Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2022.020724

REVIEW

Machine Learning Techniques for Intrusion Detection Systems in SDN-Recent Advances, Challenges and Future Directions

Gulshan Kumar[1,*] **and Hamed Alqahtani**[2]

[1]Shaheed Bhagat Singh State University, Firozpur, 152024, India

[2]King Khalid University, Abha, 61421, Saudi Arabia

*Corresponding Author: Gulshan Kumar. Email: gulshanahuja@gmail.com

Received: 08 December 2021 Accepted: 16 March 2022

ABSTRACT

Software-Defined Networking (SDN) enables flexibility in developing security tools that can effectively and efficiently analyze and detect malicious network traffic for detecting intrusions. Recently Machine Learning (ML) techniques have attracted lots of attention from researchers and industry for developing intrusion detection systems (IDSs) considering logically centralized control and global view of the network provided by SDN. Many IDSs have developed using advances in machine learning and deep learning. This study presents a comprehensive review of recent work of ML-based IDS in context to SDN. It presents a comprehensive study of the existing review papers in the field. It is followed by introducing intrusion detection, ML techniques and their types. Specifically, we present a systematic study of recent works, discuss ongoing research challenges for effective implementation of ML-based intrusion detection in SDN, and promising future works in this field.

KEYWORDS

Controller; intrusion detection; intrusion detection system; OpenFlow; security; software defined networking; traffic analysis

1 Introduction

The recent use of IT technology and several interconnected smart devices resulted in an abrupt increase in network communication traffic. It has been predicted that there will be financial growth in network traffic in the coming year [1]. To keep up with increased network traffic, several heterogeneous networks have been formed consisting of different communication network protocols and various network equipment in different domains. For example, cellular networks transfer data from different kinds of devices with different standards for communicating data over the network. Therefore, heterogeneous networks becoming more complex in terms of their management of computing resources effectively. Security of the data over a heterogeneous network is considered one of the most important critical issues. Recently several incidents have happened against the security of confidential information of communication networks [2].

To avoid network attacks, several anti-intrusion techniques have been proposed. These anti-intrusion techniques can be divided into six categories; namely, intrusion prevention, intrusion detection, intrusion preemption, intrusion deterrence, intrusion deflection and intrusion countermeasures as presented in Fig. 1 [3,4]. Intrusion detection is considered one of the most effective techniques for handling intrusion into the network. Timely and accurate intrusion detection can help in minimizing the damage and take appropriate countermeasures to block the ongoing attack.

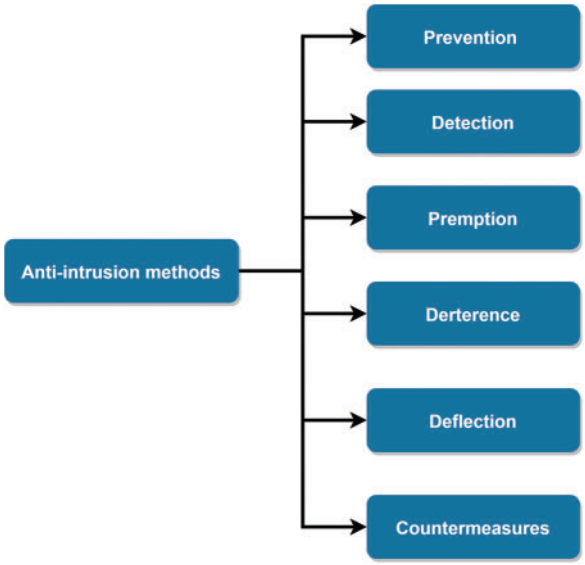

Figure 1: Anti intrusion techniques

Therefore, developing an accurate and intrusion detection system (IDS) is the need of the hour for providing another security layer over the conventional security mechanism like firewalls.

Recently several techniques have been proposed for developing an effective IDS by incorporating more intelligence to handle security issues. Artificial intelligence-based techniques, particularly machine learning (ML) techniques, has been incorporated into IDSs for adding more intelligence into the network data analysis [1]. However, ML techniques have limited access to the data for analysis because of distributed features of traditional networks. Network devices such as switches contain a limited view of data belonging to a small segment of the entire network. Thus, ML models trained on a particular segment of the network is unable to work for detecting the intrusion in the entire network [5,6].

Software Defined Network (SDN) has opened many new possibilities for researchers to address the limited view of the data in traditional network devices [7,8]. In SDN, the control plane and data plane have been decoupled. A centralized controller controls all network resources. A centralized controller enables the dynamic programming of networks by providing a global view of the data at a single point. The global view of the entire network's data helps develop accurate ML models. Therefore, SDNs are more suitable for applications of ML techniques due to the following salient features.

- Recent development in computing devices such as GPUs enables processing a large amount of data in SDN help in training efficient ML model for their application in different fields [9].
- Global view of data in SDN helps to learn entire network behaviour by ML models.

- Global mean of the data at SDN can help deploy feature selection techniques resulting in reducing a considerable amount of data and hence in fast and accurate training of ML models.

Therefore, SDN provides a suitable framework for implementing ML techniques to detect intrusions in the real world [10].

Several intelligent IDSs have been proposed by considering the advantages of SDN architecture and the capability of ML techniques [11]. This paper presents a comprehensive review ML techniques based IDSs specifically for SDN architecture. This review aims to discuss ML-based IDSs architecture for better understanding the current status of intrusion detection in SDNs and project significant clues to conduct future research in this field.

Rest of the paper is structured as follows. A comprehensive review of the existing studies is presented in Section 2. Section 3 explains intrusion detection preliminaries. Section 4 presents ML techniques and their types. Section 5 introduces the SDN and its architecture. Section 6 presents significant ML studies applied for intrusion detection in context of SDN. Section 7 highlights the major research issues in the field. Finally, Section 8 concludes the paper at the end.

2 Related Work

Several types of research have been reported on developing effective and efficient IDS using ML techniques in the recent past. Researches attempted to summarize further development in various review papers. For analyzing the trend in developing IDSs using ML techniques, these reviews can be divided as scenario based reviews, technique based reviews and attack based reviews as presented in Fig. 2.

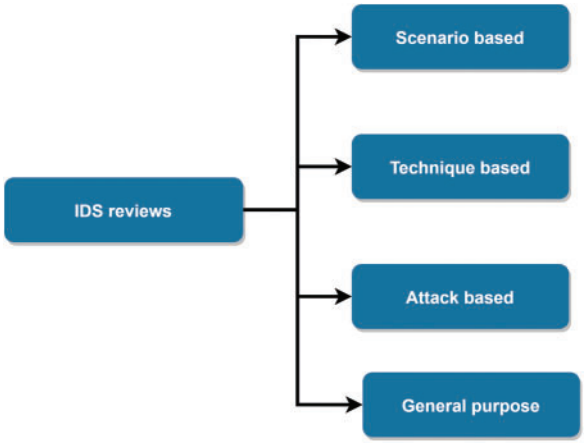

Figure 2: Classification of ML-based IDS reviews

2.1 Scenario-Based Reviews

Scenario-based reviews mainly focus on specific network architecture or scenarios for discussing the trends in ML-based intrusion detection techniques. Several researchers attempted to exploit network configurations' features to explain intrusion detection techniques.

For example, Anantvalee et al. [12] focused on mobile ad-hoc networks (MANETs) for reviewing intrusion detection techniques in this category. The authors conducted a comprehensive study of existing IDSs and provided many clues for future research in this field. Similarly, Nadeem et al.

[13] have also focused on MANETs in their review of intrusion detection techniques. Patel et al. [14] developed intrusion detection and prevention systems for cloud computing environments. They used the features of cloud computing to explain intrusion detection techniques and present different issues in developing intrusion detection and prevention systems for the cloud computing environment. Butun et al. [15] presented their work on IDSs for wireless sensor networks by specifying the pros and cons of intrusion detection techniques in the context of wireless sensor networks.

Bkassiny et al. [16] reviewed existing learning techniques in context to cognitive radio networks. They mainly focused on ML approaches for detecting intrusion accurately. ML approaches have also been reviewed for intrusion detection in context to wireless sensor networks in [17]. Wang et al. [18] focused on artificial intelligence-based techniques for evolving heterogeneous networks. They highlighted significant issues in heterogeneous networks and provided many e points for future research in their review. Klaine et al. [19] provided a comparative analysis of ML techniques applied in self-organizing cellular networks. Whereas ML techniques based network traffic control having focus in [20]. Chen et al. [21] analyzed the solutions proposed for solving issues in wireless sensor networks such as virtual reality, communication and education using neural networks. Xie et al. [1] mainly focused on ML techniques used in SDN. The authors provided details of different ML techniques in context to SDN from different aspects like routing, Resource Management, network traffic analysis and quality of service prediction. They highlighted many issues in developing ML-based systems for SDN. Sultana et al. [22] conducted a comprehensive analysis of ML techniques for detecting the intrusion is in SDN. The authors mainly focused on deep learning techniques for developing network-based IDSs [11]. They also highlighted many challenges for developing deep learning-based IDSs in SDN. Table 1 summarizes the scenario-based IDS reviews mentioned above.

Table 1: Summary of scenario-based IDS reviews

Study	Domain	ML techniques
Anantvalee et al. [12]	MANETs	
Nadeem et al. [13]	MANETs	
Patel et al. [14]	Cloud computing	
Butun et al. [15]	WSNs	
Bkassiny et al. [16]	CRNs	
Wang et al. [18]	Heterogeneous networks	AI-based techniques
Klaine et al. [19]	Cellular networks	Supervised, unsupervised and reinforcement learning
Zhou et al. [23]	Wireless networks	Supervised, unsupervised and reinforcement
Chen et al. [21]	Wireless networks	ANN
Sultana et al. [22]	SDN	Deep learning
Xie et al. [1]	SDN	Supervised, unsupervised and reinforcement learning

2.2 Technique-Based Reviews

Technique-based reviews mainly focus on analyzing the IDS waste on detection techniques. Generally, these papers follow some predefined taxonomy and analyze the existing research papers for each category proposed in the taxonomy. Such reviews are helpful in performing a comparative analysis of different techniques used in IDSs. For example, in 2009, Garcia-Teodoro et al. [24] analyzed the anomaly-based intrusion detection technique by categorizing them into three classes, statistical techniques, ML techniques and knowledge-based techniques. The author provided the pros and cons of each category in detecting intrusions. They have also provided a list of available commercial IDSs. They provided significant research challenges in detecting anomaly-based intrusion detection. Similar and extended work is also reported by Kumar et al. [4]. Here the authors provided a review of artificial intelligence-based IDSs. They explained the general architecture of IDSs and divided IDSs based upon their functional components.

