different set of evaluation metrics, this indicates that the majority of studies fail to provide a sufficient level of performance analysis.

Besides typical machine learning evaluation metrics, researchers also used some other metrics to evaluate DL models including Round Trip Time, CPU utilization, memory utilization, request per second, number of accommodated flows, and response time [6, 163]. In [155] authors presented CPU Utilization time, HTTP response time, rate of infected packets, and the average number of installed entries. Moreover, Vishwakarma et al. used computational cost, storage cost, communication cost, consensus delay, and energy consumption when performing the theoretical analysis of a proposed framework [164]. [120] presented Adjusted Mutual Information, Adjusted RAND, Completeness, Homogeneity, and Mutual Information when comparing different clustering-based unsupervised learning methods for attack detection.

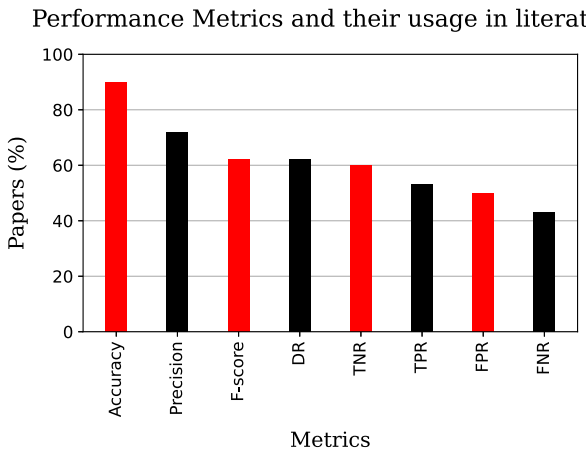

Fig. 11 The statistical evaluation of the different performance metrics used in the literature. Each metric is measured in terms of the percentage of the overall literature presented in evaluations. For example, accuracy is presented in 90% of the papers we presented in this review. DR stands for detection rate

5 Issues, challenges and opportunities for the future

In this section, we list the limitations and challenges that SDNs face despite the advancements in the protocol and application areas (e.g., centralized/decentralized controllers, controller types, hybrid SDNs, stateful/stateless data planes, etc.). Some of the open challenges and issues have been discussed in prior survey [29, 32, 34, 37, 53], thus, the purpose is not to repeat existing information but to provide updated, contextually relevant, and cohesive information in light of the literature findings in recent years. Wherever applicable, we will also provide potential research directions to inspire future work.

One of the major issues that affect the overall performance of DL models is the need for large-scale, high-quality datasets [1, 33, 165]. Section 4 discussed that while some researchers generate custom datasets in simulated environments, it raises concerns about the quality of the data as the characteristics of reproduced anomalies may not be similar to the ones observed in production systems. Moreover, the scale and structure of network topology may not be able to represent the complexity of real-world networks. While there are few open source datasets, more labeled datasets from production systems are necessary to develop robust DL solutions and evaluate the performance of existing DL solutions in a wide range of network configurations and attack scenarios.

Another challenge in the adoption of DL methods is the computational needs of the training phase as DL is notorious for its long and resource-intensive training times. While the cost of one time training may not be significant, especially when using high-performance clusters, it becomes prohibitively expensive as the complexity of models as well as training frequency increases. Researchers proposed online training to initiate a model with a relatively small dataset and update it with new data as it becomes available to avoid the cost of retraining from scratch every time models need to be updated [166]. Moreover, deep transfer learning can also be used to mitigate the need for complete model training for each network [167-169]. Once challenges to adopting DL solutions are handled, they can be widely adopted for more attack types. Section 3 showed that there is a wide gap between the existing vulnerabilities and the utilization of DL-based approaches for developing mitigation mechanisms. Except for DDoS attacks, most other attack types have not received any attention from researchers for the

development of DL-based mitigation mechanisms. Hence, the application of DL for other types of attacks in any level of SDN architecture is worth investigating to take advantage of it or at least evaluate its performance.

Finally, Virtualized Network Functions (VNFs) and associated network services offer many possibilities for enhancing security in SDN environments [34]. Using VNFs can reduce the need for middle-box-based security approaches for SDNs, as VNFs can be directly implemented at the different layers to protect against various security vulnerabilities. Although SDN and NFV evolve as essential technologies for developing next-generation telecommunications, the sufficient effort has not been put towards the development of standard binding interfaces between SDN and NFV paradigms [34]. One of the potential solutions for enhancing the optimization of SDN resource utilization and management is also through the synergy between NFV and SDN [32]. Existing ML-based methods have been leveraged to improve network optimization and service utilization costs (e.g., Reinforcement Learning, Markov Decision Processes, and Bayesian learning) [32, 170]. Therefore, there is considerable potential regarding utilizing NFV in conjunction with DL methods to enhance the security of SDNs in particular and provide other benefits (e.g., network optimization, service delivery costs, optimal resource utilization) in the future.

Funding The work in this study was supported in part by the NSF grants 2019164, 2145742, and 2007789.

