Network Threat Detection with ML/DL in SDN-Based Platforms

Subjects: Computer Science, Artificial Intelligence

Contributor: Naveed Ahmed , Asri bin Ngadi , Johan Mohamad Sharif , Muhammad Siraj Rathore , Mueen Uddin , Saddam Hussain , Jawaid Iqbal , Maha Abdelhaq , Fatima Tul Zuhra , Raed Alsaqour , Syed Sajid Ullah

A revolution in network technology has been ushered in by software defined networking (SDN), which makes it possible to control the network from a central location and provides an overview of the network's security. Despite this, SDN has a single point of failure that increases the risk of potential threats. Network intrusion detection systems (NIDS) prevent intrusions into a network and preserve the network's integrity, availability, and confidentiality. Much work has been done on NIDS but there are still improvements needed in reducing false alarms and increasing threat detection accuracy. Advanced approaches such as deep learning (DL) and machine learning (ML) have been implemented in SDN-based NIDS to overcome the security issues within a network.

software defined network | intrusion detection systems | machine learning | deep learning

1. Introduction

Over the last two decades, network technologies have tremendously improved; at the same time, network security threats have also increased. Web-based security attacks, denial-of-service (DoS), and malicious insiders are a few examples that cause the devastating cybercrimes. With such malicious activities, critical disruptions can occur within a network. To ensure network security, antivirus software, firewalls, and network intrusion detection systems (NIDS) can be deployed. Among these, NIDS is broadly used for detecting intruders within a network by continuously monitoring the network traffic for any suspicious and malicious behavior. NIDS is useful to detect different kinds of network threats, including distributed denial-of-service (DDoS) attacks, worms, and viruses. Reliability, accuracy, and detection speed are the success factors of NIDS. Enormous research work has been done on NIDS, but it still requires improvements in reducing false alarm and increasing detection accuracy. To reduce false alarm rate [1] and increase threat detection accuracy [2], different approaches of machine learning (ML) have been used in NIDS. The advanced type of ML that is deep learning (DL) is also used in developing a more advanced field of NIDS.

Software defined networking (SDN) has revolutionized network technology in recent years. In contrast to a traditional network, SDN decouples the control plane and data plane of a network switch. In SDN, the control plane is moved to a remote controller (server), which can add packet forwarding rules in network switches according to a given program. This central control of a network offers more programmability and visibility compared with a traditional network. In addition, it is also attractive from a network security perspective, as having central control can offer better network monitoring [3].

In SDN, innovative network applications can be developed to monitor and control the network. In this regard, NIDS is extended for SDN-based architecture. To enhance network security and traffic monitoring, different approaches of ML/DL can be implemented in the controller of SDN. From the past few years, the invention of graphics processor units (GPUs) has increased the popularity of ML/DL approaches in network security. Both ML and DL techniques are very efficient at predicting any malicious or suspicious behavior from network traffic, as they can extract and learn new features from network traffic. ML-based NIDS heavily depend upon the learned features from network traffic, whereas DL-based NIDS automatically learn from the raw data of complex features and do not rely on learned features [4].

Many researchers have worked on ML- and DL-based NIDS in order to improve its performance in detecting network intruders. However, in larger networks, security threats are also increased due to increased network traffic, which affects the efficiency of NIDS in detecting malicious activities. Very few studies have been conducted on developing SDN-based NIDS systems through DL approaches so that there is enough room to deploy these techniques for improving detection efficiency of intrusions within a network.

2. Background

2.1. General Architecture of SDN

SDNs configure the whole network through programming, with a central location, as per organizational business needs. SDN is a network emerging paradigm adopted by telecommunication industries including Cisco Systems and Google. It decouples the control plane (i.e., intelligence of a network) from the data plane. A SDN-enabled network device (such as a router or network switch) performs forwarding only (i.e., data plane), whereas a remote controller implements the control plane. As a result, the controller controls the entire network and maintains a global view of the whole network. The separation of control and data planes in SDN can enhance the visibility, adaptability, and other local security operations of the network.

Learning and teaching have been profoundly impacted by the development of technology and proliferation of the Internet. E-learning has emerged from these developments and is generally understood as “the use of computer network technology, typically through an intranet or over the internet, to provide information and instruction to people.” However, e-learning faces several obstacles, such as the wide variety of learning styles and complications that may arise from cultural differences [5].

A SDN-enabled network device maintains a flow table that is consulted to perform a forwarding decision for an incoming packet. An incoming packet is matched against a flow table entry. This matching can be performed on the different header values (such as IP address and port number) of an incoming packet. For a matching flow, an action is listed in the flow table. For instance, a particular packet could either be dropped, forwarded on a particular output port, or forwarded to the controller.

Flow table entries are populated by the external controller. Within the application or module running on the controller, the forwarding rules are defined. To modify the data plane with the help of an external application, an application programming interface (API) is offered by SDN. In SDN, the capabilities of controlling the network have increased compared with traditional networks. The reason for this is that SDN implements a flow-based structure. As SDN is software-based, it is easy to modify policies in SDN that are less prone to errors. As it is programming-based, complex functions in the network can be developed in simpler ways [6].