Zhang et al. [25] also focused on anomaly-based detection techniques used in computer networks. They proposed to divide anomaly-based techniques into four categories, classification techniques, Statistical Techniques, ML techniques and finite state machines. The authors described advantages and disadvantages for techniques of each category with their future improvement in the field of IDSs. Tsai et al. [26] also reviewed ML-based IDSs and compare them based on classified design, experimental settings and benchmark datasets. They highlighted the challenges of effective IDSs and provided many future directions for research in this field. Wu et al. [27] presented a comprehensive survey of computational intelligence based intrusion detection techniques. They have highlighted applications of computational intelligence-based techniques in different fields for detecting intrusions. Their survey focuses on fuzzy system, artificial neural networks, artificial immune systems, soft computing paradigm, and evolutionary algorithms. Buczak et al. [28] focused ML techniques employed for detecting intrusions effectively. They divided ML techniques into 12 different categories and analyzed their computational complexity. Based upon their analysis of computational complexity, they recommended using ML techniques to detect intrusion in the network. Drasar et al. [29] studied flow-based intrusion detection techniques in their review paper. They targeted flow-based techniques based on similarity matching for detecting internet-based attacks. They proposed to group flow-based intrusion detection techniques based on their similarity functions. Vasilomanolakis et al. [30] focused on collaborative IDS. They identified the requirement for implementing collaborative IDS in large organizations. In the review, they proposed a taxonomy for collaborative IDSs. They divided the collaborative IDSs into centralized, decentralized, and distributed categories. They reviewed the vital research work for each category as per their taxonomy.

Similarly, Patcha et al. [31] also focused on ML techniques for IDSs. Whereas Hodo et al. [32] also focused on deep learning-based IDSs in their review.

Table 2 summarizes the technique-based IDS reviews mentioned above.

Table 2: Summary of technique-based IDS reviews

Study	Domain	ML techniques
Garcia-Teodoro et al. [24]	Networks	ML techniques
Kumar et al. [4]	Networks	AI-based techniques
Zhang et al. [25]	Networks	Anomaly detection techniques
Tsai et al. [26]	Networks	ML techniques

(Continued)

Table 2 (continued)

Study	Domain	ML techniques
Wu et al. [27]	Networks	Computational intelligence based techniques
Buczak et al. [28]	Networks	Data mining and ML algorithms
Drasar et al. [29]	Networks	Flow-based techniques
Vasilomanolakis et al. [33]	Networks	Collaborative intrusion detection techniques
Patcha et al. [31]	Networks	Supervised and unsupervised learning
Hodo et al. [32]	Networks	Supervised and unsupervised learning
Nguyen et al. [34]	Networks	Supervised and unsupervised learning

2.3 Attack-Based Reviews

The research work in this category has been proposed to classify different kinds of network intrusion. These papers follow a specific taxonomy of network inclusions and present a review of different techniques as per the adopted taxonomy. Such reviews are beneficial for comparing different intrusion detection techniques to detect specific kinds of intrusions. For example, Sperotto et al. [35] focused on flow-based intrusion detection techniques. The authors proposed a taxonomic classified network intrusion and flow-based techniques used to detect each intrusion category. They also highlighted the research issues specifically for flow-based IDSs and provided many directions for future research in IDS. Umer et al. [36] focused on flow-based IDSs and compared different intrusion detection techniques in different aspects. They presented different benchmark data sets used for validating flow-based intrusion detection techniques. They also proposed a taxonomy of intrusion detection techniques for detecting malicious network flows. They identified different research issues regarding flow-based IDSs and highlighted different research directions for future research in this field. Table 3 summarizes the attack-based IDS reviews mentioned above.

Table 3: Summary of attack-based IDS reviews

Study	Domain	ML techniques
Sperotto et al. [35]	Network attack classification	Flow-based techniques
Umer et al. [36]	Networks	Flow-based techniques

2.4 General-Purpose Reviews

This category of research work for IDS attempts to analyze network intrusions in different aspects. Such reviews follow a General taxonomy of intrusion and review the current research work as per the adopted taxonomy. For example, Patel et al. [14] focused on intrusion detection and prevention techniques. They identified the limitations of existing systems and proposed using ML-based techniques for detecting intrusions effectively and accurately. Liao et al. [37] proposed a taxonomy of IDS based on different aspects such as deployment, timeline, source of data and detection method. They identified several limitations of the existing method and highlighted different research directions in the field. Bhuyan et al. [38] reviewed the network anomaly detection techniques tools and systems. Their review proposed a taxonomy that divides our existing network anomaly detection techniques into six categories. They highlighted the advantages and disadvantages of each category.

They also highlighted the most commonly used performance metrics and data sets for validating intrusion detection techniques. Table 4 summarizes the General-purpose IDS reviews mentioned above.

Table 4: Summary of General-purpose IDS reviews

Study	Domain	ML techniques
Patel et al. [14]	Networks	ML and autonomic computing techniques
Liao et al. [37]	Networks	ML techniques
Bhuyan et al. [38]	Networks	Anomaly detection approaches
Buczak et al. [28]	Networks	Data mining and ML techniques
Usama et al. [39]	Networks	Unsupervised learning techniques

It can be concluded from Tables 1–4 that many researchers have successfully implemented ML techniques in different network scenarios. However, a few studies have been proposed for intrusion detection in SDN. To that end, we provide a comprehensive review of ML techniques proposed in recent years for intrusion detection, specifically for SDN. We aim to explore ML techniques, identify research gaps, and highlight future research directions in intrusion detection in context to SDN.

The above cited reviews can be summarized in Fig. 3.

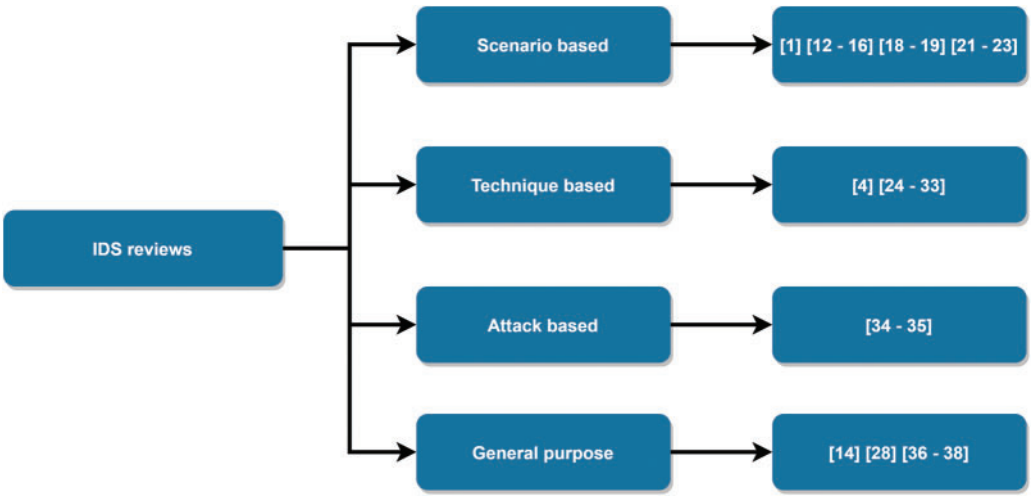

Figure 3: Summary of IDS reviews

3 Intrusion Detection

An IDS is defined as "an effective security technology, which can detect, prevent and possibly react to the computer attacks", is one of the standard components in security infrastructures [4]. It monitors target sources of activities, such as audit and network traffic data in a computer or network systems and deploys various techniques to provide security services. The main objective of IDS is to detect all intrusions efficiently. The implementation of IDS allows network administrators to detect security objective violations. These security objective violations range from external attackers trying to gain

unauthorized access to network security infrastructure or making resources unavailable to insiders abusing their access to the system resources. With the passage of time and the growth of computer attacks, several IDSs architectures have been proposed. Axelsson [40] proposed a common architecture for IDS as depicted in Fig. 4.

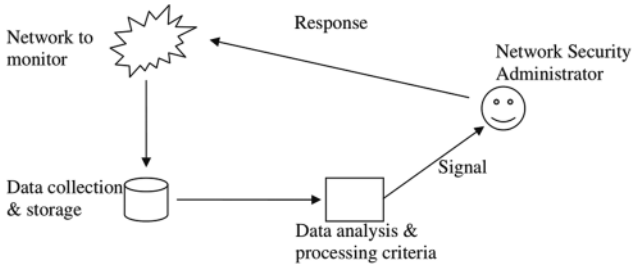

Figure 4: IDS architecture [4]

According to Axelsson [40], standard components of IDS consist of the following: Network to monitor is the identity to be monitored for intrusions. This can be a single host or a network; Data collection & storage unit is responsible for collecting the data of various events and converting them in proper format and store to disk; Data analysis & processing unit is the brain of IDS. It contains the complete functionality to find the suspicious behaviour of attack traffic. On detecting an attack, a signal is generated. Based on the type of IDS, the system can raise the action to alleviate the problem or a signal is passed to the network administrator to take appropriate action; Signal: This part of the system handles all output from IDS. The output may be an automated response to an intrusion or alert of malicious activity for a network security administrator. IDSs can be categorized into various classes depending upon different modules.

Based on data collected & storage unit, IDS can be divided into two classes: host-based IDS and Network-based IDS. Host-based IDS collects the data from a host to be protected. They generally collect the data from system calls, operating system logs, NT events log files, CPU utilization, application log files, etc. The advantage of Host-based IDS is that they are operating system dependent & are very efficient to detect attacks like buffer overflow. These systems become inefficient in the case of encrypted data and switched networks. Network-based IDS collects the data from the network directly in the form of packets. These IDS are operating system independent and easy to deploy to various systems.

Based upon criteria adopted for data analysis & processing unit, IDS can be divided into two classes; namely, Misuse or signature-based IDS and anomaly-based IDS. Signature-based IDSs maintain a database of known attack signatures. The detection of attack involves comparing data from the data collection unit and data stored in the database. If the match occurs, then an attack signal gets generated. The challenging task is to keep the database of signatures up to date. Signature-based IDS perform well for attacks whose signatures are in the database, but they are inefficient to detect zero-day attacks. They also have a meagre false alarm rate. Anomaly-based IDS reacts to abnormal behaviour as defined by some history of the monitored systems, previous behaviour or some previously defined profile. The system matches the current profile with the previous profile. If there is any significant deviation, that activity is notified as an attack. These systems are capable of detecting zero-day attacks.

Depending upon the criteria adopted for generating the response, IDS can be divided into two classes: Passive IDS and Active IDS. Active IDS responds to attacks by initiating specific actions. The action can be against two entities, further classifying Active IDS into subclasses. These entities can

be: Attacking system: In this class, the IDS try to control the attacking system. IDS tries to attack the attacker system to remove his operation platform. Attacked system: In this class, the IDS tries to control the attacked system. They modified the state of the attacked system to mitigate the attack. They can terminate the network connections, increase the security logging, kill the concerned processes, etc. Passive IDS respond to attacks by generating network administrator or user signals to act. They do not themselves try to mitigate the damage done or actively seek to harm or hamper the attacker.

The available commercially as well as open-source IDSs have been categorized and summarized based on different criteria mentioned-above as shown in Tables 5–7 and Figs. 5 and 6.