Data availability Enquiries about data availability should be directed to the authors

Declarations

Competing Interests The authors have not disclosed any competing interests

References

1. Phan, T.V., Nguyen, T.G., Dao, N.-N., Huong, T.T., Thanh, N.H., Bauschert, T.: Deepguard: efficient anomaly detection in sdn with fine-grained traffic monitoring. IEEE Trans. Netw. Serv. Manage. **17**(3), 1349–1363 (2020)
2. Jain, S., Kumar, A., Mandal, S., Ong, J., Poutievski, L., Singh, A., Venkata, S., Wanderer, J., Zhou, J., Zhu, M., et al.: B4: Experience with a globally-deployed software defined wan. ACM SIGCOMM Comput. Commun. Rev. **43**(4), 3–14 (2013)
3. Wang, T., Chen, H.: Sguard: a lightweight sdn safe-guard architecture for dos attacks. China Commun. **14**(6), 113–125 (2017)
4. Shin, S., Yegneswaran, Y., Porras, P., Gu, G.: Avant-guard: scalable and vigilant switch flow management in software-defined networks. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, vol. Berlin, Germany, pp. 1–10 (2013)
5. Dotcenko, S., Vladyko, A., Letenko, I.: A fuzzy logic-based information security management for software-defined networks. Paper presented at: 2014 16th International Conference on Advanced Communication Technology (ICACT), vol. Pyeongchang, South Korea, pp. 1-8 (2014)
6. Gao, S., Peng, Z., Xiao, B., Hu, A., Song, Y., Ren, K.: Detection and mitigation of dos attacks in software defined networks. IEEE Trans. Net. **28**(3), 1419–1433 (2020)
7. Tang, D., Yan, Y., Zhang, S., Chen, J., Qin, Z.: Performance and features: Mitigating the low-rate tcp-targeted dos attack via sdn. IEEE J. Selected Areas of Commun. **40**(1), 428–435 (2022)
8. Wang, H., Xu, L., Gu, G.: Floodguard: a dos attack prevention extension in software-defined networks. *In: 2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pp. 239-250 (2015)
9. Zheng, J., Li, Q., Gu, G., Cao, J., Yau, D. K. Y., Wu, J.: Realtime ddos defense using cots sdn switches via adaptive correlation analysis, IEEE Transactions on Information Forensics and Security, pp. 1838-1834 (2018)
10. Alshra'a, A., Seitz, J.: Using inspector device to stop packet injection attack in sdn. IEEE Commun. Lett. **23**(7), 1174–1177 (2019)
11. 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 (2018)
12. Hu, B.:, et al.: A deep one-class intrusion detection scheme in software defined industrial networks. IEEE Trans. Industrial Inform. **18**(6), 4286–4297 (2022)
13. Janabi, A.H., Kanakis, T., Johnson, M.: Convolutional neural network based algorithm for early warning proactive system security in software defined networks. IEEE Access **10**, 14–301 (2022)
14. Yang, L., Song, Y., Gao, S., Hu, A., Xiao, B.: Griffin: Real-time network intrusion detection system via ensemble of autoencoder in sdn. IEEE Trans. Network and Service Manag. **19**, 1–13 (2022)
15. Muthanna, M.S.A., Alkanhel, R., Muthanna, A., Rafiq, A., Abdullah, W.A.M.: Towards sdn-enabled, intelligent intrusion detection system for internet of things (iot). IEEE Access. **22**, 756–769 (2022)
16. Zhou, Y.-F., Jiang, R.-H., Wu, X., He, J.-Y., Weng, S., Peng, Q.: Branchgan: unsupervised mutual image-to-image transfer with a single encoder and dual decoders. IEEE Trans. Multimedia. **21**, 3136–3150 (2019)
17. Ren, S., an Ross Girshick, K. H., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Adv. Neural Inf. Process. Syst. **28** (2017)
18. Zhou, Z., Rahman, S.M.M., Tajbakhsh, N., Liang, J.: Unet++: A nested u-net architecture for medical image segmentation. Lect. Notes Comput. Sci. **11045**, 3–11 (2018)
19. Roy, S., Menapace, W., Oei, S., Luijten, B., Fini, E., Saltori, C., Huijben, I., Chennakeshava, N., Mento, F., Sentelli, A., Peschiera, E., Trevisan, R., Maschietto, G., Torri, E., Inchingolo, R., Smargiassi, A., Soldati, G., Rota, P., Passerini, A., van Sloun, R.J.G., Ricci, E., Demi, L.: Deep learning for classification and localization of covid-19 markers in point-of-care lung ultrasound. IEEE Trans. Med. Imaging **13**, 2676–2688 (2020)
20. Oksuz, I., Clough, J.R., Ruijsink, B., Anton, E.P., Bustin, A., Cruz, G., Prieto, C., King, A.P., Schnabel, J.A.: Deep learning-based detection and correction of cardiac mr motion artefacts during reconstruction for high-quality segmentation''. IEEE Trans. Med.l Imaging **13**, 4001–4011 (2020)