2.1.1. Data Plane

Another name for the data plane is the infrastructure layer. Normally, there are various network devices in the data plane, including virtual switches and physical switches that are interconnected with each other through wireless or wired media. The data plane is responsible for sending network traffic on to predefined destination by the control plane, which is also called the forwarding plane. Virtual switches are based on software that can be operated through Linux. Examples of virtual switches implementations are Pantou [7], Indigo, and Open vSwitch [8]. Physical switches are based on hardware. Physical switches can be of two types; one type of physical switch can be implemented on networking hardware, whereas the other type is implemented on open network hardware (e.g., NetFPGA [9]). Two open network hardware-based physical switches are ServerSwitch [10] and SwitchBlade [11]. These data plane switches work according to received policies from the control plane, and as a result, they can modify, drop, or forward a packet.

2.1.2. Control Plane

The control layer of the SDN, also named the central layer, is considered the SDN brain, and includes controllers that are responsible for the programming and monitoring of the network and terminating and creating flows. The control layer is also accountable for routing. For example, it identifies the path in which data has to be forwarded using a routing algorithm. On one hand, the control plane extracts information from the data plane and transmits it to application plane; on the other hand, the control plane translates the application plane requirements into policies, and sends these policies to network switches (forwarding elements, FEs). Moreover, the control plane includes features such as providing state information notifications, device configuration, network topology storage, and shortest path routing etc. Beacon, OpenDayLight [12], Ryu, Flood-light [13], POX, and NOX [14] are different kinds of controllers. The control plane can interact within and outside the plane, with the help of three communication interfaces: the westbound/eastbound interface, southbound interface, and northbound interface [15].

2.1.3. Southbound Interface

The southbound interface is responsible for the interaction between the control plane and data plane. Through the southbound interface, a controller communicates with a switch, for instance, to add a new entry in flow table. Apart from forwarding operations, other important information (such as statistics reports and event notifications) are also exchanged through the southbound interface.

2.1.4. The Northbound Interface

The northbound interface is an API communication interface connecting the control layer and application layer. Through the northbound interface, the control plane translates the application plane requirements into policies, and sends these policies to FEs of the data plane [16].

2.1.5. Westbound/Eastbound Interfaces

The westbound/eastbound interfaces are used in SDN containing the multi-controller. When SDN is deployed in large networks, one controller may be unable to process the large amount of network traffic or data flows, so the larger networks are divided into smaller domains with separate controllers dedicated to each domain. Thus, communication between these separate controllers is necessary so that the global network view can be presented to the application plane; this communication is where westbound/eastbound interfaces are used.

2.2. Network Intrusion Detection System

Threats are detected by monitoring packets using IDS. Malicious and abnormal activities are detected by IDS from both the inside and outside [17]. Highly rough distribution of data and vast volumes of network traffic are some problems related to the IDS.

Networks or computers are some of the information sources that are monitored by the IDS, as its main function is to report illegal activities or access. Data from various network sources and systems are collected and analyzed by IDS for all possible threats and attacks. IDS has a wide array of implementations, including systems from tiered monitoring systems to antivirus software by which traffic of a complete network is followed. It can be categorized into the following classes:

- Incoming network traffic analyzed by the system known as NIDS.
- Important files of the operating system are monitored by the system and defined as “Host-based intrusion detection systems (HIDS)”.
- The aforementioned classifications of IDS are further classified. Signature and anomaly detection are the basis of commonly used variants [18].

2.2.1. Signature-Based Detection

Possible threats are detected by signature-based IDS by considering some special patterns, such as some known intrusion sequences that are malicious and used by Trojan or some byte sequences used in the traffic of network. This terminology originated from antivirus software that referred to these detected patterns as signatures. Known attacks are easily detected by signature-based IDS, but detection of new attacks for which there is no recognizable pattern is not possible [19].

2.2.2. Anomaly-Based Detection

This is a new technique that was designed for the adaption and detection of unknown attacks mainly caused by explosion of malware. ML is used in this detection method to model a trustworthy activity, and then the new behavior of the newly developed model is compared with the true model. Although unknown attacks are detected by this approach, there is also a risk for false positives, i.e., the classification of unknown authentic activities can be reported as malicious [20]. In [21], an algorithm was designed for an anomaly-based system named the AdaBoost algorithm. In this algorithm, two feature selection approaches, i.e., principal component analysis (PCA) and ensemble feature selection (EFS), were utilized for selecting features from a novel set of data, CICIDS 2018. Experimental results showed that integration of EFS with AdaBoost gave better results compared with PCA with AdaBoost. An analysis related to passing traffic is performed by IDS located at a premeditated point in the network to monitor traffic from network devices, and then the traffic is matched on subnets to a library of all known threats. Once the identification of an attack is made, it senses any abnormal behavior and the administrator receives an alert [21].

3. ML- and DL-Based IDS in SDN

3.1. Machine Learning-Based IDS in SDN

In SDN, one of the most significant usage of IDS is to ensure security. As traffic statistics are provided by Open Flow protocol (communication medium between switches and the controller), using messages such as “Stats Request” and “Stats Response”, IDS is the most compelling tool for the identification of threats and anomalies. For both traditional and SDN environments, the operations of IDS are equally applicable.