Table 5: Classification of IDSs (based on data collection & storage unit)

Category	IDS	Processing criteria	Audit data	Response
Host based IDS	Haystack [41]	Hybrid	Host	Passive
	Intrusion Detection Expert System (IDES) [42]	Anomaly	Host	Passive
	MIDAS [43]	Hybrid	Host	Passive
	OSSEC HIDS [44]	Hybrid	Host	Active
	Samhain [45]			
	Tripwire [46]	Signature	Host	Passive
	Bro [47]	Signature	Network	Passive
Network based IDS	Cisco Secure [48]		Network	
	EMERLARD [49]	Hybrid	Hybrid	Active
	NADIR [50]	Anomaly	Network	Passive
	NSM [51]	Hybrid	Network	Passive
	Snort [52]	Hybrid	Network	Active

Table 6: Classification of IDSs (based on data analysis & processing unit)

Category	IDS	Processing criteria	Audit data	Response
Misuse or signature based IDS	ASAX [53]	Signature	Host	Passive
	Intrusion Detection Expert System (IDES) [42]	Anomaly	Host	Passive
	Bro [47]	Signature	Network	Passive
	GrIDS [54]	Hybrid	Hybrid	Passive
	IDIOT [55]	Signature	Host	Passive
	RealSecure [56]	Signature	Hybrid	Active
	Suricata [57]	Signature	Hybrid	Active
	Security Onion [58]	Signature	Hybrid	Active
	AirMagnet [59]	Signature	Wireless network	Active
	WIPS-NG [60]	Signature	Wireless network	Active
	Sagan [61]	Signature	Host	Active

(Continued)

Table 6 (continued)

Category	IDS	Processing criteria	Audit data	Response
	Tripwire [46]	Signature	Host	Passive
Network based IDS	AAFID [62]	Anomaly	Host	Active
	Comp Watch [63]	Anomaly	Host	Passive
	IDES [42]	Anomaly	Host	Passive
	NADIR [50]	Anomaly	Network	Passive
	W&S [64]	Anomaly	Host	Passive

Table 7: Classification of IDSs (based on response)

Category	IDS	Processing criteria	Audit data	Response
Passive IDS	IDES [42]	Anomaly	Host	Passive
	GrIDS [54]	Hybrid	Hybrid	Passive
	NIDES [65]	Hybrid	Host	Passive
Active IDS	EMERLARD [49]	Hybrid	Hybrid	Active
	Janus [66]	Signature	Host	Active
	OSSEC HIDS [44]	Hybrid	Host	Active
	RealSecure [56]	Signature	Hybrid	Active

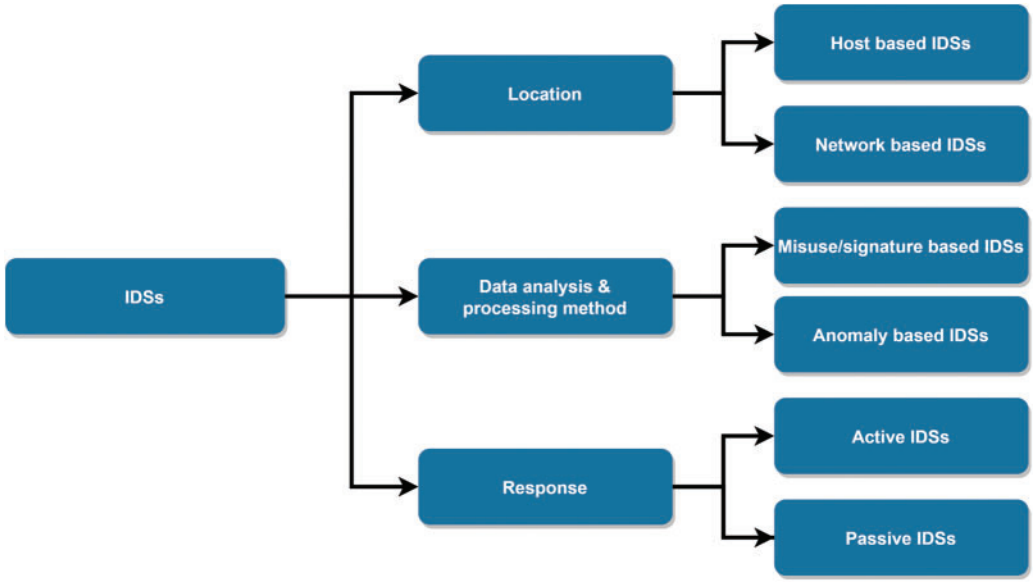

Figure 5: Summary of IDSs

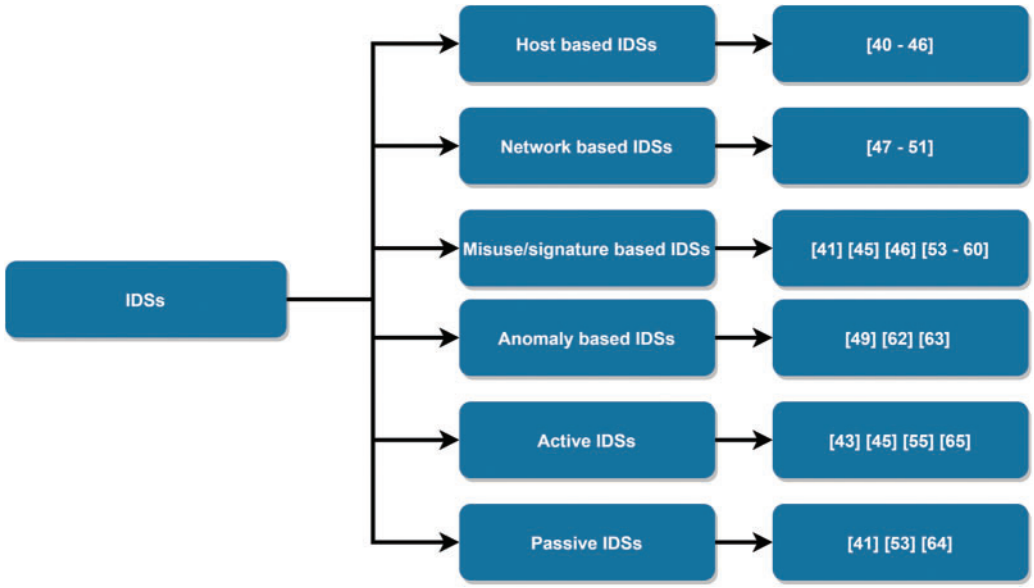

Figure 6: Summary of IDS studies

4 ML Techniques

Several techniques from different disciplines have been designed for developing effective and efficient IDS. Statistical techniques, Knowledge-based techniques and artificial intelligence (AI) based techniques are the trending techniques for IDS development. AI-based techniques, specifically ML (ML) techniques have many advantages of Flexibility (*vs.* Threshold definition of conventional technique); Adaptability (*vs.* specific rules of conventional technique); Pattern recognition (and detection of new patterns); Fast computing (faster than humans, actually) and Learning abilities [67]. ML techniques can learn from data automatically without explicit programming during the training phase [22].

Fig. 7 depicts a general work-flow of machine learning project [68–70]. The first phase consists of the data management phase of any ML project. It collects the data and uses it as training and test data for training and validation of the ML model. The data management phase also applies data cleaning management techniques for 1) data cleaning to remove missing values and noisy data; and 2) data transformation to normalize data, select relevant features, and discretize features for ensuring the quality of data and compatibility with the ML model. After pre-processing the data, it is split into training and test datasets and loaded for the training and test of the ML model.

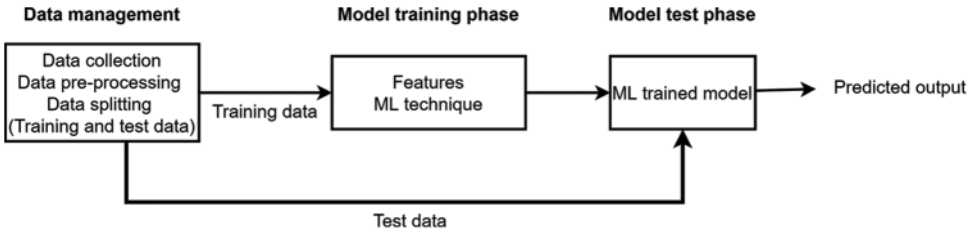

Figure 7: ML phases [4]

An appropriate ML model is chosen based on learning tasks such as classification, regression and clustering. The training dataset is fed to ML model for achieving optimized parameters during the training phase [71,72]. Finally, trained ML is evaluated for the test dataset by getting its predictions and comparing them with actual output. The performance of the trained ML model using suitable metrics like accuracy, true positive rate, false-positive rate, F1-score, kappa statistics, precision and recall. After achieving satisfaction on validation metrics and performance of ML model, it is deployed in real-world scenario for making actual predictions [32]. ML model are generally retrained for new training data to update it with changing scenarios up to a benchmark performance satisfaction.

Generally, ML techniques are classified based on learning style, such as supervised learning, unsupervised learning and semi-supervised learning, and reinforcement learning techniques [4,73] as presented in Fig. 8.

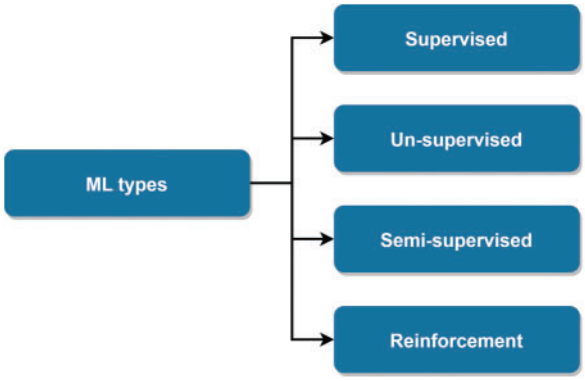

Figure 8: ML types

The supervised learning process consists of labelled training data samples [74]. In contrast, unsupervised learning of ML techniques used un-labelled data during the training phase. Reinforcement learning attempts to learn the problem by taking suitable action per given circumstances to optimize the objective function. ML techniques can be applied for predicting the class of data samples in a given discrete category (known as classification task) or estimating one or more continuous variables (known as regression task) [75].

Supervised learning has several potential benefits, such as clarity of data and ease of training [76,77]. However, there are many disadvantages, including the inability to learn by itself, requirement of labelled data. Supervised techniques take advantages of using prior knowledge to clearly classify unknown sample data. Supervised learning process is easy to understand, however, in case of unsupervised learning, it is difficult to understand machine learning process. Supervised learning does not require holding training data in memory after training phase. In stead, only mathematical function representing boundary function can be maintained for predicting unknown samples.

Supervised learning techniques generally provide biased results in case of imbalanced training datasets, hence it become difficult for dealing with a large amount of imbalanced training data. However, supervised learning cannot give you unknown information from the training data like unsupervised learning do. In contrast, un-supervised learning can cluster or classify data by discovering its features on its own that is not feasible in case of supervised learning.

Supervised and un-supervised learning have different goals. Supervised learning aims to predict outcomes for new data [78]. Expected result types are known in advance. Whereas, in case of un-supervised learning, the main aim is to get insights from large volumes of new data. The learning process itself determines what is different or interesting from the dataset. Supervised learning methods are computationally less complex than un-supervised learning methods. These models are generally time-consuming while their training, and the labels for input and output variables require expertise. Meanwhile, unsupervised learning methods can have wildly inaccurate results unless some human intervention for validating the output variables.

Reinforcement learning is different from supervised and un-supervised learning methods [79]. Here, the machine learns by itself after making several mistakes. From all the mistakes made, the machine can understand what the causes were, and it will try to avoid those mistakes again and again. Reinforcement learning is also known as the trial and error way of learning.