21. Yu, J., Chen, H., Dou, Q., Qin, J., Heng, P.-A.: Automated melanoma recognition in dermoscopy images via very deep residual networks. IEEE Trans. Med. Imaging **12**, 994–1015 (2018)
22. Ahmed, H., La, H.M., Tran, K.: Rebar detection and localization for bridge deck inspection and evaluation using deep residual network. Automat. Constr. **120**, 1–38 (2020)
23. Ahmed, H., Gucunski, N., La, H. M.: Rebar detection using ground penetrating radar with state-of-the-art convolutional neural networks," *The 9th International Conference on Structural Health Monitoring of Intelligent infrastructure*, pp. 1-6 (2019). [Online]. Available: https://ara.cse.unr.edu/wp-content/uploads/2014/12/SHMII-GPR-Paper-Final-Version-4.pdf [Accessed on 20 June 2022]
24. Ahmed, H., La, H. M., Pekcan, G.: Rebar detection and localization for non-destructive infrastructure evaluation using deep residual networks. Proceedings of the 14th International Symposium on Visual Computing. pp. 1-6 (2019)
25. Ahmed, H., Tavakolli, A., La, H. M.: Use of deep encoder-decoder network for sub-surface inspection and evaluation of bridge decks. Proceedings of the 13th International Workshop on Structural Health Monitoring 2022. p. (Accepted for Publication), (2022)
26. Ahmed, H., Nguyen, S. T., La, D., Le, C. P., La, H. M.: Multi-directional bicycle robot for bridge inspection with steel defect detection system. IEEE International Conference on Robotics and Automation (ICRA) 2022, p. (Accepted for Publication), (2022)
27. Chen, S., Lin, H., Yao, M.: Improving the efficiency of encoder-decoder architecture for pixel-level crack detection. IEEE Access. **186**, 657–671 (2019)
28. Ahmed, H., La, H.M., Gucunski, N.: Review of non-destructive civil infrastructure evaluation for bridges: State-of-the-art robotic platforms, sensors and algorithms. Sensors **14**, 1–38 (2020)
29. Ahmed, I., Din, S., Jeon, G., Piccialli, F., Fortino, G.: Towards collaborative robotics in top view surveillance: A framework for multiple object tracking by detection using deep learning. IEEE/CAA J. Automatica Sinica. **8**, 1253–1270 (2021)
30. Church, A., Lloyd, J., Hadsell, R., Lepora, N.F.: Deep reinforcement learning for tactile robotics: Learning to type on a braille keyboard. IEEE Robotics and Automation Letters. **5**, 6145–6152 (2020)
31. Nguyen, T.T., Nguyen, N.D., Nahavandi, S.: Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications. IEEE Trans. Cybernet. **50**, 3826–3839 (2020)
32. X. J. et al.: A survey of machine learning techniques applied to software defined networking (sdn): Research issues and challenges. IEEE Commun. Surveys and Tutorials **21**, 1393–430 (2019)
33. Ahmad, I., Shahabuddin, S., Malik, H., Harjula, E., Leppänen, T., Loven, L., Anttonen, A., Sodhro, A.H., Alam, M.M., Juntti, M., et al.: Machine learning meets communication networks: current trends and future challenges. IEEE Access **8**, 223–418 (2020)
34. Chica, J.C.C., Imbachi, J.C., Vega, J.F.B.: Security in sdn: A comprehensive survey. J. Net. Comput. Appl. **8**, 1–23 (2020)
35. Jimenez, M.B., Fernandez, D., Rivaneira, J.E., Bellido, L., Cardenas, A.: A survey of the main security issues and solutions for the sdn architecture. IEEE Access. **122**, 016–039 (2021)
36. Maleh, Y., Qasmaoui, Y., El Gholami, K., Sadqi, Y., Mounir, S.: A comprehensive survey on sdn security: threats, mitigations, and future directions. J. Reliable Intell. Environ. **1**, 39 (2022)
37. Rahouti, M., Xiong, K., Xin, Y., Jagatheesaperumal, S.K., Ayyash, M., Shaheed, M.: Sdn security review: threat taxonomy, implications, and open challenges. IEEE Access **45**, 820–855 (2022)
38. Deb, R., Roy, S.: A comprehensive survey of vulnerability and information security in sdn. Comput. Net. **5**, 1–30 (2022)
39. Singh, M.P., Bhandari, A.: New-flow-based ddos attacks in sdn: Taxonomy, rationales and research challenges. Comp. Commun. **154**, 509–527 (2020)
40. Amin, R., Rojas, E., Aqdus, A., Ramzan, S., Casillas-Perez, D., Arco, J.M.: A survey on machine learning techniques for routing optimization in sdn. IEEE Access **104**, 582–612 (2019)
41. Amin, R., Reisslein, M., Shah, N.: Hybrid sdn networks: a survey of existing approaches''. IEEE Commun. Surveys and Tutorials **20**, 3259–3307 (2018)
42. Kellerer, W., Kalmbach, P., Blenk, A., Basta, A., Reisslein, M., Schmid, S.: Adaptable and data-driven softwarized networks: Review, opportunities, and challenges. Proceedings of the IEEE **107**, 1–35 (2019)
43. Bannour, F., Souihi, S., Mellouk, A.: Distributed sdn control: survey, taxonomy, and challenges. IEEE Commun Surveys and Tutorials **20**, 333–355 (2018)