In [22][23], some examples of IDS using traditional SDN techniques are explained. In [24], anomaly problems were identified leveraging SDN. The main intention of proposing an SDN-based solution was to determine the main problems regarding cloud computing environment security that would react when an attack occurred. On the other hand, the authors in [22] used NFV and NID to create a deep packet inspection system. In [25], the authors used a detection technique based on statistics to get rid of abnormalities in SDN. “Normal profile of traffic” is defined as it is the foundation of statistics analysis. Information based on statistics at the packet level for the network and RMS, such as size of packet and variance, are related to the traffic profile. Traffic of network can be characterized by using the Hurst parameter, H, for instance, to measure bustiness and self-similarity (H is higher when traffic is bustier) [23].

(a) Data Plane

In this part, all network devices are immersed of collector agents. Centralized collector records send flow and agent samples. Specific flow metrics are collected by the device configurations, and then they are exported to the collector. Currently, major vendors such as Cisco offer export support and built-in flow collection.

- (b) Control Plane

Records of network flow are collected by the data collector embedded in the module of the control plane. Then, the control plane filters the data and conducts feature extraction. Thus, different datasets are created and generated by the collector, which are crucial for the ML approach [26]. The OpenFlow controller should be communicating with all network devices known as data sources.

- (c) Application Plane

The constructed and implemented model of ML is used as an application of SDN. Various methods of ML using generated datasets of different flow collectors can be used as applications of SDN for different purposes. The operation of a network can be influenced by constructing different applications powered by the ML-based model. Applications of incident handling include selection of path and rule enforcement. In the following case, the ML model used is an application of SDN-based IDS.

It is predicted that network flows can be classified as being normal or malicious.

3.1.1. DoS, U2R, Probe, and R2L

Four types of attacks are addressed in the studies discussed in the following section: DoS, U2R, probe, and R2L. The NSL-KDD dataset used between all of these include the common characteristics and attacks classified in the four classes.

The D-NN was used by the authors in [27] to detect six features based on anomalies and are suitable for SDN: type of protocol, count, duration, srv count, src bytes, and dst bytes. The model was trained and tested by the authors, and compared with other processes, such as SVM, NB, multi-layer perceptron, J48, random tree (RT), and random forest (RF) methods. The paper explore the applications of DL used in a detection system known as flow-based anomaly detection. At the same time, the authors claimed that the development of machine learning is not fully completed. Ref. [28] presented a discussion of nine classifiers based on ML with a supervised learning technique. Different tests were performed on accuracy, recall, execution time, false alarm rate, area under curve of ROC, f1-measure, McNemar's test, and precision. PCA was used in the tests to reduce dimensions with DT, K-NN, LDA, NN, linear SVM, extreme learning machine (ELM), BaggingTrees, NB, RUSBoost, RF, AdaBoost, and LogitBoost. The results showed that performance of bagging and boosting techniques was higher than the other techniques. A subset of dataset features were selected as features in which content features were not included. A hybrid classification system of level-5 was used by the same authors in [29] for IDS, in a network that was not based on SDN. Use of flow-statistics was the main aim of the paper; these flow-statistics were provided by the controller for the development of NIDS. In the first level, k-NN was used as the classification method; for the second level, ELM was used. For the other levels, the hierarchical extreme learning machine (H-ELM) was used. One type of attack was detected by each level using the same preferred features from [29]. For scalability purposes, the implemented system was in the form of a POX controller module, rather than an application plane function. The effectiveness of the method chosen to select features was because the features could be directly accessed from the controller. The results showed improved accuracy compared with the other methods.

3.1.2. DDoS Attacks

In the following section, DDoS attacks are specifically investigated in the presented studies for two reasons. First, DDoS attacks have been focused on large sections of IDS studies. Second, attacks should be individually considered with the perspective of recent threats, for example, the Mirai botnet and Internet of Things (IoT) [30]. A specific application using SDN was presented by the authors in [31] to tackle challenges in anomaly detection regarding scalability. The scenario was wireless SDN, which enabled the E-Health system. Massive machine-type communications were the main feature of such a network, where humans do not interact. For semi-supervised operations, CPLE was the ML technique used, with offline training. The main intention was online testing, so running localized detection was allowed within the devices. The requirement for frequent network traffic collection was avoided by using online testing to update the model of anomaly detection. The features used for classification were similar to the features defined in [32]. An overview of IDS based on ML in SDN is provided by authors in [28].

Millions of people may experience significant power outages if an attacker exploits cyber security weaknesses. This entry addresses this problem by presenting an OPNET-based network model exposed to many DoS assaults, illustrating the cyber security features of IEC 61850-based digital substations [33].