Popular supervised ML techniques include Naive Bayes, Nearest Neighbor, Decision Trees, Support Vector Machines (SVM), Linear Regression, Neural Networks. Different supervised ML techniques different concepts for classification tasks based on training dataset's features. For example, Decision Trees (DTs) refers to feature values. They use a tree-like model of decisions and their results. DT algorithm contains conditional control statements and branch symbolizes a feature of the dataset. Whereas, Naive Bayes (NB) algorithm works on independence assumption of all the datasets. NB suits for large datasets and uses direct acyclic graph for classification tasks. It is most appropriate for solving multi-class prediction models. This algorithm is computationally less expensive for handling huge and complex data. In contrast, Random Forests (RF) algorithm, an advanced version of DT, involves generating decision trees on data samples and then predicts for each attempt till best solution obtained. RF reduces the over-fitting issues of DT by taking average the result. Neural Networks (NN) algorithm involves clustering raw input and identify patterns. NN are comparatively computationally expensive and become more complicated for multiple observations. NNs are generally known as 'black-box' algorithms. Support Vector Method (SVM) involves separation of hyper-planes as discriminative classifiers. This method is concerned with kernel networks that produces an optimal hyperplane as output for binary classification problems.

Standard unsupervised ML techniques are k-means clustering, Hierarchical Cluster Analysis (HCA), Expectation Maximization, Locally-Linear Embedding (LLE), and t-distributed Stochastic Neighbor Embedding (t-SNE).

Standard reinforcement ML techniques include Q-Learning, Temporal Difference (TD), and Deep Adversarial Networks.

Tables 8–10 and Figs. 9–11 summarize the most common supervised, unsupervised and semi-supervised ML techniques with respective pros and cons.

Table 8: Summary of supervised ML techniques

ML technique	Pros	Cons
k-NN	Easy implementation Choice of distance functions	CPU intensive due to the distance calculation Memory intensive for storing all the training dataset

(Continued)

Table 8 (continued)

ML technique	Pros	Cons
Decision Tree	Easy interpretation Selection of discriminatory features Less CPU intensive Works with continuous and discrete data	Unstable, subject to training data Over-fitting issue
Random forest	suitable for large training data Comparatively less instability Avoids over-fitting problem	Slow training process Biased results in case of imbalanced data
Neural network	Quick prediction after training Suitable for high-dimensional data	Requires high computationally power for training Difficult to interpret the results
SVM	Suitable for high-dimensional data Suitable for linearly and non-linearly separable data	Computationally expensive for large data Avoids over-fitting problem
Bayesian network	Easy implementation Good results for a small training data	Independence assumption Difficult to handle continuous data
HMM	Statistical fundamentals Instable	Computationally expensive for large data

Table 9: Summary of unsupervised ML techniques

ML technique	Pros	Cons
k-means	Easy implementation	Depends on initialazation points and outliers
	Easy interpretation	Computational expensive
SOM	Easy understanding	Computationally expensive
	Works well with high-dimensional data	

Table 10: Summary of semi-supervised ML techniques

ML technique	Pros	Cons
Semi-supervised learning	Work with labelled and unlabeled data	Depends on many assumptions
Reinforcement learning	Requires no prior knowledge Faster predictions after training	Slow convergence Difficult with high-dimensional data

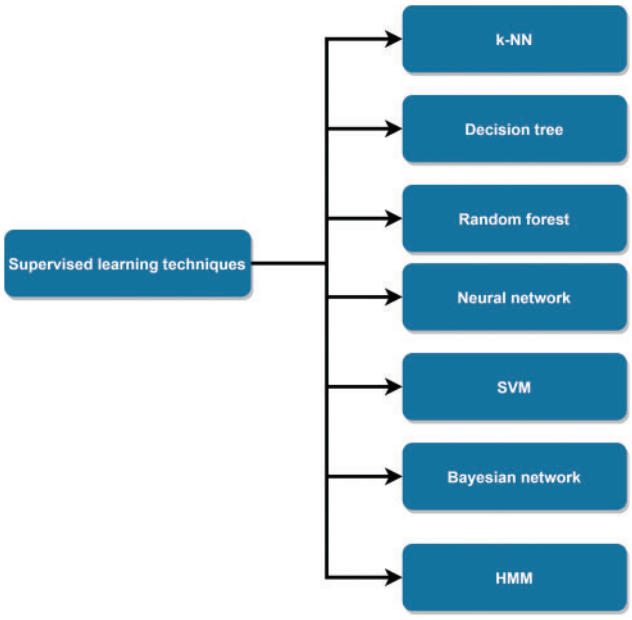

Figure 9: Supervised learning techniques

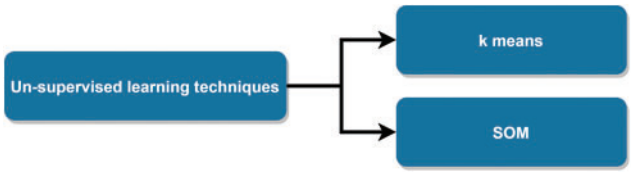

Figure 10: Un-supervised learning techniques

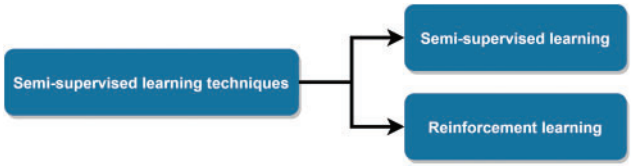

Figure 11: Semi-supervised learning techniques

5 Software Defined Networking (SDN) and Its Architecture

SDN enables flexibility in network control by decoupling the control plane and data plane in a conventional network. It helps the network administrators in customizing the network as per dynamic requirements of the organizations [80], presented in Fig. 12 [81].

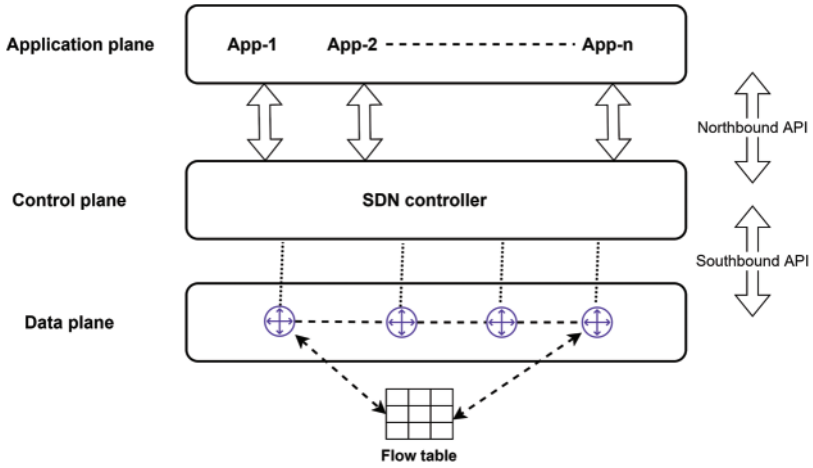

Figure 12: SDN architecture [81]

The decoupling of control plane and data plane allows data plane devices called switches in forwarding data as per decisions of the controller [82]. The controller decisions are maintained in the form of flow tables of switches. OpenFlow protocol is used for ensuring communication between controller and switch.

Fig. 12 shows SDN architecture and interaction of different planes. SDN architecture consists of three planes: data plane, control plane and application plane.

- **Data plane:** This plane is responsible for forwarding data among different nodes of the network using various forwarding devices. Several forwarding devices, virtual switches and physical switches can be equipped in this plane. The most common virtual devices at this layer include Open vSwitch [83], Indigo and Pantou switches. Whereas, physical switches includes NetFPGA [84], SwitchBlade [85] and ServerSwitch [86]. Virtual switches have exclusive features of SDN but provide a low flow forwarding rate. In contrast, physical switches possess limited flexibility but show a higher flow forwarding rate. These switches forward, drop and modify data packets as per policies provided in the control plane. The communication between the data plane and control plane occurs through Southbound Interfaces (SBIs).
- **Control plane:** It is the central controlling part of SDN systems. It enables network device programming, maintains forwarding rules, and provides flexibility in the SDN. Logically Central controller is the primary component in the control plane of SDN architecture. The central controller controls the communication between different applications and forwarding devices at the data plane. The central controller also allows the translation of application requirements into respective policies for forwarding devices. It also provides the functionality of network application requirements such as network topology storage, shortest path routing. Several central controller architectures have been proposed, including NOX [87], POX [87], Floodlight [88], Ryu [89], OpenDaylight [90] and Beacon [91]. There are three interfaces for interacting with the controllers, southbound, northbound and eastbound/westbound interfaces. A southbound interface defines the communication between the data and control planes. This interface enables forwarding devices to transmit network state information and control policies to and from the control plane. It also provides functionality for programming of all devices for or forwarding operation notifications and statistical reports. The northbound interface enables communication between the application plane and the control plane. Applications can access

abstract network perspectives provided by the control plane using northbound interfaces to define network behaviour and requirements. The northbound interface helps in automating, innovating and managing the SDN. Eastbound/westbound interfaces are mainly used in a multi-controller SDN. These interfaces are deployed in a large scale SDN consisting of a massive amount of data flows.

- **Application plane:** This is the top layer in SDN system architecture consisting of business applications. It enables new network services for managing and optimizing business applications. The business applications access network state information through the controllers for implementing control logic to update the network behaviour.

SDN flexibility feature helps reduce dependence on software and hardware vendors, thus reducing operational expenses. It also enables node level security implementation by replacing firewalls with flow tables of switches. Despite several advantages, SDN architecture has several security vulnerabilities due to the single point of failure of SDN controller [92]. Single point of failure of central SDN controller can lead to failure of the entire network. Most attackers target the central SDN controller to control the entire network [93]. Several attacks such as Denial of Service (DoS) attack, black-hole attack [94], malicious controller application deployment, and global network view manipulation [92] can be easily mounted by compromising SDN controller.

Data plane is also suspectable to several attacks, including flow-table overflow attacks. Such attacks exploit the flow table's limited size and non-availability of standards. Security issues at different SDN planes can be further explored in [22,81,82]

6 ML Techniques for Intrusion Detection in SDN

SDN architecture comprising of a central controller that provides a global network perspective [1]. The global network perspective helps manage and control network easily. It provides an edge for ML techniques for analyzing network data and optimizing network configuration and other functionality by adding intelligence to the SDN central controller. Besides, the programmability feature of SDN also allows to detect and mitigate network attacks quickly. Notably, from a security perspective, ML techniques have been successfully applied in SDN to differentiate intrusive and non-intrusive network traffic.

Several industrial and academic efforts have been made to address the security problems of SDN, considering its wide acceptability. Researchers focused on improving security by adopting SDN in conventional networks, and the security of SDN framework [80,92,95–98]. Song et al. [99] suggested an IDS for SDN architecture. The proposed architecture comprises different subsystems: data preprocessor, predictive data model, and response system. The authors proposed using the feature selection method for data processing to select relevant features, followed by the decision tree and random forest method to differentiate intrusive and non-intrusive network traffic. Based on classification results, the proposed architecture makes the decision and triggers the response using reactive routing in different flow tables. The experimental deserts of the proposed architecture demonstrate that the threat-aware system can reduce the data processing and provide high intrusion detection accuracy.