44. Huang, X., Cheng, S., Cao, K., Cong, P., Wei, T., Hu, S.: A survey of deployment solutions and optimization strategies for hybrid sdn networks. IEEE Commun. Surveys and Tutorials **21**, 1483–1507 (2019)
45. Khorsandroo, S., Sanchez, A.G., Tosun, A.S., Arco, J., Doriguzzi-Corin, R.: Hybrid sdn evolution: A comprehensive survey of the state-of-the-art. Comput. Net. **192**, 107981 (2021)
46. Al-Heety, O., Zakaria, Z., Ismail, M., Shakir, M.M., Alani, S., Alsariera, H.: A comprehensive survey: benefits, services, recent works, challenges, security, and use cases for sdn-vanet. IEEE Access **91**, 028–048 (2020)
47. Alam, I., Sharif, K., Li, F., Latif, Z., Karim, M.M., Biswas, S., Nour, B., Wang, Y.: A survey of network virtualization techniques for internet of things using sdn and nfv. ACM Comput. Survey **53**, 1–40 (2020)
48. Farris, I., Taleb, T., Khettab, Y., Song, J.: A survey on emerging sdn and nfv security mechanisms for iot systems. IEEE Commun. Surveys and Tutorials **21**, 812–838 (2019)
49. Ali, A., Yousaf, M.M.: Novel three-tier intrusion detection and prevention system in software-defined networks. IEEE Access **8**, 109–677 (2020)
50. Wang, J., Liu, J., Guo, H., Mao, B.: Deep reinforcement learning for securing software-defined industrial networks with distributed control plane. IEEE Trans. Industr. Inf. **18**(6), 4275–4285 (2021)
51. Ali, S.T., Sivaraman, V., Radford, A., Jha, S.: A suvey of securing network using software defined networking. IEEE Trans. Reliab. **64**, 1086–1098 (2015)
52. Nunes, B.A.A., Mendonca, M., Nguyen, X.-N., Obraczka, K., Turletti, T.: A survey of software-defined networking: Past, present, and future of programmable networks. IEEE Commun. Surveys and Tutorials **16**, 1617–1635 (2014)
53. Scott-Hayward, S., Natarajan, S., Sezer, S.: A survey of security in software defined networks. IEEE Commun. Surveys and Tutorials **18**, 623–655 (2016)
54. Ahmad, I., Namal, S., Ylianttila, M., Gurtov, A.: Security in software defined networks: a survey. IEEE Commun. Surveys and Tutorial **17**, 2317–2347 (2015)
55. Benzekki, K., El Fergougui, A., Elalaoui, A.E.: Software-defined networking (sdn): a survey. Security and Commun. Net. **9**, 5803–5833 (2017)
56. Li, W., Meng, W., Kwok, L.F.: A survey on openflow-based software-defined networks: security challenges and countermeasures. J. Net. Comput. Appl. **68**, 126–139 (2016)
57. 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. Surveys Tutorials. **82**, 602–623 (2016)
58. Dargahi, T., Alberto Caponi, M.A., Bianchi, G., Conti, M.: A survey on the security of stateful sdn data planes. IEEE Commun. Surveys and Tutorials **19**, 1701–1726 (2017)
59. Dong, S., Abbas, K., Jain, R.: A survey on distributed denial of service (ddos) attacks in sdn and cloud computing environments. IEEE Access **80**, 813–828 (2019)
60. Sultana, N., Chilamkurti, N., Peng, W., Alhadad, R.: Survey on sdn based network intrusion detection system using machine learning approaches. Peer-to-Peer Network. Appl. **12**, 493–501 (2019)
61. Ahmed, M., Shatabda, S., Islam, A., Robin, M., Islam, T.: *et al.*, Intrusion detection system in software-defined networks using machine learning and deep learning techniques–a comprehensive survey. (2021)
62. Jafarian, T., Masdari, M., Ghaffari, A., Majidzadeh, K.: A survey and classification of the security anomaly detection mechanisms in software defined networks. Cluster Comput. **24**, 1235–1253 (2021)
63. Zhao, Y., Li, Y., Zhang, X., Geng, G., Zhang, W., Sun, Y.: A survey of networking applications applying the software defined networking concept based on machine learning. IEEE Access **95**, 397–418 (2019)
64. Han, T., Jan, S.R.U., Tan, Z., Usman, M., Jan, M.A., Khan, R., Xu, Y.: A comprehensive survey of security threats and their mitigation techniques for next-generation sdn controllers. Concurrency Computat. Pract. Exper. **32**, 1–21 (2020)
65. Schmidhuber, J.: Deep learning in neural networks: an overview. Neural Netw. **61**, 85–117 (2015)
66. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
67. Sarker, I.H.: Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. SN Comput. Sci. **2**, 420 (2021)
68. Aldweesh, A., Derhab, A., Emam, A.Z.: Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. Knowl. Based Syst. **189**, 105–124 (2020)
69. O'Shea, K., Nash, R.: An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458, (2015)
70. Glorot, X., Bengio, Y.:Understanding the difficulty of training deep feedforward neural networks. Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings, pp. 249-256, (2010)
71. Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M.P., Shyu, M.-L., Chen, S.-C., Iyengar, S.S.: A survey on deep learning: algorithms, techniques, and applications. ACM Comput. Surveys (CSUR) **51**(5), 1–36 (2018)
72. Salehinejad, H., Sankar, S., Barfett, J., Colak, E., Valaee, S.: Recent advances in recurrent neural networks. arXiv preprint arXiv:1801.01078, (2017)
73. Naskath, J., Sivakamasundari, G., Begum, A.: A study on different deep learning algorithms used in deep neural nets: Mlp som and dbn. Wireless Personal Commun. **14**, 1–24 (2022)
74. Tan1, C., Sun2, F., Kong1, T., Zhang1, W., Yang1, C., Liu, C.: A survey on deep transfer learning. International Conference on Artificial Neural Networks, p. 270-279, (2018)
75. Liu, X., Yu, W., Liang, F., Griffith, D., Golmie, N.: On deep reinforcement learning security for industrial internet of things. Comput Commun. **168**, 20–32 (2021)
76. Wang, Y., Hu, T., Tang, G., Xi, J., Lu, J.: Sgs: safe-guard scheme for protecting control plane against ddos attacks in software-defined networking. IEEE Access **7**, 34–699 (2019)
77. Min, J., Yuejie, S., Qing, G., Zihe, G., Suofe, X.: Ddos attack detection method for space-based network based on sdn architecture. ZTE Commun. **18**(4), 18–25 (2020)
78. Alanazi, F., Jambi, K., Eassa, F., Khemakhem, M., Basuhail, A., Alsubhi, K.: Ensemble deep learning models for mitigating ddos attack in software-defined network. Intell. Automat. Soft Comput. **33**(2), 923–938 (2022)
79. H., S. et al.: A deep cnn ensemble framework for efficient ddos attack detection in software defined networks. IEEE Access **8**(53), 972–983 (2021)
80. Lent, D.M.B., Novaes, M.P., Carvalho, L.F., Lloret, J., Rodriguez, J.J.P.C., Proenca, M.L.: A gated recurrent unit deep learning model to detect and mitigate distributed denial of service and portscan attacks. IEEE Access **10**, 73–229 (2022)
81. Ujjan, R.M.A., Pervez, Z., Dahal, K., Bashir, A.K., Mumtaz, R., González, J.: Towards sflow and adaptive polling sampling for deep learning based ddos detection in sdn. Futur. Gener. Comput. Syst. **111**, 763–779 (2020)
82. Yeom, S., Choi, C., Kim, K.: Lstm-based collaborative source-side ddos attack detection. IEEE Access **7**, 44–046 (2022)
83. Gadze, J.D., Bamfo-Asante, A.A., Agyemang, J.O., Nunoo-Mensah, H., Opare, K.A.-B.: An investigation into the application of deep learning in the detection and mitigation of ddos attack on sdn controllers. Technologies **14**, 25 (2021)
84. Shu, J., Zhou, L., Zhang, W., Du, X., Guizani, M.: Collaborative intrusion detection for vanets: a deep learning-based distributed sdn approach. IEEE Trans. Intell. Transport. Syst. **22**, 4519–4523 (2021)
85. Ravi, N., Shalinie, S.M.: Learning-driven detection and mitigation of ddos attack in iot via sdn-cloud architecture. IEEE Int. Things J. **7**, 3559–3571 (2020)
86. Rezapour, A., Tzeng, W.-G.: Rl-shield: mitigating target link-flooding attacks using sdn and deep reinforcement learning routing algorithm. IEEE Trans. Depend. Secure Comput. **19**, 1–17 (2022)
87. ur Rasool, R., Ashraf, U., Ahmed, K., Wang, H., Rafique, W., Anwar, Z.: Cyberpulse: a machine learning based link flooding attack mitigation system for software defined networks. IEEE Access **34**, 885–900 (2019)
88. Ahuja, N., Singal, G., Mukhopadhyay, D.: Dlsdn: Deep learning for ddos attack detection in software defined networking. 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), (2021)
89. Wang, J., Liu, J.: Deep learning for securing software-defined industrial internet of things: attacks and countermeasures. IEEE Int. Things J. **9**, 1–11 (2022)
90. Soltani, S., Shojafar, M., Mostafaeit, H., Pooranian, Z., Tafazolli, R.: Link latency attack in software-defined networks. 17th International Conference on Network and Service Management (CNSM), (2021)
91. Wang, J., Tan, Y., Liu, J., Zhang, Y.: Topology poisoning attack in sdn-enabled vehicular edge network. IEEE Int. Things J. **7**(10), 9563–9575 (2020)
92. Mohammadi, R., Javidan, R., Conti, M.: Slicots: an sdn-based lightweight countermeasure for tcp syn flooding attacks. IEEE Trans. Net. Service Manag. **14**, 487–498 (2017)
93. Chen, M.-H., Ciou, J.-Y., Chung, I.-H., Chou, C.-F.: Flexprotect: a sdn-based ddos attack protection architecture for multi-tenant data centers.In: Proceedings of International Conference on High Performance Computing Asia-Pacific Region., pp. 1-6, (2018)
94. Boite, J., Nardin, P.-A., Rebecchi, F., Bouet, M., Conan, V.: Statesec: stateful monitoring for ddos protection in software defined networks. Paper presented at: 2017 IEEE Conference on Network Softwarization (NetSoft), vol. Bologna, Italy, pp. 1-6, (2017)