Five ML techniques were investigated by the study to mitigate DDoS attacks and intrusion (support vector machine, neural networks, Bayesian networks, genetic algorithms, decision tree, and fuzzy logic). Each method was theoretically analyzed by researchers and a comparison scheme was generated that presented the advantages and disadvantages of the approaches. The review could be used to choose the best technique, in accordance with system requirements. In [34], the authors compared the SVM analysis in SDN with other approaches in defending against a DDoS attack. Threats to the controller regarding security and types of SDN-based DDoS attacks are briefly discussed in the paper. Moreover, four methods of SVM and a description of system was provided in the paper. For training and testing, the datasets used were the 1999 and 1998 DARPA, and a comparison of approach was done with bagging, RBF, random forest, J48, and naive Bayes methods. Highest accuracy was shown for the proposed SMV, at approximately 95%. The support vector classifier-based learning algorithm, where features are selected by using an ID3 decision tree, was used by authors in [35]. The following three components, along with the software testbed, was used to evaluate the model: (1) Data collection was done using the sFlow Toolkit. (2) For the virtual switch, Open vSwitch was used. (3) The controller used was Ryu and the dataset was KDD-Cup 1999.

The model used in [36] was the Dirichlet process mixture model, to mitigate DDoS attacks based on DNS. An owned dataset was used by the authors in [3]; they created a dataset to generate DDoS attacks. The IDS system was presented by the authors in [37] for the identification of DDoS attacks. They compared three methods: SVM with 97% accuracy, KNN BEST was 83% accurate, and naïve Bayes was approximately 83% accurate. Features considered as inputs included the number of packets, bandwidth, destination IP, protocol, source IP, and protocol. They used an owned dataset for testing. A proposal was presented by the authors of [16] to improve resiliency by detecting some DDoS attacks, preferably the SYN flood attack, in the SDN network. Three different techniques were studied for classification: NB, DT, and SVM. DT showed 99% recall, precision, and accuracy. KDD 99 was the

dataset used, with features including protocol, src port, src IP address, dst port, and dst IP address. Later, PCA was used by the researchers for reduction. DDoS attack detection and classification was done by researchers using an approach in the cloud environment [38]. They used a two-stage learning scheme with two stages using two techniques: Bayesian and multivariate Gaussian. The employed features were blacklist IP, dst IP, number of packets, src IP, and spoof dst IP. Although complementary elements to the ML method were included in the study, however they did not directly secure the SDN. As an alternative, some steps were for the security of cloud infrastructure.

3.1.3. Comparison of Various Approaches in SDN

When considering a wide range of attacks regarding cyber security, it is important to have five attacks, including DDoS, U2R, DoS, probe, and R2L. Though SDN is an innovative paradigm, it is still subject to all known attack types. New adapted attacks should also be considered by the papers community. Reviewing the adaption of approaches to SDN is essential for the recognition, prevention, and extenuation of attacks. In all traditional networks, the main point of applied ML approaches are to recognize attacks. ML uses miscellaneous techniques. Most of the studies reviewed investigated a single ML approach. One of two approaches from at least two methods was used in other papers; the techniques were compared or combined to improve anomaly detection. Half of the reviewed papers used neural networks (generic NN, CNN, RBM, ANN, and NEAT). SVM was another approach used in the reviewed papers. The naïve Bayes method was also presented in several articles. However, a set of six features suitable for SDN was presented by authors in [39], which were further used in four studies. In different cases, independent selection of techniques of features are conducted, which have to be included in ML. In [40], it is demonstrated that a network attack in SDN can be predicted with an accuracy of 91.68% using Bayesian Networks machine learning approach.

3.2. Deep Learning-Based IDS in SDN

There are several studies that have used DL-based IDS in SDN. Normally, seven different kinds of threat vectors exist in the SDN, of which three threat vectors are definite and linked with the controller application plane, controller data plane, and control plane. NIDS is broadly used for detecting intruders within the network by continuously monitoring any suspicious and malicious behavior in network traffic. Based on strategies for network attack detection, NIDS are mainly of two types. The first strategy compares network traffic with pre-defined intrusion samples, and this is called signature-based detection [41]. New kinds of attack strategies cannot be detected by this kind of NID system; despite this fact, this technique is still very popular and commonly used in commercial IDS. In the second strategy, network traffic is compared with a normal user behavior model and any deviations from normal behavior of traffic is marked as an anomaly, using ML approaches. This type of NIDS is called anomaly-based detection. This technique can even detect attacks that have never seen before. Normally, flow-based monitoring of network traffic is combined with the latter NIDS strategy, i.e., anomaly-based detection [42].

Flow-based network traffic monitoring relies upon packet header information, which is why it handles a lower amount of data compared with payload-based NIDs,. Applications of ML approaches are found in multiple zones of

computer science, including speech recognition, face detection, and intrusion detection systems; however, such ML applications have faced some issues. In [43], the authors discuss the various issues in which ML algorithm applications affect the NID system. Although deep learning research on NID systems that leverage SDN is in its pre-stages, it is gaining more attention among researchers due to its results. Until now, DL algorithms have been extensively used in different areas of computer science for image, face, and voice recognition and has had real success. Through DL algorithms, correlations in bulk amounts of raw data can be easily found, and due to this feature, it can be broadly used in the next generation of NID systems. With the help of DL-based NID systems, one can obtain high detection accuracy and even efficiently detect attacks that have never been seen. In another study [44] authors investigate how DL might be useful for detecting malicious java script code.