Similarly, Hurley et al. [100] also proposed a network IDS for SDN using Hidden Markov Model (HMM) based upon selected flow traffic features: packet length, source sport, destination port, source IP address and destination IP address.

In contrast, da Silva et al. [101] proposed a framework called ATLANTIC. The proposed framework can detect the anomalies in SDN network traffic and classify them into different categories. This framework performs classification tasks in two phases: lightweight and heavyweight faces. The former phase computes the derivation of network traffic based on entropy values of flow tables. At the same time, the later phase applies an SVM classifier to classify the abnormal network traffic. The classification is followed by mitigation actions to handle abnormal network flows.

Similarly, the authors of [102] also proposed an intrusion detection and mitigation system for the smart home environment based on ML techniques for detecting inclusive activities.

In [103], the authors used different ML techniques for predicting malicious connections and vulnerable hosts. They used decision tree (DT), decision tables (D table), Bayesnet and Naive Bayes (NB) ML techniques. They performed a comprehensive comparison of ML techniques. They demonstrated in their results that BayesNet could produce more accurate results than the other techniques.

Some researchers also focused on deep learning techniques for detecting intrusions in SDN. For example, Tang et al. [104] used a deep neural network ok for detecting inclusions in SDN. They use the KDD dataset for validating the proposed approach.

Similarly, They also used a deep recurrent neural network for detecting anomalies in SDN traffic using six flow features in [105].

Wang et al. [106] proposed an approach for detecting intrusions in SDN using SVM classifier. Their approach applied a feature selection method to select relevant features using a decision tree followed by classifying network traffic into intrusive and non-intrusive categories.

Shone et al. [107] proposed a hybrid approach of deep learning and random forest method. The deep learning method reduces the features, and the random forest is applied for classification network traffic.

The researchers have focused on detecting DDoS attacks targeting the availability of SDN. DDoS attacks exhaust the network or system resources by sending tremendous traffic into the network. The enormous network traffic makes the system unavailable to legitimate users.

Braga et al. [108] proposed a lightweight DDoS attack detection system and implemented it on a NOX a based SDN. They used network traffic flow features collected using OpenFlow switches at NOX controller. The collected features are used for classifying attacks and normal network traffic. They used a self-organising map neural network for detecting flooding based DDoS attacks in SDN. They demonstrated that their proposed system provide promising result in detecting DDoS attacks. However, they have not installed any flow rules in their system.

Barki et al. [109] implemented an IDS in SDN controller for detecting DDoS attacks using a hybrid approach of Signature and advanced IDS. They've used different ML techniques in signature-based IDS modules: k-NN, Naive Bayes, k-means and k-medoids. The packets detected as abnormal are forwarded to the advanced IDS module to differentiate anomalous or legitimate traffic.

Li et al. [110] also applied recurrent neural networks and convolutional neural networks in detecting DDoS attacks. Their deep learning architecture consists of input, forward recursive, reverse recursive, and fully connected hidden layers followed by an output layer for detecting DDoS attacks based upon the features extracted using deep learning models. Similarly, Jankowski et al. [111] used a self-organizing map (SOM) along with a learning vector quantization (LVQ) method for detecting intrusion in SDN.

Similarly, Niyaz et al. [112] used deep learning techniques stacked autoencoder for feature reduction to detect the DDOS attacks in SDN. They reported that their system could detect the DDOS attacks but have a controller bottleneck in an extensive network.

Table 11 summarizes the above-cited studies of ML techniques for intrusion detection in SDN. Fig. 13 presents dataset wise analysis of intrusion detection studies in SDNs. It can be observed that most researchers preferred KDD dataset for validating their intrusion detection approaches in SDNs.

Table 11: Summary of ML techniques for intrusion detection in SDN

Study	Learning method	Pros	Cons	Dataset	Avg. Acc. (%)
[99]	DT, RF	Use of reactive routing for installing flow rules corresponding to flow types	Ignored relevant contextual information	KDD	82.48 (DT), 98.75 (RF)
[100]	HMM	Detecting malicious traffic using HMM	Independence assumption of features Not present in real world events	Synthetic	88
[101]	SVM	Detection of intrusive traffic using SVM	Manual inspection of unknown traffic flows	Synthetic	88.7
[102]	SVM	SVM used for detecting malicious traffic in smart devices	Limited use of the proposed system all smart devices in a home network	Synthetic	96.2
[103]	DT, BayesNet, D table, NB	Multiple ML techniques applied to detect malicious connections and vulnerable hosts	Blockage of entire network for avoiding attacks	Synthetic	86.19 (DT), 91.68 (Bayes), 88.52 (D table), 87.78 (NB)
[104]	DL-NN	DL-NN model for differentiating intrusive and non-intrusive traffic	Limited features used	NSL-KDD	75.75
[105]	RNN	DL-NN model based anomaly detection	Limited features used	NSL-KDD	89

(Continued)

Table 11 (continued)

Study	Learning method	Pros	Cons	Dataset	Avg. Acc. (%)
[106]	Hybrid of DT and SVM	Use of reduced features using decision tree for accurate classification by SVM	Comparative result not provided	KDD	97.55
[107]	Hybrid of DL-NN and RF	Use of reduced features using DL-NN for accurate classification by RF	Not evaluated in real backbone traffic	KDD Cup'99 and NSL-KDD	99.79
[108]	SOM	DDoS attack detection using SOM	Unable to detect attack launching hosts	KDD	98.61
[110]	DL-NN	DDoS attack detection and defense method based on DL-NN	-	ISCX	98
[112]	DL-NN	stack auto-encoder based DL model for reducing features	Controller bottleneck for large networks Computational cost for SDN controller for extracting features and attack detection	Synthetic	95.65
[111]	Hybrid of SOM and LVQ	Used SOM and LVQ for intrusion detection	Not evaluated in real backbone traffic Poor results for minority attack classes like U2R and R2L	Synthetic	TPR = 99.6
[113]	RF	Used RF for intrusion detection in SDN	Not evaluated in real backbone traffic	CICIDS 2017	99.968

(Continued)

Table 11 (continued)

Study	Learning method	Pros	Cons	Dataset	Avg. Acc. (%)
[114]	SVM	Used selective logging for IP Traceback in SDN Low computational overhead Ability to track the actual source of the packets in the eventuality of an attack	Used outdated KDD dataset	KDD	95.98 (Full KDD dataset), 87.74 (selective features)
[115]	Tree-based machine learning techniques	XGBoost model outperformed	Used outdated KDD dataset	KDD	95.95
[116]	SVM	Used Mininet emulator based virtual network	Not evaluated on real datasets	UNSW-NB15 and NSL-KDD datasets	99.8
[117]	GRU and BiLSTM	Hybrid model with GRU, GRU-LSTM, deep neural network, DNN-LSTM	Not evaluated on real datasets	CICIDS 2018	99.87
[118]	Stacked auto-encoder	Hybrid model of stacked auto-encoder, SoftMax classifier and parameter optimizer	Not evaluated on real datasets	NSL-KDD and CICIDS 2017	98.5

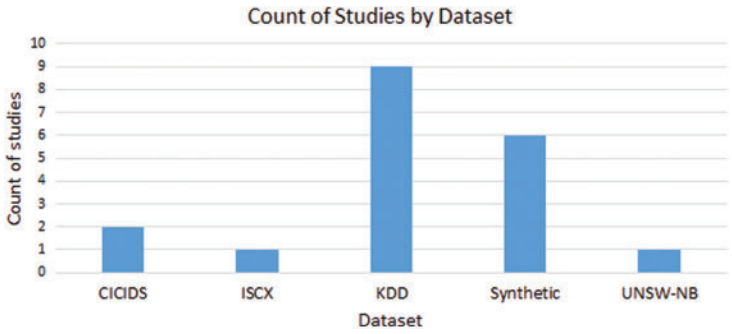

Figure 13: Dataset wise analysis

Fig. 14 presents accuracy analysis of intrusion detection studies in SDNs. It can be observed that researchers reported an accuracy of 99.96% and 99.79% based on CICIDS and KDD datasets, respectively.

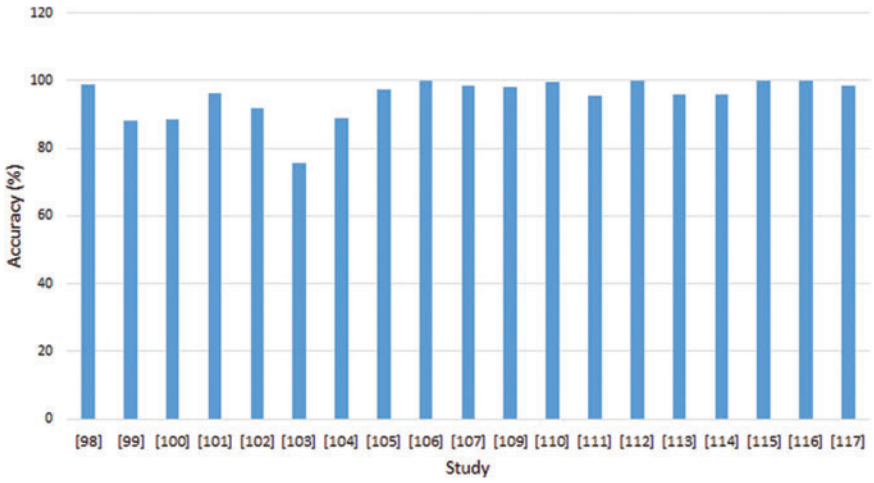

Figure 14: Accuracy analysis

7 Research Challenges and Future Directions

Despite much prominent research in ML and SDN fields, there is a requirement to improve robustness and security by addressing many significant challenges. The most significant research challenges that require the community's immediate attention follows:

- To improve the intelligence in SDN using ML techniques, quality training data set are required [5,119]. ML techniques require a high-quality training data set for training models that can be used to detect intrusions. However, the lack of publicly available updated benchmark datasets leads to the failure to validate new approaches. Therefore, there is a requirement for developing benchmark data set [120,121].
- It can be observed from the discussion cited in Section 6 that many IDS suffers from the limitation of scalability in SDN. A single controller deployment can be a significant cause for scalability issues in SDN [122–124].

- To solve the scalability issue, distributed multi-controller platforms can be is promising direction [125,126].
- SDNs involve decoupling of the data plane and control plane to provide a flexibility feature. The data plane comprises forwarding devices without any intelligence. This can be a severe flaw in the system that the attacker can exploit to launch many attacks. The attack can be overloading the controller by forwarding a massive amount of flow requests. In this scenario, ML model trained on historical data may not effectively detect new attack variants. This issue can be resolved by using recent developments in deep learning techniques such as generative adversarial network (GAN) [71,75,127,128].
- SDN implementation requires updating network switches that can be economically costlier. Therefore, incremental deployment of SDN can be a promising solution for handling the deployment issue of SDN [129,130].
- Training time and accuracy of ML techniques are highly dependent upon features selected for the training of ML models. However, selecting appropriate features for training the ML model is challenging. Feature selection techniques for automatically selecting high-level features can be a promising solution to this issue [131–133].
- It can be noticed that ML techniques achieved exemplary performance and flexibility by learning and representing real-world problem features as nested hierarchy of concepts in a simple way [134,135]. However, the performance of ML techniques depends upon the quality of training data and handcrafted features. In contrast, a deep learning technique can learn incrementally using its layered architecture and can extract high-level features automatically from data with minimal human interaction [136,137]. Several deep learning architectures have been developed for different types of the task such as CNN, ResNet, Inception Nets, RNN and LSTM. Deep learning techniques can be a promising research direction for detecting intrusions accurately without requiring handcrafted features, particularly in SDN due to the availability of centralized data.
- It can be observed from Table 11 that many researchers have used outdated KDD dataset for validating their approach. KDD dataset have been critically analyzed for not representing real-world network traffic [138,139].
- Deploying the SDN in large networks can face the performance issue due to the processing of massive network traffic. Therefore, successful deployment of SDN IDS requires reduction of controller bottleneck [140–142].