95. Varghese, J.E., Muniyal, B.: An efficient ids framework for ddos attacks in sdn environment. IEEE Access **69**, 680–700 (2021)
96. Xu, Y., Sun, H., aand Shijin Sun, F. X.: Efficient ddos detection based on k-fknn in software defined networks. IEEE Access **7**, 160–547 (2019)
97. Novaes, M.P., Carvalho, L.F., Lloret, J., Proença, M.L.: Long short-term memory and fuzzy logic for anomaly detection and mitigation in software-defined network environment. IEEE Access **8**, 83–765 (2020)
98. Hussain, J., Hnamte, V.: Novel three-tier intrusion detection and prevention system in software defined network. IEEE Access **109**, 662–677 (2020)
99. Gkounis, D., Kotronis, V., Liaskos, C., Dimitropoulos, X.: On the interplay of link-flooding attacks and traffic engineering. SIGCOMM Comput. Commun. Rev. **46**(2), 5–11 (2016)
100. Ahuja, N., Singal, G., Mukhopadhyay, D.: Ddos attack sdn dataset," https://data.mendeley.com/datasets/jxpfjc64kr/1, 2020
101. Xiang, S., Zhu, H., Xiao, L., Xie, W.: Modeling and verifying topoguard in openflow-based software defined networks. In: Proceedings of 2018 International Symposium on Theoretical Aspects of Software Engineering (TASE). pp. 84-91, (2018)
102. Skowyra, R., Xu, L., Gu, G., Dedhia, V., Hobson, T., Okhravi, H., Landry, J.: 2018 Effective topology tampering attacks and defenses in software-defined networks. In: Proceeding of 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, pp. 374-386,
103. Deng, S., Gao, X., Lu, Z., Gao, X.: Packet injection attack and its defense in software-defined networks. IEEE Trans. Inf. Forensics Secur. **13**(3), 695–705 (2018)
104. Phan, T.V., Bauschert, T.: Deepair: deep reinforcement learning for intrusion response in software-defined networks. IEEE Trans. Net. Service Manag. **19**, 1–12 (2022)
105. Razib, M.A., Javeed, D., Khan, M.T., Alkanhel, R., Muthanna, M.S.A.: Cyber threats detection in smart environments using sdn-enabled dnn-lstm hybrid framework. IEEE Access **10**, 1–12 (2022)
106. Tu, Z., Zhou, H., Li, K., Li, M., Tian, A.: An energy-efficient topology design and ddos attacks mitigation for green software-defined satellite network. IEEE Access **211**, 434–451 (2020)
107. Javeed, D., Gao, T., Khan, M.T., Ahmad, I.: A hybrid deep learning-driven sdn enabled mechanism for secure communication in internet of things (iot). Sensors **21**(14), 48–84 (2021)
108. Garg, S., Kaur, K., Kumar, N., Rodrigues, J.J.: Hybrid deep-learning-based anomaly detection scheme for suspicious flow detection in sdn: a social multimedia perspective. IEEE Trans. Multimedia **21**(3), 566–578 (2019)
109. Hu, D., Hong, P., Chen, Y.: 2017 Fadm: Ddos flooding attack detection and mitigation system in software-defined networking. GLOBECOM 2017-2017 IEEE Global Communications Conference. IEEE, pp. 1-7, (2017)
110. Li, C., Wu, Y., Yuan, X., Sun, Z., Wang, W., Li, X., Gong, L.: Detection and defense of ddos attack-based on deep learning in openflow-based sdn. Int. J. Commun. Syst. **31**(5), 1–20 (2018)
111. Shafi, Q., Basit, A., Qaisar, S., Koay, A., Welch, I.: Fog-assisted sdn controlled framework for enduring anomaly detection in an iot network. IEEE Access **73**, 713–724 (2018)
112. Yue, M., Wang, H., Liu, L., Wu, Z.: Detecting dos attacks based on multi-features in sdn. IEEE Access **8**, 104–688 (2020)
113. Ali, A., Yousaf, M. M.: Deep learning based intrusion detection system : software defined network. Asian Conference on Innovation in Technology (ASIANCON), (2021)
114. Elsayed, M.S., Le-Khac, N.-A., Dev, S., Jurcut, A.D., Ddosnet: A deep-learning model for detecting network attacks, in,: IEEE 21st International Symposium on A World of Wireless, Mobile and Multimedia Networks"(WoWMoM). IEEE **2020**, 391-396 (2020)
115. ElSayed, M.S., Le-Khac, N.-A., Azer, M.A., Jurcut, A.D.: A flow based anomaly detection approach with feature selection method against ddos attacks in sdns. IEEE Trans. Cognitive Commun. **8**, 1–20 (2022)
116. Scaranti, G.F., Carvalho, L.F., Proenca, M.L.: Artificial immune systems and fuzzy logic to detect flooding attacks in software-defined networks. IEEE Access **100**, 172–185 (2020)
117. Ahuja, N., Singal, G., Mukhopadhyay, D., Kumar, N.: Automated ddos attack detection in software defined networking. J. Netw. Comput. Appl. **187**, 1–20 (2021)
118. Novaes, M.P., Carvalho, L.F., Lloret, J., Jr., M. L. P.: Adversarial deep learning approach detection and defense against ddos attacks in sdn environments. Fut. Gene. Comput. Syst. **125**, 1–20 (2021)
119. Peng, H., Sun, Z., Zhao, X., Tan, S., Sun, Z.: A detection method for anomaly flow in software defined network. IEEE Access **27**, 809–818 (2018)
120. He, D., Chan, S., Ni, X., Guizani, M.: Software-defined-networking-enabled traffic anomaly detection and mitigation. IEEE Int. Things J. **4**, 1890–1899 (2017)
121. Li, Q., Liu, Y., Liu, Z., Pang, C.: Efficient forwarding anomaly detection in software-defined networks. IEEE Transacctions on Parallel and Distributed Systems. **32**, 2676–1697 (2021)
122. Dhawan, M., Poddar, R., Mahajan, K., Mann, V.: Sphinx: detecting security attacks in software-defined networks. Ndss **15**, 8–11 (2015)
123. Musumeci, F., Fidanci, A.C., Paolucci, F., Cugini, F., Tornatore, M.: Machine-learning-enabled ddos attacks detection in p4 programmable networks. J. Net. Syst. Manag. vol. **30**(21), 1–27 (2022)
124. Zhang, X., Cui, L., Tso, F.P., Jia, W.: pheavy: predicting heavy flows in the programmable data plane. IEEE Trans. Netw. Serv. Manage. **18**(4), 4353–4365 (2021)
125. da Silveira Ilha, A., Cardoso Lapolli, Â., Marques, J.A., Gaspary, L.P.: Euclid: a fully in-network, p4-based approach for real-time ddos attack detection and mitigation. IEEE Trans. Net. Serv. Manag. **18**(3), 3121–3140 (2021)
126. The caida ucsd anonymized internet traces 2016. [Online]. Available: https://www.caida.org/data/passive/passive_2016_dataset.xml
127. The caida ucsd ddos attack 2007 dataset. [Online]. Available: ttp://www.caida.org/data/passive/ddos-20070804_dataset.xml
128. Shin, S., Gu, G.: Attacking software-defined networks: A first feasibility study. In: Proc. Second ACM SIGCOMM Work. Hot Top. Softw. Defin. Netw., pp. 165-166, (2013)
129. Klöti, R., Kotronis, V., Smith, P.: Openflow: a security analysis. In Proceedings of International Conference on Network Protocols (ICNP), pp. 1-6, (2013)
130. Zhang, M., Hou, J., Zhang, Z., Shi, W., Qin, B., Liang, B., Fine-grained fingerprinting threats to software-defined networks, in,: IEEE Trustcom/BigDataSE/ICESS. IEEE **2017**, 128–135 (2017)
131. Sonchack, J., Aviv, A. J., Keller, E.: Timing sdn control planes to infer network configurations In Proceedings of the 2016 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization, pp. 19–22 , (2016)
132. Gao, B.X.S., Li, Z., Wei, G.: Security threats in the data plane of software-defined networks. IEEE Netw. **32**(4), 108–113 (2018)
133. Farhin, F., Sultana, I., Islam, N., Kaiser, M.S., Rahman, M.S., Mahmud, M.: Attack detection in internet of things using software defined network and fuzzy neural network. IEEE Trans. Industr. Inf. **18**(1), 467–476 (2021)
134. Krishnan, P., Duttagupta, S., Achuthan, K.: Varman: multi-plane security framework for software defined networks. Comput. Commun. **148**, 215–239 (2019)
135. Ahuja, N., Singal, G., Mukhopadhyay, D., Nehra, A.: Ascertain the efficient machine learning approach to detect different arp attacks. Comput. Elect. Eng. **99**, 107757 (2022)