There are many advantages exhibited by the SDN-based NID system using deep learning algorithms, including quality of service, virtual management, and security enforcement. SDN eliminates dependency on hardware because the whole network can be configured through programming and SDN enables a global view of the entire network, providing the chance to strengthen the network security [45]. Different SDN-based NID systems using a deep learning algorithm are briefly overviewed and compared. Using emulation and simulation platforms, SDN can be developed with programmable features and software switch implementations. SDN can be easily implemented in both software and hardware environments with the help of protocol standards, i.e., OpenFlow [46]. OMNeT++ [47], NS-3, Mininet [48], and NS-2 are some other simulation tools used for SDN. In SDN, its control plane is considered its most vital part, and is also called the operating system of the entire network. The control plane of SDN is accountable for providing a global view of the entire network and communicate with all programmable network elements. Beacon, OpenDayLight [49], Ryu, Floodlight [12], POX, and NOX [13] are different kinds of controllers.

The focus of recent studies have been on employing DL algorithms in NIDS, rather than ML algorithms. Through DL algorithms, correlations in bulk amounts of raw data can be easily found, and due to this feature, it could be broadly used in the next generation of NID systems. Compared with the results of various NID systems based on ML algorithms, DL-based NID systems gave much better results in the context of SDN [14]. Most machine learning algorithms are trained in a supervised way and these can give good results in classification tasks, but not in logic modelling; deep learning-based algorithms outperformed ML algorithms in logic modelling. As attack behaviors consistently change, they introduce new types of attacks, and unsupervised learning approaches such as RNN, stacked deep auto-encoder, and hybrid approaches are the best options in detecting these attacks in SDN-based NID systems. Currently, researchers are focusing on SDN-based NIDS using DL algorithms for SOHO networks and their satisfactory results suggest that intrusion detection system accuracy has greatly improved because of SDN scalability and DL algorithms [50].

3.2.1. DDoS Attack Detection Using DL Algorithms

Different kinds of vulnerabilities exist in the SDN platform, due to which the architecture of SDN is being targeted by various kinds of attacks such as a DDoS attack. In a DDoS attack, the intended user cannot get access to the network resources or machine. Multiple bots or multiple people are usually responsible for a DDoS attack, e.g., to launch a DDoS attack, the intruder can take advantage of SDN characteristics against the application plane,

infrastructure plane, and controller of SDN. In the SDN environment, it is very easy to deploy a DDoS attack; preventing such an attack is very difficult [51][52]. The occurrence of DDoS attacks in SDN is increasing daily, since the advent of the internet. The major reason behind the increased occurrence of DDoS attacks is the development and emergence of botnets that are formed within a network by machines or bots when they are exploited with malware. According to [53], the increase in DDoS attacks in 2016 was 125.36% of those that occurred in 2015. In [54], a lightweight detection system for DDoS attacks based on SOM in SDN was suggested, with a 6-tuple feature extraction: growth of different ports and single-flows, percentage of air-flows, average of duration per flow and bytes per flow, and packets per flow. In the flow-table, statistics features are extracted at certain intervals and are used in the implementation of this proposed method, making it a light-weight system. However, as a downside, this system has some limitations; it cannot be used for traffic handling that is not based on flow rules. Enhanced power plant monitoring is now possible thanks to modern sensors. As part of the overall cogeneration procedure, cooling towers condense exhaust steam to cool the facility [55].

In [56], an SDN-based DDoS blocking application was designed that could block a DDoS attack. For attack detection, this scheme worked in cooperation with two targeted servers. This was a prototype study that was designed to detect HTTP flooding attacks. In [57], the authors proposed a technique to detect a DDoS attack by using entropy calculations in the SDN controller and a deep auto-encoder approach for feature reduction. To detect attack, a threshold value was implemented, and selection of the threshold value was based on experimental results. A downside to a vast network is that there is a controller bottleneck; it can also affect the reliability affects because the threshold value can change in different scenarios.

In [58], a system combining a fuzzy interface, a hard detection threshold under attack, and normal states based on real characteristics of traffic was proposed for the detection of DDoS attack. For attack detection, three features were chosen, including flow quality to a server, packet quantity distribution per flow, and interval time distribution. Currently, researchers are working on the detection of flow-based intrusion.

In [28], IDS was also placed by the authors in the control plane. NSLKDD was the used dataset and a meta-heuristic Bayesian network was the technique used for the classification of traffic. Phase of selection and extraction of features was included in the proposed process to optimize the classifier. Fitness evaluation of the features which were selected was included in the classifier. After that, a Bayesian classifier was used. Seven other approaches are used were the comparison, but this proposed method was more accurate than the other algorithms, having 82.99% accuracy.

3.2.2. Anomaly Detection Using DL Algorithms

In the SDN environment, many approaches have been implemented for anomaly detection to secure the OpenFlow network. In [59], the author designed a programmable router for a home network by using the programmability feature of the SDN network to provide an ideal location and platform for detecting malicious behavior in SOHO (small office/home office). The four most popular SDN-based anomaly detection methods include NETAD, maximum entropy detector, rate-limiting, and TRW-CB algorithms were implemented in NOX and the OpenFlow

compliant switch of SDN. Experimental results of these algorithms showed that, in detecting malicious activities, these algorithms had more accurate results in SOHO networks compared with ISP. Without introducing any new performances overhead, these anomaly detection algorithm work at line rates for the SOHO network.