8 Conclusion

This study presented a comprehensive review of ML techniques for detecting intrusion detection in SDN. It presented intrusion detection, ML techniques, and types, followed by SDN and its architecture. We explained the benefits of using SDN. We presented prominent research on using ML techniques for detecting intrusion in SDN. We provided a comprehensive comparison of different studies describing the pros and cons of each study. Finally, we presented and discussed significant research issues and future directions for applying ML to detect SDN intrusions.

In a nutshell, it can be concluded that the application of ML techniques in detecting intrusion in SDN faces many challenges. The findings of this study can help fellow researchers understand the development of ML-based intrusion detection in the SDN context.

Funding Statement: This work is supported by King Khalid University, Saudi Arabia under Grant No. RGP.2/61/43.

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

References

1. Xie, J., Yu, F. R., Huang, T., Xie, R., Liu, J. et al. (2018). A survey of machine learning techniques applied to software defined networking (SDN) Research issues and challenges. *IEEE Communications Surveys & Tutorials, 21(1),* 393–430. DOI 10.1109/COMST.2018.2866942.
2. Hamed, T., Ernst, J. B., Kremer, S. C. (2018). A survey and taxonomy of classifiers of intrusion detection systems. In: Daimi, K. (Ed.), *Computer and network security essentials,* pp. 21–39. Cham: Springer.
3. Halme, L. (1995). Ain't misbehaving–A taxonomy of anti-intrusion techniques. *Computers and Security, 14(7),* 606–606. DOI 10.1016/0167-4048(96)81669-5.
4. Kumar, G., Kumar, K., Sachdeva, M. (2010). The use of artificial intelligence based techniques for intrusion detection: A review. *Artificial Intelligence Review, 34(4),* 369–387. DOI 10.1007/s10462-010-9179-5.
5. Mestres, A., Rodriguez-Natal, A., Carner, J., Barlet-Ros, P., Alarcón, E. et al. (2017). Knowledge-defined networking. *ACM SIGCOMM Computer Communication Review, 47(3),* 2–10. DOI 10.1145/3138808.3138810.
6. Thakur, K., Alqahtani, H., Kumar, G. (2021). An intelligent algorithmically generated domain detection system. *Computers & Electrical Engineering, 92,* 107129. DOI 10.1016/j.compeleceng.2021.107129.
7. Varghese, J. E., Muniyal, B. (2021). An efficient IDS framework for DDoS attacks in SDN environment. *IEEE Access, 9,* 69680–69699. DOI 10.1109/ACCESS.2021.3078065.
8. Ashraf, J., Moustafa, N., Bukhshi, A. D., Javed, A. (2021). Intrusion detection system for SDN-enabled IoT networks using machine learning techniques. *2021 IEEE 25th International Enterprise Distributed Object Computing Workshop (EDOCW)*, pp. 46–52. DOI 10.1109/EDOCW52865.2021.00031.
9. Wang, M., Cui, Y., Wang, X., Xiao, S., Jiang, J. (2017). Machine learning for networking: Workflow, advances and opportunities. *IEEE Network, 32(2),* 92–99. DOI 10.1109/MNET.2017.1700200.
10. Xu, G., Mu, Y., Liu, J. (2017). Inclusion of artificial intelligence in communication networks and services. *ITU Journal: ICT Discoveries, (1),* 1–6.
11. Kumar, G. (2020). An improved ensemble approach for effective intrusion detection. *The Journal of Supercomputing, 76(1),* 275–291.
12. Anantvalee, T., Wu, J. (2007). A survey on intrusion detection in mobile ad hoc networks. In: *Wireless network security*, pp. 159–180. Boston, MA: Springer.
13. Nadeem, A., Howarth, M. P. (2013). A survey of manet intrusion detection & prevention approaches for network layer attacks. *IEEE Communications Surveys & Tutorials, 15(4),* 2027–2045. DOI 10.1109/SURV.2013.030713.00201.
14. Patel, A., Qassim, Q., Wills, C. (2010). A survey of intrusion detection and prevention systems. *Information Management & Computer Security, 18(4),* 277–290. DOI 10.1108/09685221011079199.
15. Butun, I., Morgera, S. D., Sankar, R. (2013). A survey of intrusion detection systems in wireless sensor networks. *IEEE Communications Surveys & Tutorials, 16(1),* 266–282. DOI 10.1109/SURV.2013.050113.00191.
16. Bkassiny, M., Li, Y., Jayaweera, S. K. (2012). A survey on machine-learning techniques in cognitive radios. *IEEE Communications Surveys & Tutorials, 15(3),* 1136–1159. DOI 10.1109/SURV.2012.100412.00017.
17. Alsheikh, M. A., Lin, S., Niyato, D., Tan, H. P. (2014). Machine learning in wireless sensor networks: Algorithms, strategies, and applications. *IEEE Communications Surveys & Tutorials, 16(4),* 1996–2018. DOI 10.1109/COMST.2014.2320099.

18. Wang, X., Li, X., Leung, V. C. (2015). Artificial intelligence-based techniques for emerging heterogeneous network: State of the arts, opportunities, and challenges. *IEEE Access, 3,* 1379–1391. DOI 10.1109/ACCESS.2015.2467174.
19. Klaine, P. V., Imran, M. A., Onireti, O., Souza, R. D. (2017). A survey of machine learning techniques applied to self-organizing cellular networks. *IEEE Communications Surveys & Tutorials, 19(4),* 2392–2431. DOI 10.1109/COMST.2017.2727878.
20. Fadlullah, Z. M., Tang, F., Mao, B., Kato, N., Akashi, O. et al. (2017). State-of-the-art deep learning: Evolving machine intelligence toward tomorrows intelligent network traffic control systems. *IEEE Communications Surveys & Tutorials, 19(4),* 2432–2455. DOI 10.1109/COMST.2017.2707140.
21. Chen, M., Challita, U., Saad, W., Yin, C., Debbah, M. (2017). Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks. arXiv preprint arXiv:1710.02913.
22. Sultana, N., Chilamkurti, N., Peng, W., Alhadad, R. (2019). Survey on sdn based network intrusion detection system using machine learning approaches. *Peer-to-Peer Networking and Applications, 12(2),* 493–501. DOI 10.1007/s12083-017-0630-0.
23. Zhou, A., Qu, B., Li, H., Zhao, S., Suganthan, P. et al. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation, 1(1),* 32–49. DOI 10.1016/j.swevo.2011.03.001.
24. Garcia-Teodoro, P., Diaz-Verdejo, J., Macia-Fernandez, G., Vazquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security, 28(1–2),* 18–28. DOI 10.1016/j.cose.2008.08.003.
25. Zhang, W., Yang, Q., Geng, Y. (2009). A survey of anomaly detection methods in networks. *2009 International Symposium on Computer Network and Multimedia Technology*, pp. 1–3. IEEE, Wuhan, China.
26. Tsai, C., Hsu, Y., Lin, C., Lin, W. (2009). Intrusion detection by machine learning: A review. *Expert Systems with Applications, 36(10),* 11994–12000. DOI 10.1016/j.eswa.2009.05.029.
27. Wu, S., Banzhaf, W. (2010). The use of computational intelligence in intrusion detection systems: A review. *Applied Soft Computing, 10(1),* 1–35. DOI 10.1016/j.asoc.2009.06.019.
28. Buczak, A. L., Guven, E. (2015). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials, 18(2),* 1153–1176. DOI 10.1109/COMST.9739.
29. Drasar, M., Vizvary, M., Vykopal, J. (2014). Similarity as a central approach to flow-based anomaly detection. *International Journal of Network Management, 24(4),* 318–336. DOI 10.1002/nem.1867.
30. Vasilomanolakis, E., Karuppayah, S., Muhlhauser, M., Fischer, M. (2015). Taxonomy and survey of collaborative intrusion detection. *ACM Computing Surveys, 47(4),* 1–33. DOI 10.1145/2716260.
31. Patcha, A., Park, J. M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks, 51(12),* 3448–3470. DOI 10.1016/j.comnet.2007.02.001.
32. Hodo, E., Bellekens, X., Hamilton, A., Tachtatzis, C., Atkinson, R. (2017). Shallow and deep networks intrusion detection system: A taxonomy and survey. arXiv preprint arXiv:1701.02145.
33. Vasilomanolakis, E., Karuppayah, S., Mühlhäuser, M., Fischer, M. (2015). Taxonomy and survey of collaborative intrusion detection. *ACM Computing Surveys, 47(4),* 1–33. DOI 10.1145/2716260.
34. Nguyen, T. T., Armitage, G. (2008). A survey of techniques for internet traffic classification using machine learning. *IEEE Communications Surveys & Tutorials, 10(4),* 56–76. DOI 10.1109/SURV.2008.080406.
35. Sperotto, A., Schaffrath, G., Sadre, R., Morariu, C., Pras, A. et al. (2010). An overview of IP flow-based intrusion detection. *IEEE Communications Surveys & Tutorials, 12(3),* 343–356. DOI 10.1109/COMST.9739.
36. Umer, M. F., Sher, M., Bi, Y. (2017). Flow-based intrusion detection: Techniques and challenges. *Computers & Security, 70,* 238–254. DOI 10.1016/j.cose.2017.05.009.