136. Lee, C., Yoon, C., Shin, S., Cha, S.: Indago: a new framework for detecting malicious sdn applications. In: Proceedings of 2018 IEEE 26th International Conference on Network Protocols (ICNP), pp. 220-230, (2018)
137. Cao, J., Li, Q., Xie, R., Sun, K., Gu, G., Xu, M., Yang, Y.: The crosspath attack: disrupting the sdn control channel via shared links. In: Proceedings of 28th USENIX Security Symposium, pp. 1-18, (2019)
138. Khamaiseh, S., Serra, E., Li, Z., Xu, D.: Detecting saturation attacks in sdn via machine learning. 4th International Conference on Computing, Communications and Security (ICCCS), (2019)
139. Divekar, M. P., Savla, V., Mishra, R., Shirole, M.: Benchmarking datasets for anomaly-based network intrusion detection: Kdd cup 99 alternatives. Proc. IEEE 3rd Int. Conf. Comput., Commun. Secur. (ICCCS), pp. 1-8, (2018)
140. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A. A.: A detailed analysis of the kdd cup 99 data set. In Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl., pp. 1-6, (2009)
141. Shiravi, H., Shiravi, M.T., Ghorbani, A.A.: Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Comput. Security **31**, 357–374 (2012)
142. Moustaf, N., Slay, J.: The evaluation of network anomaly detection systems: statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set. Inform. Security J. **25**, 18–31 (2016)
143. Kolias, C., Kambourakis, G., Stavrou, A., Gritzalis, S.: Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset. IEEE Commun. Surveys Tuts. **18**, 184–208 (2016)
144. Sharafaldin, A., Lashkari, H., Ghorbani, A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. Proc. ICISSP. **1**, 108–116 (2018)
145. of Cybersecurity, C. I.: Cse-cic-ids2018. Accessed July 10, 2022, [Online]
146. Ring, M., Wunderlich, S., Grüdl, D., Landes, D., Hotho, A.,: "Flow-based benchmark data sets for intrusion detection. In: Eur. Conf. Inf. Warf. Secur. ECCWS, pp. 361-369, 2017
147. Sharafaldin, A. H., Lashkari, S. H., Ghorbani, A. A.: Developing realistic distributed denial of service (ddos) attack dataset and taxonomy. In Proc. Int. Carnahan Conf. Secur. Technol. (ICCST), pp. 1–8, (2019)
148. Song, H. T., Okabe, Y.: Description of kyoto university benchmark data, (2006)
149. ElSayed, M. S., Le-Khac, N.-A., Jorcot, A. D. : Insdn: a novel sdn intrusion dataset. IEEE Access, pp. 165-623, (2020)
150. Garg, S., Singh, A., Aujla, G.S., Kaur, S., Batra, S., Kumar, N.: Probabilistic data structures-based anomaly detection scheme for software-defined internet of vehicles. IEEE Trans. Intell. Transport. Syst. **22**, 3557–3567 (2021)
151. Wang, B., Sun, Y., Xu, X.: A scalable and energy-efficient anomaly detection scheme in wireless sdn-based mmtc networks for iot. IEEE Int. Things J. **8**, 1388–1406 (2021)
152. Yin, D., Zhang, L., Yang, K.: A ddos attack detection and mitigation with software-defined internet of things framework. IEEE Access **24**, 606–624 (2018)
153. Assis, M.V.O.D., Hamamoto, A.H., Abrao, T., Proenca, M.L.: A game theoretical based system using holt-winters and genetic algorithm with fuzzy logic for dos/ddos mitigation on sdn networks. IEEE Access **5**, 9485–9497 (2017)
154. Ravi, N., Shalinie, S.M., Theres, D.D.J.: Balance: Link flooding attack detection and mitigation via hybrid-sdn. IEEE Trans. Netw. Serv. Manage. **17**(3), 1715–1730 (2020)
155. Kumar, P., Tripathi, M., Nehra, A., Conti, M., Lal, C.: Safety: early detection and mitigation of tcp syn flood utilizing entropy in sdn. IEEE Trans. Net. Service Manag. **15**, 1545–1560 (2018)
156. Aliyu, I., Feliciano, M.C., Engelenburg, S.V., Kim, D.O., Lim, C.G.: A blockchain-based federated forest for sdn-enabled in-vehicle network intrusion detection system. IEEE Access **102**, 593–619 (2021)
157. Li, J., Zhao, Z., Li, R., Zhang, H.: Ai-based two-stage intrusion detection for software defined iot networks. IEEE Int. Things J. **6**, 2093–2103 (2019)
158. Segura, G.A.N., Chorti, A., Margi, C.B.: Centralized and distributed intrusion detection for resource-constrained wireless sdn networks. IEEE Int. Things J. **9**, 7746–7759 (2022)
159. Janabi, A.H., Kanakis, T., Johnson, M.: Overhead reduction technique for software-defined network based intrusion detection systems. IEEE Access **66**, 481–492 (2022)
160. Bagaa, M., Taleb, T., Bernabe, J.B., Skarmeta, A.: A machine learning security framework for iot systems. IEEE Access **114**, 066–078 (2020)
161. Raja, G., Anbalagan, S., Vijayaraghavan, G., Dhanasekaran, P., Al-Otaibi, Y.D., Bashir, A.K.: Energy-efficient end-to-end security for software-defined vehicular networks. IEEE Trans. Industrial Informatics **17**, 5730–5738 (2021)
162. Assis, M. V. O. D., Novaes, M. P., . Zerbini, C. B, Carvalho, L. F., Abrao, T., Jr, M. L. P.: "Fast defense system against attacks in software defined networks," *IEEE Access*, pp. pp. 69 620–69 640, 2018
163. Zhou, Y., Cheng, G., Yu, S.: "An sdn-enabled proactive defense framework for ddos mitigation in iot networks," *IEEE Transactions on Information Forensics and Security*, pp. pp. 5366–5381, 2021
164. Vishwakarma, L., Nahar, A., Das, D.: "Lbsv: Lightweight blockchain security protocol for secure storage and communication in sdn-enabled iov," *IEEE Transactions on Vehicular Technology*, pp. pp. 5983–5995, 2022
165. L. F. M. et al.,: "A self-adaptive deep learning-based system for anomaly detection in 5g networks," *IEEE Access, vol. 6*, pp. pp. 7700–7712, 2018
166. Sahoo, D., Pham, Q., Lu, J., Hoi, S. C.: "Online deep learning: Learning deep neural networks on the fly," arXiv preprint arXiv: 1711.03705, 2017
167. Tan, C., Sun, F., Kong, T., Zhang, W., ang, C. Y, Liu, C.: "A survey on deep transfer learning," in *International conference on artificial neural networks*. Springer, 2018, pp. 270–279
168. Alonso, R. S., Sittón-Candanedo, I., Casado-Vara, R., Prieto, J., Corchado, J. M.: "Deep reinforcement learning for the management of software-defined networks in smart farming," in *2020 International Conference on Omni-layer Intelligent Systems (COINS)*. IEEE, 2020, pp. 1–6
169. Phan, T. V., Sultana, S., Nguyen, T. G., Bauschert, T.: "*q*-transfer: A novel framework for efficient deep transfer learning in networking," in *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*. IEEE, 2020, pp. 146–151
170. R. S. et al.:, "Mdp and machine learning-based cost-optimization of dynamic resource allocation for network function virtualization," *In: Proceedings of IEEE International Conference on Service Computing*, pp. pp. 65–73, 2015