In [60], an anomaly-based detection system based on flow was proposed, using a gravitational search algorithm and multi-layer perceptron. Experimental results showed that this proposed system gave a high accuracy ratio in classifying malicious and benign flows. In [61], an SDN-based NID system was proposed by using SVM. Experimental results showed that the positive alarm rate was very high, with a lower false alarm rate. The traffic system was trained with malicious network traffic, rather than with normal data.

In [62], the authors proposed an anomaly detection system based on DL algorithms in which flow for anomaly detection and OpenFlow was combined to reduce overhead processing. In this proposed method, the false positive rate was high in detecting attacks because network traffic flow was used for implementation.

The network traffic of social multimedia is continuously increasing due to increases in usage and continuous development of multimedia services and applications. The secure transmission of data requires a network that includes features of quality of service, quality of information, scalability, and reliability. In this context, SDN is a significant network, but energy-aware networking and runtime security affects its capability; thus, to increase SDN reliability, a SDN-based anomaly detection system was proposed in [1]. By using DL approaches in the context of social multi-media, this system was used to detect any suspicious flow in network traffic. This proposed system consisted of two modules. The first was an anomaly-detection module based on RBM and gradient descent-based SVM for the detection of any suspicious behavior. The second module was end-to-end data delivery to satisfy SDN's quality of service requirements. For the performance evaluation of this proposed scheme, both benchmark and real-time datasets were used. Experimental evaluation showed that this proposed scheme was very efficient and effective in effective in data delivery and anomaly detection for social multimedia.

Some studies used deep learning for general anomaly detection. In the paper [63], the authors presented DL-based IDS for SDN environments. IDS was implemented as a component of the control plane in both studies, rather than using it as an application. The location allowed for direct interaction to protect the controller. Moving target defense and IDS was the aim of the authors [63]. To get data from training, a simulated network was generated by the authors (of about 40,000 packets). A neuroevolutionary model was presented as a light-weight detector for the architecture, by which real-time operation was allowed. Two different detectors, DDoS and worm, were developed to achieve it, with each detector identifying one type of attack. "Neuroevolutionary of Augmenting Topologies (NEAT)" was used by researchers to combine the two detectors. NEAT is a method related to neuro-evolution with a crossover background.

The general environment of SDN was presented by authors in [4] with unsupervised learning. The auto encoder was used in the approach; the encoder and decoder were the two phases of the auto encoder, used to identify reconstruction error and minimize it for each test sample. TensorFlow was the development library, but the used dataset was not clear.

3.2.3. Specific Circumstances of Network

Some studies also investigated specific circumstances of network. The implementation of IDS based on DL in optimal SDN was presented in [64]. Attacks were reviewed in the control plane in [65], and then they were classified into leakage of data, modification of data, unauthorized access, misuse of security policy, and denial of service. Anomaly detection considered features about optimal links, as the process was based on optical networks. These included usage of average bandwidth, destination nodes, formats of modulation, frequent source, and average length of route. Light-path creation, deletion, and modification were some attacks included in related networks. Point-anomaly-based methods were the first detection methods, with a point being used to represent a data instance. Probability was calculated by a user-created algorithm. A sequence-anomaly based method was the second detection method, where the occurrence of anomalies was in a sequence and a cumulative sum approach was used. NSFNET topology was used by the researchers to test an owned dataset, and the results of the detection method showed 85% accuracy.

Numerous compromised nodes were included in the attack by which synchronized traffic with low intensity was generated to disconnect hosts and links from any network. There is a growing reliance on digital measurements in the monitoring and managing of electrical power networks. Recently, wide-area monitoring systems (WAMS) have been established to enhance the situational awareness of complex networks and, by extension, their transmission efficiency [66].

Coordinated attacks are classified using three DL techniques including convolutional neural networks, artificial neural networks, and LSTM networks. In [65], a testbed was created by authors for the generation of their dataset in Mininet [67] with increased traffic. After that, training and testing of the model was performed using this dataset. The results showed that when there was an increase in the speed of the vehicles, the performance was reduced, as well as efficiency. The training time of each algorithm was 100 s. The short time allowed the system to re-train as necessary.

References

1. Mehdi, S.A.; Khalid, J.; Khayam, S.A. Revisiting Traffic Anomaly Detection Using Software Defined Networking. In International Workshop on Recent Advances in Intrusion Detection; Springer: Berlin/Heidelberg, Germany, 2011; pp. 161–180.
2. Garcia-Teodoro, P.; Diaz-Verdejo, J.; Maciá-Fernández, G.; Vázquez, E. Anomaly-Based Network Intrusion Detection: Techniques, Systems and Challenges. Comput. Secur. 2009, 28, 18–28.
3. Ahmed, M.E.; Kim, H.; Park, M. Mitigating DNS Query-Based DDoS Attacks with Machine Learning on Software-Defined Networking. In Proceedings of the MILCOM 2017–2017 IEEE Military Communications Conference (MILCOM), Baltimore, MD, USA, 23–25 October 2017; pp. 11–16.