37. Liao, H. J., Lin, C. H. R., Lin, Y. C., Tung, K. Y. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications, 36(1),* 16–24. DOI 10.1016/j.jnca.2012.09.004.
38. Bhuyan, M. H., Bhattacharyya, D. K., Kalita, J. K. (2013). Network anomaly detection: Methods, systems and tools. *IEEE Communications Surveys & Tutorials, 16(1),* 303–336. DOI 10.1109/SURV.2013.052213.00046.
39. Usama, M., Qadir, J., Raza, A., Arif, H., Yau, K. L. A. et al. (2019). Unsupervised machine learning for networking: Techniques, applications and research challenges. *IEEE Access, 7,* 65579–65615. DOI 10.1109/Access.6287639.
40. Axelsson, S. (2000). Intrusion detection systems: A survey and taxonomy. *Technical Report*.
41. Smaha, S. (1988). Haystack: An intrusion detection system. *Fourth Aerospace Computer Security Applications Conference*, vol. 44. IEEE.
42. Lunt, T. (1993). A survey of intrusion detection techniques. *Computers & Security, 12(4),* 405–418. DOI 10.1016/0167-4048(93)90029-5.
43. Sebring, M., Shellhouse, E., Hanna, M., Whitehurst, R. (1988). Expert systems in intrusion detection: A case study. *Proceedings of the 11th National Computer Security Conference*, pp. 74–81. Baltimore, Maryland.
44. Hay, A., Cid, D., Bray, R. (2008). *OSSEC host-based intrusion detection guide*. Syngress.
45. Samhain (2010). The samhain file integrity/intrusion detection system. Samhain Labs. http://la-samhna.de/samhain/.
46. Kim, G., Spafford, E. (1997). Tripwire: A case study in integrity monitoring. In: *Internet besieged: Countering cyberspace scofflaws*, pp. 175–210.
47. Paxson, V. (1999). Bro: A system for detecting network intruders in real-time. *Computer Networks, 31(23),* 2435–2463. DOI 10.1016/S1389-1286(99)00112-7.
48. Secure, C. (2010). Ids. http://www.cisco.com/warp/public/cc/pd/sqsw/sqidsz/index.shtml.
49. Porras, P., Neumann, P. (1997). Emerald: Event monitoring enabling response to anomalous live disturbances. *Proceedings of the 20th National Information Systems Security Conference*, vol. 3, pp. 353–365.
50. Hochberg, J., Jackson, K., Stallings, C., McClary, J., DuBois, D. et al. (1993). Nadir: An automated system for detecting network intrusion and misuse. *Computers & Security, 12(3),* 235–248. DOI 10.1016/0167-4048(93)90110-Q.
51. Heberlein, L., Dias, G., Levitt, K., Mukherjee, B., Wood, J. et al. (1989). *A network security monitor*. *Proceedings of IEEE Computer Society Symposium on Research in Security and Privacy*, IEEE. Lawrence Livermore National Lab., CA, USA.
52. Beale, J., Baker, A., Caswell, B., Poor, M. (2004). *Snort 2.1 intrusion detection*. Elsevier.
53. Habra, N., Charlier, B., Mounji, A., Mathieu, I. (1992). Asax: Software architecture and rule-based language for universal audit trail analysis. In: *Computer security-ESORICS 92*, pp. 435–450. *Second European Symposium on Research in Computer Security Toulouse*, France. DOI 10.1007/BFb0013888.
54. Staniford-Chen, S., Cheung, S., Crawford, R., Dilger, M., Frank, J. et al. (1996). Grids-A graph based intrusion detection system for large networks. *Proceedings of 19th National Information Systems Security Conference*, vol. 1. Baltimore.
55. Crosbie, M., Dole, B., Ellis, T., Krsul, I. (1996). E. spa ord. idiot-users guide. *Technical Report TR-96-050*. Purdue University, COAST Laboratory.
56. Systems, I. S. (2012). Real secure. http://www.iss.net.
57. Nam, K., Kim, K. (2018). A study on SDN security enhancement using open source IDS/IPS Suricata. *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 1124–1126. DOI 10.1109/ICTC.2018.8539455.
58. Mikail, A., Pranggono, B. (2019). Securing infrastructure-as-a-service public clouds using security onion. *Applied System Innovation, 2(1),* 6. DOI 10.3390/asi2010006.

59. AirMagnet (2008). http://www.airmagnet.com.
60. WIPS-NG (2020). http://openwips-ng.org/index.html.
61. Segan (2020). https://quadrantsec.com/sagan_log_analysis_engine/.
62. Spafford, E., Zamboni, D. (2000). Intrusion detection using autonomous agents. *Computer Networks, 34(4),* 547–570. DOI 10.1016/S1389-1286(00)00136-5.
63. Dowell, C. (1990). The computerwatch data reduction tool. *Proceedings of the 13th National Computer Security Conference*, pp. 99–108. Washington DC, USA.
64. Liepins, G., Vaccaro, H. (1989). Detection of anomalous computer session activity. *Proceedings of the 1996 USENIX Security Symposium*, vol. 19. Berkeley, CA, USENIX Association.
65. Anderson, D., Frivold, T., Valdes, A. (1995). *Next-generation Intrusion Detection Expert System (NIDES): A Summary*. SRI International, Computer Science Laboratory.
66. Goldberg, I., Wagner, D., Thomas, R., Brewer, E. (1996). A secure environment for untrusted helper applications confining the wily hacker. *Proceedings of the 6th Conference on USENIX Security Symposium, Focusing on Applications of Cryptography*, vol. 6. USENIX Association, Berkeley, CA.
67. Ponce, M. C. (2004). Intrusion detection system with artificial intelligence. *FIST Conference Edition*, Universidad Pontificia Comillas de Madrid.
68. Dini, P., Saponara, S. (2021). Analysis, design, and comparison of machine-learning techniques for networking intrusion detection. *Designs, 5(1),* 9. DOI 10.3390/designs5010009.
69. Afuwape, A. A., Xu, Y., Anajemba, J. H., Srivastava, G. (2021). Performance evaluation of secured network traffic classification using a machine learning approach. *Computer Standards & Interfaces, 78,* 103545. DOI 10.1016/j.csi.2021.103545.
70. Rani, S., Barak, D. D., Singh, Y. (2021). Enhancing performance of network traffic classification using machine learning: A review. *EFFLATOUNIA-Multidisciplinary Journal, 5(2)*.
71. Aledhari, M., Razzak, R., Parizi, R. M. (2021). Machine learning for network application security: Empirical evaluation and optimization. *Computers & Electrical Engineering, 91,* 107052. DOI 10.1016/j.compeleceng.2021.107052.
72. Le Jeune, L., Goedemé, T., Mentens, N. (2021). Machine learning for misuse-based network intrusion detection: Overview, unified evaluation and feature choice comparison framework. *IEEE Access, 9,* 63995–64015. DOI 10.1109/ACCESS.2021.3075066.
73. Aburomman, A. A., Reaz, M. B. I. (2016). Survey of learning methods in intrusion detection systems. *2016 International Conference on Advances in Electrical, Electronic and Systems Engineering (ICAEES)*, pp. 362–365. IEEE, Putrajaya, Malaysia.
74. Zaheer, A., Asghar, M. Z., Qayyum, A. (2021). Intrusion detection and mitigation framework for SDN controlled IoTs network. *2021 IEEE 18th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET)*, pp. 147–151. IEEE, Karachi, Pakistan.
75. Goodfellow, I., Bengio, Y., Courville, A. (2016). *Deep learning*, vol. 1. MIT Press.
76. Wang, Z., Zeng, Y., Liu, Y., Li, D. (2021). Deep belief network integrating improved kernel-based extreme learning machine for network intrusion detection. *IEEE Access, 9,* 16062–16091. DOI 10.1109/Access.6287639.
77. Singh, G., Khare, N. (2021). A survey of intrusion detection from the perspective of intrusion datasets and machine learning techniques. *International Journal of Computers and Applications,* 1–11. DOI 10.1080/1206212X.2021.1885150.
78. Kocher, G., Kumar, G. (2021). Machine learning and deep learning methods for intrusion detection systems: Recent developments and challenges. *Soft Computing, 25(15),* 9731–9763. DOI 10.1007/s00500-021-05893-0.

79. Sethi, K., Madhav, Y. V., Kumar, R., Bera, P. (2021). Attention based multi-agent intrusion detection systems using reinforcement learning. *Journal of Information Security and Applications, 61,* 102923. DOI 10.1016/j.jisa.2021.102923.
80. Anand, N., Babu, S., Manoj, B. (2018). On detecting compromised controller in software defined networks. *Computer Networks, 137,* 107–118. DOI 10.1016/j.comnet.2018.03.021.
81. Shaghaghi, A., Kaafar, M. A., Buyya, R., Jha, S. (2020). Software-defined network (SDN) data plane security: Issues, solutions, and future directions. In: Gupta, B., Perez, G., Agrawal, D., Gupta, D. (Eds), *Handbook of computer networks and cyber security*, pp. 341–387. Cham: Springer. DOI 10.1007/978-3-030-22277-2_14.
82. Dabbagh, M., Hamdaoui, B., Guizani, M., Rayes, A. (2015). Software-defined networking security: Pros and cons. *IEEE Communications Magazine, 53(6),* 73–79. DOI 10.1109/MCOM.2015.7120048.
83. Pfaff, B., Pettit, J., Koponen, T., Jackson, E., Zhou, A. et al. (2015). The design and implementation of open vSwitch. *12th USENIX Symposium on Networked Systems Design and Implementation*, pp. 117–130. Oakland, CA.
84. Lockwood, J. W., McKeown, N., Watson, G., Gibb, G., Hartke, P. et al. (2007). Netfpga–An open platform for gigabit-rate network switching and routing. *2007 IEEE International Conference on Microelectronic Systems Education (MSE'07)*, pp. 160–161. IEEE, San Diego, CA, USA.
85. Anwer, M. B., Motiwala, M., Tariq, M. B., Feamster, N. (2010). Switchblade: A platform for rapid deployment of network protocols on programmable hardware. *Proceedings of the ACM SIGCOMM 2010 Conference*, pp. 183–194.
86. Lu, G., Guo, C., Li, Y., Zhou, Z., Yuan, T. et al. (2011). Serverswitch: A programmable and high performance platform for data center networks. *8th USENIX Symposium on Networked Systems Design and Implementation (NSDI 11)*.
87. Sheikh, M. N. A. (2019). SDN-based approach to evaluate the best controller: Internal controller NOX and external controllers POX, ONOS, RYU. *Global Journal of Computer Science and Technology, 19(1)*, 21–32.
88. Morzhov, S., Alekseev, I., Nikitinskiy, M. (2016). Firewall application for Floodlight SDN controller. *2016 International Siberian Conference on Control and Communications (SIBCON)*, pp. 1–5. IEEE, Moscow, Russia.
89. Tomonori, F. (2013). Introduction to RYU SDN framework. https://ryu-sdn.org/slides/ONS2013-april-ryu-intro.pdf.
90. Khattak, Z. K., Awais, M., Iqbal, A. (2014). Performance evaluation of OpenDaylight SDN controller. *2014 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 671–676. IEEE, Hsinchu, Taiwan.
91. Erickson, D. (2013). The beacon openflow controller. *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, pp. 13–18.
92. Yoon, C., Lee, S., Kang, H., Park, T., Shin, S. et al. (2017). Flow wars: Systemizing the attack surface and defenses in software-defined networks. *IEEE/ACM Transactions on Networking, 25(6),* 3514–3530. DOI 10.1109/TNET.2017.2748159.
93. Hoque, N., Bhattacharyya, D. K., Kalita, J. K. (2015). Botnet in DDoS attacks: Trends and challenges. *IEEE Communications Surveys & Tutorials, 17(4),* 2242–2270. DOI 10.1109/COMST.2015.2457491.
94. Scott-Hayward, S., Natarajan, S., Sezer, S. (2015). A survey of security in software defined networks. *IEEE Communications Surveys & Tutorials, 18(1),* 623–654. DOI 10.1109/COMST.2015.2453114.
95. Peleh, N., Shpur, O., Klymash, M. (2022). Intelligent detection of DDoS attacks in SDN networks. In: Klymash, M., Beshley, M., Luntovskyy, A. (Eds.), *Lecture notes in electrical engineering,* vol. 831. Cham: Springer.