Roya Taheri is a PhD student with expertise in Computer Science at University of Nevada, Reno. She holds a Bachelor's degree in Computer Engineering in 2017 and a Master's degree in Computer Security in 2021 from Amirkabir University of Technology in Iran. Her research focus is on monitoring, data accuracy and anomaly detection for high-performance computing.

Habib Ahmed is an Assistant Professor at the Ghulam Ishaq Khan Institute (GIKI) of Engineering, Sciences and Technology, Pakistan. He completed his PhD from University of Nevada Reno, USA in 2022. Between 2019 and 2022, he was part of the Advanced Robotics and Automation lab, University of Nevada Reno. In 2016, he completed his Master's in Robotics and Intelligent Machines Engineering (RIME) from School of Mechanical and Manufacturing Engineering, National University of Science and Technology, Pakistan. In 2012, he completed his Bachelor's degree in Telecommunication Engineering from National University of Computer and Emerging Sciences, Pakistan. His field of expertise include Computer Vision, Deep Learning, Robotics, Software Defined Networks.

Engin Arslan is an Assistant Professor at the Department of Computer Science and Engineering at the University of Nevada, Reno (UNR). He received PhD from University at Buffalo in 2016 and worked at National Science for Supercomputing Applications (NCSA) as a postdoctoral research associate before joining UNR. His research interests include high-performance computing and networking, edge/-cloud computing, and quantum networking. His work in these areas has been funded by the National Science Foundation, the Department of Energy, Amazon Web Services, and UNR. Most notably, he was the recipient of the prestigious NSF CAREER in 2022. He serves on several committees, including the review board of IEEE TPDS and the UNR Cyberinfrastructure Committee.