4. Dawoud, A.; Shahristani, S.; Raun, C. Deep Learning and Software-Defined Networks: Towards Secure IoT Architecture. Internet Things 2018, 3, 82–89.
5. Injadat, M.; Moubayed, A.; Nassif, A.B.; Shami, A. Systematic ensemble model selection approach for educational data mining. Knowl. -Based Syst. 2020, 200, 105992.
6. Sultana, N.; Chilamkurti, N.; Peng, W.; Alhadad, R. Survey on SDN Based Network Intrusion Detection System Using Machine Learning Approaches. Peer-Peer Netw. Appl. 2019, 12, 493–501.
7. Singh, D.; Ng, B.; Lai, Y.-C.; Lin, Y.-D.; Seah, W.K. Modelling Software-Defined Networking: Software and Hardware Switches. J. Netw. Comput. Appl. 2018, 122, 24–36.
8. Krongbaramee, P.; Somchit, Y. Implementation of SDN Stateful Firewall on Data Plane Using Open VSwitch. In Proceedings of the 2018 15th International Joint Conference on Computer Science and Software Engineering (JCSSE), Nakhonpathom, Thailand, 11–13 July 2018; pp. 1–5.
9. Lockwood, J.W.; McKeown, N.; Watson, G.; Gibb, G.; Hartke, P.; Naous, J.; Raghuraman, R.; Luo, J. NetFPGA–An Open Platform for Gigabit-Rate Network Switching and Routing. In Proceedings of the 2007 IEEE International Conference on Microelectronic Systems Education (MSE'07), San Diego, CA, USA, 3–4 June 2007; pp. 160–161.
10. Lu, G.; Guo, C.; Li, Y.; Zhou, Z.; Yuan, T.; Wu, H.; Xiong, Y.; Gao, R.; Zhang, Y. ServerSwitch: A Programmable and High Performance Platform for Data Center Networks. In Proceedings of the 8th USENIX Symposium on Networked Systems Design and Implementation (NSDI 11), Boston, MA, USA, 30 March–1 April 2011.
11. Anwer, M.B.; Motiwala, M.; Tariq, M.B.; Feamster, N. Switchblade: A Platform for Rapid Deployment of Network Protocols on Programmable Hardware. In Proceedings of the ACM SIGCOMM 2010 Conference, New Delhi, India, 30 August–3 September 2010; pp. 183–194.
12. Medved, J.; Varga, R.; Tkacik, A.; Gray, K. Opendaylight: Towards a Model-Driven Sdn Controller Architecture. In Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, Sydney, NSW, Australia, 19 June 2014; pp. 1–6.
13. Itoh, T.; Sakai, M.; Okada, M. Floodlight. Google Patents Application CA 139471, 12 October 2011.
14. Gude, N.; Koponen, T.; Pettit, J.; Pfaff, B.; Casado, M.; McKeown, N.; Shenker, S. NOX: Towards an Operating System for Networks. ACM SIGCOMM Comput. Commun. Rev. 2008, 38, 105–110.
15. Tootoonchian, A.; Ganjali, Y. Hyperflow: A Distributed Control Plane for Openflow. In Proceedings of the 2010 Internet Network Management Conference on Research on Enterprise Networking, San Jose, CA, USA, 27 April 2010; Volume 3, pp. 10–5555.

16. Prakash, A.; Priyadarshini, R. An Intelligent Software Defined Network Controller for Preventing Distributed Denial of Service Attack. In Proceedings of the 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 20–21 April 2018; pp. 585–589.
17. Hu, F.; Hao, Q.; Bao, K. A Survey on Software-Defined Network and Openflow: From Concept to Implementation. IEEE Commun. Surv. Tutor. 2014, 16, 2181–2206.
18. Srivastava, R.; Richhariya, V. Survey of Current Network Intrusion Detection Techniques. J. Inf. Eng. Appl. 2013, 3, 27–33.
19. Kumari, K.; Prasad, A.; Prasad, K. Dielectric, Impedance/Modulus and Conductivity Studies on 0.94 Ba0. 06TiO3, (0.16 ≤ x ≤ 0.20) Lead-Free Ceramics. Am. J. Mater. Sci 2016, 6, 1–18.
20. Wu, H.; Schwab, S.; Peckham, R.L. Signature Based Network Intrusion Detection System and Method. Google Patents Application US10/092,179, 9 September 2008.
21. Yulianto, A.; Sukarno, P.; Suwastika, N.A. Improving Adaboost-Based Intrusion Detection System (IDS) Performance on CIC IDS 2017 Dataset. J. Phys. Conf. Ser. 2019, 1192, 012018.
22. Marotta, A.; Carrozza, G.; Avallone, S.; Manetti, V. An OpenFlow-Based Architecture for IaaS Security. In Proceedings of the ATACCS ‘13: International Conference on Application and Theory of Automation in Command and Control Systems, Naples, Italy, 28–30 May 2013; pp. 118–121.
23. Yasrebi, P.; Monfared, S.; Bannazadeh, H.; Leon-Garcia, A. Security Function Virtualization in Software Defined Infrastructure. In Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, ON, Canada, 11–15 May 2015; pp. 778–781.
24. Carvalho, L.F.; Abrão, T.; Leonardo; Lemes, M. An Ecosystem for Anomaly Detection and Mitigation in Software-Defined Networking. Expert Syst. Appl. 2018, 104, 121–133.
25. Chen, C.; Gong, Y.; Tian, Y. Semi-Supervised Learning Methods for Network Intrusion Detection. In Proceedings of the 2008 IEEE International Conference on Systems, Man and Cybernetics, Singapore, 12–15 October 2008; pp. 2603–2608.
26. Zwane, S.; Tarwireyi, P.; Adigun, M. A Flow-Based IDS for SDN-Enabled Tactical Networks. In Proceedings of the 2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC), Vanderbijlpark, South Africa, 21–22 November 2019; pp. 1–6.
27. Herrera, A.; Camargo, J.E. A Survey on Machine Learning Applications for Software Defined Network Security. In Proceedings of the International Conference on Applied Cryptography and Network Security, Bogota, Colombia, 5–7 June 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 70–93.
28. Tang, T.A.; Mhamdi, L.; McLernon, D.; Raza, A.; Ghogho, M. Deep Learning Approach for Network Intrusion Detection in Software Defined Networking. In Proceedings of the 2016