96. Bhayo, J., Hameed, S., Shah, S. A., Nasir, J., Ahmed, A. et al. (2022). A novel DDoS attack detection framework for software-defined IoT (Sd-IoT) networks using machine learning. *SSRN Electronic Journal*, 4022910.
97. Sudar, K. M., Deepalakshmi, P. (2022). Flow-based detection and mitigation of low-rate DDOS attack in sdn environment using machine learning techniques. In: *IoT and analytics for sensor networks*, pp. 193–205. Springer.
98. Muthamil Sudar, K., Deepalakshmi, P. (2021). An intelligent flow-based and signature-based IDS for SDNs using ensemble feature selection and a multi-layer machine learning-based classifier. *Journal of Intelligent & Fuzzy Systems, 40(3),* 4237–4256. DOI 10.3233/JIFS-200850.
99. Song, C., Park, Y., Golani, K., Kim, Y., Bhatt, K. et al. (2017). Machine-learning based threat-aware system in software defined networks. *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–9. IEEE. Vancouver, BC, Canada.
100. Hurley, T., Perdomo, J. E., Perez-Pons, A. (2016). Hmm-based intrusion detection system for software defined networking. *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 617–621. IEEE, Anaheim, CA, USA.
101. da Silva, A. S., Wickboldt, J. A., Granville, L. Z., Schaeffer-Filho, A. (2016). Atlantic: A framework for anomaly traffic detection, classification, and mitigation in SDN. *2016 IEEE/IFIP Network Operations and Management Symposium*, pp. 27–35. IEEE, Istanbul, Turkey.
102. Nobakht, M., Sivaraman, V., Boreli, R. (2016). A host-based intrusion detection and mitigation framework for smart home IoT using openflow. *2016 11th International Conference on Availability, Reliability and Security (ARES)*, pp. 147–156. IEEE, Salzburg, Austria.
103. Nanda, S., Zafari, F., DeCusatis, C., Wedaa, E., Yang, B. (2016). Predicting network attack patterns in SDN using machine learning approach. *2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pp. 167–172. IEEE, Palo Alto, CA, USA.
104. Tang, T. A., Mhamdi, L., McLernon, D., Zaidi, S. A. R., Ghogho, M. (2016). Deep learning approach for network intrusion detection in software defined networking. *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, pp. 258–263. IEEE, Fez, Morocco.
105. Tang, T. A., Mhamdi, L., McLernon, D., Zaidi, S. A. R., Ghogho, M. (2018). Deep recurrent neural network for intrusion detection in SDN-based networks. *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*, pp. 202–206. IEEE, Montreal, QC, Canada.
106. Wang, P., Chao, K. M., Lin, H. C., Lin, W. H., Lo, C. C. (2016). An efficient flow control approach for SDN-based network threat detection and migration using support vector machine. *2016 IEEE 13th International Conference on e-Business Engineering (ICEBE)*, pp. 56–63. IEEE, Macau, China.
107. Shone, N., Ngoc, T. N., Phai, V. D., Shi, Q. (2018). A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence, 2(1),* 41–50. DOI 10.1109/TETCI.2017.2772792.
108. Braga, R., Mota, E., Passito, A. (2010). Lightweight DDoS flooding attack detection using NOX/openflow. *IEEE Local Computer Network Conference*, pp. 408–415. IEEE, Denver, CO, USA.
109. Barki, L., Shidling, A., Meti, N., Narayan, D., Mulla, M. M. (2016). Detection of distributed denial of service attacks in software defined networks. *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2576–2581. IEEE, Jaipur, India.
110. Li, C., Wu, Y., Yuan, X., Sun, Z., Wang, W. et al. (2018). Detection and defense of DDoS attack–based on deep learning in openflow-based SDN. *International Journal of Communication Systems, 31(5),* e3497. DOI 10.1002/dac.3497.
111. Jankowski, D., Amanowicz, M. (2016). On efficiency of selected machine learning algorithms for intrusion detection in software defined networks. *International Journal of Electronics and Telecommunications, 62(3),* 247–252. DOI 10.1515/eletel-2016-0033.

112. Niyaz, Q., Sun, W., Javaid, A. Y. (2016). A deep learning based DDoS detection system in software-defined networking (SDN). arXiv preprint arXiv:1611.07400.
113. Zeleke, E. M., Melaku, H. M., Mengistu, F. G. (2021). Efficient intrusion detection system for sdn orchestrated Internet of Things. *Journal of Computer Networks and Communications, 2021*, 5593214. DOI 10.1155/2021/5593214.
114. Hadem, P., Saikia, D. K., Moulik, S. (2021). An SDN-based intrusion detection system using SVM with selective logging for IP traceback. *Computer Networks, 191,* 108015. DOI 10.1016/j.comnet.2021.108015.
115. Alzahrani, A. O., Alenazi, M. J. (2021). Designing a network intrusion detection system based on machine learning for software defined networks. *Future Internet, 13(5),* 111. DOI 10.3390/fi13050111.
116. Ibrahim, O. J., Bhaya, W. S. (2021). Intrusion detection system for cloud based software-defined networks. *Journal of Physics: Conference Series, 1804(1),* 012007.
117. Javeed, D., Gao, T., Khan, M. T., Ahmad, I. (2021). A hybrid deep learning-driven SDN enabled mechanism for secure communication in Internet of Things (IoT). *Sensors, 21(14),* 4884. DOI 10.3390/s21144884.
118. Choobdar, P., Naderan, M., Naderan, M. (2022). Detection and multi-class classification of intrusion in software defined networks using stacked auto-encoders and CICIDS2017 dataset. *Wireless Personal Communications,* 123, 437–471. DOI 10.1007/s11277-021-09139-y.
119. Janiesch, C., Zschech, P., Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets, 31(3),* 685–695. DOI 10.1007/s12525-021-00475-2.
120. Thakkar, A., Lohiya, R. (2021). A review on machine learning and deep learning perspectives of IDS for IoT: Recent updates, security issues, and challenges. *Archives of Computational Methods in Engineering, 28(4),* 3211–3243. DOI 10.1007/s11831-020-09496-0.
121. Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies, 32(1),* e4150. DOI 10.1002/ett.4150.
122. Kreutz, D., Ramos, F. M., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S. et al. (2014). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE, 103(1),* 14–76. DOI 10.1109/JPROC.2014.2371999.
123. Yazdinejadna, A., Parizi, R. M., Dehghantanha, A., Khan, M. S. (2021). A Kangaroo-based intrusion detection system on software-defined networks. *Computer Networks, 184,* 107688. DOI 10.1016/j.comnet.2020.107688.
124. Ghaffar, Z., Alshahrani, A., Fayaz, M., Alghamdi, A. M., Gwak, J. et al. (2021). A topical review on machine learning, software defined networking, Internet of Things applications: Research limitations and challenges. *Electronics, 10(8),* 880. DOI 10.3390/electronics10080880.
125. Hassas Yeganeh, S., Ganjali, Y. (2012). Kandoo: A framework for efficient and scalable offloading of control applications. *Proceedings of the first Workshop on Hot Topics in Software Defined Networks*, pp. 19–24.
126. Lin, S. C., Akyildiz, I. F., Wang, P., Luo, M. (2016). QoS-aware adaptive routing in multi-layer hierarchical software defined networks: A reinforcement learning approach. *2016 IEEE International Conference on Services Computing (SCC)*, pp. 25–33. IEEE, San Francisco, CA, USA.
127. Duy, P. T., Khoa, N. H., Nguyen, A. G. T., Pham, V. H. (2021). Digfupas: Deceive IDS with gan and function-preserving on adversarial samples in SDN-enabled networks. *Computers & Security, 109,* 102367. DOI 10.1016/j.cose.2021.102367.
128. Sreerag, V., Aswin, S., Menon, A. A., Namboothiri, L. V. (2022). Reinforce NIDS using GAN to detect U2R and R2L attacks. In: Karuppusamy, P., Perikos, I., García Márquez, F. P. (Eds.), *Ubiquitous intelligent systems. Smart innovation, systems and technologies*, vol. 243, pp. 357–369. Singapore: Springer.
129. Yurekten, O., Demirci, M. (2021). SDN-based cyber defense: A survey. *Future Generation Computer Systems, 115,* 126–149. DOI 10.1016/j.future.2020.09.006.

130. Khorsandroo, S., Sanchez, A. G., Tosun, A. S., Arco, J. M., Doriguzzi-Corin, R. (2021). Hybrid SDN evolution: A comprehensive survey of the state-of-the-art. *Computer Networks, 192,* 107981. DOI 10.1016/j.comnet.2021.107981.
131. Rashid, M., Kamruzzaman, J., Imam, T., Wibowo, S., Gordon, S. (2022). A tree-based stacking ensemble technique with feature selection for network intrusion detection. *Applied Intelligence,* 1–14. DOI 10.1007/s10489-021-02968-1.
132. Dora, V., Lakshmi, V. N. (2022). Optimal feature selection with CNN-feature learning for DDoS attack detection using meta-heuristic-based LSTM. *International Journal of Intelligent Robotics and Applications,* 1–27. DOI 10.1007/s41315-022-00224-4.
133. Sharma, B., Sharma, L., Lal, C. (2022). Feature selection and deep learning technique for intrusion detection system in IoT. In: Tiwari, R., Mishra, A., Yadav, N., Pavone, M. (Eds.), *Algorithms for intelligent systems.* Singapore: Springer.
134. Picón Ruiz, A., Medela Ayarzaguena, A., Sánchez Peralta, U., Cicchi, J. Bilbao, R. (2020). Why deep learning performs better than classical machine learning? *Dyna Ingenieria e Industria*. DOI 10.6036/DYNAII.
135. Liu, H., Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. *Applied Sciences, 9(20),* 4396. DOI 10.3390/app9204396.
136. Otoum, Y., Liu, D., Nayak, A. (2019). DL-IDS: A deep learning–based intrusion detection framework for securing IoT. *Transactions on Emerging Telecommunications Technologies*, e3803.
137. Tsimenidis, S., Lagkas, T., Rantos, K. (2022). Deep learning in IoT intrusion detection. *Journal of Network and Systems Management, 30(1),* 1–40. DOI 10.1007/s10922-021-09621-9.
138. McHugh, J. (2000). Testing intrusion detection systems: A critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by Lincoln laboratory. *ACM Transactions on Information and System Security, 3(4),* 262–294. DOI 10.1145/382912.382923.
139. Tavallaee, M. (2011). *An adaptive hybrid intrusion detection system* (*Ph.D. Thesis)*. University of New Brunswick.
140. Yan, Q., Yu, F. R., Gong, Q., Li, J. (2015). Software-defined networking SDN and distributed denial of service DDOS attacks in cloud computing environments: A survey, some research issues, and challenges. *IEEE Communications Surveys & Tutorials, 18(1),* 602–622. DOI 10.1109/COMST.9739.
141. Ghosh, U., Chatterjee, P., Shetty, S. (2022). Securing SDN-enabled smart power grids: SDN-enabled smart grid security. In: *Research anthology on smart grid and microgrid development*, pp. 1028–1046. IGI Global.
142. Zhou, Q., Zhao, T., Chen, X., Zhong, Y., Luo, H. (2022). A Fault-tolerant transmission scheme in SDN-based industrial IoT (IIoT) over fiber-wireless networks. *Entropy, 24(2),* 157. DOI 10.3390/e24020157.