International Conference on Wireless Networks and Mobile Communications (WINCOM), Fez, Morocco, 26–29 October 2016; pp. 258–263.

29. Latah, M.; Toker, L. Towards an Efficient Anomaly-Based Intrusion Detection for Software-Defined Networks. IET Netw. 2018, 7, 453–459.
30. Prasath, M.K.; Perumal, B. A Meta-Heuristic Bayesian Network Classification for Intrusion Detection. Int. J. Netw. Manag. 2019, 29, e2047.
31. Kannadiga, P.; Zulkernine, M. DIDMA: A Distributed Intrusion Detection System Using Mobile Agents. In Proceedings of the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Network, Towson, MD, USA, 23–25 May 2005; pp. 238–245.
32. Wang, B.; Sun, Y.; Yuan, C.; Xu, X. LESLA: A Smart Solution for SDN-Enabled MMTC E-Health Monitoring System. In Proceedings of the 8th ACM MobiHoc 2018 Workshop on Pervasive Wireless Healthcare Workshop, Los Angeles, CA, USA, 26–25 June 2018; pp. 1–6.
33. Ashraf, S.; Shawon, M.H.; Khalid, H.M.; Muyeen, S.M. Denial-of-Service Attack on IEC 61850-Based Substation Automation System: A Crucial Cyber Threat towards Smart Substation Pathways. Sensors 2021, 21, 6415.
34. Ashraf, J.; Latif, S. Handling Intrusion and DDoS Attacks in Software Defined Networks Using Machine Learning Techniques. In Proceedings of the 2014 National Software Engineering Conference, Rawalpindi, Pakistan, 11–12 November 2014; pp. 55–60.
35. Kokila, R.; Thamarai, S.S.; Govindarajan, K. DDoS Detection and Analysis in SDN-Based Environment Using Support Vector Machine Classifier. In Proceeding of the 2014 Sixth International Conference on Advanced Computing (ICoAC), Chennai, India, 17–19 December 2014; pp. 205–210.
36. Wang, P.; Chao, K.-M.; Lin, H.-C.; Lin, W.-H.; Lo, C.-C. An Efficient Flow Control Approach for SDN-Based Network Threat Detection and Migration Using Support Vector Machine. In Proceedings of the 2016 IEEE 13th International Conference on e-Business Engineering (ICEBE), Macau, China, 4–6 November 2016; pp. 56–63.
37. Shiravi, A.; Shiravi, H.; Tavallaee, M.; Ghorbani, A.A. Toward Developing a Systematic Approach to Generate Benchmark Datasets for Intrusion Detection. Comput. Secur. 2012, 31, 357–374.
38. Gangadhar, S.; Sterbenz, J.P. Machine Learning Aided Traffic Tolerance to Improve Resilience for Software Defined Networks. In Proceedings of the 2017 9th International Workshop on Resilient Networks Design and Modeling (RNDM), Alghero, Italy, 4–6 September 2017; pp. 1–7.
39. Neupane, R.L.; Neely, T.; Chettri, N.; Vassell, M.; Zhang, Y.; Calyam, P.; Durairajan, R. Dolus: Cyber Defense Using Pretense against DDoS Attacks in Cloud Platforms. In Proceedings of the

19th International Conference on Distributed Computing and Networking, Varanasi, India, 4–7 January 2018; pp. 1–10.

40. Nanda, S.; Zafari, F.; DeCusatis, C.; Wedaa, E.; Yang, B. Predicting Network Attack Patterns in SDN Using Machine Learning Approach. In Proceedings of the 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Palo Alto, CA, USA, 7–10 November 2016; pp. 167–172.
41. Anderson, S.; Wickboldt, J.A.; Granville, L.Z.; Schaeffer-Filho, A. ATLANTIC: A Framework for Anomaly Traffic Detection, Classification, and Mitigation in SDN. In Proceedings of the NOMS 2016—2016 IEEE/IFIP Network Operations and Management Symposium, Istanbul, Turkey, 25–29 April 2016; pp. 27–35.
42. Kreutz, D.; Ramos, F.M.; Verissimo, P. Towards Secure and Dependable Software-Defined Networks. In Proceedings of the HotSDN '13: Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, Hong Kong China, 16 August 2013; pp. 55–60.
43. Akhunzada, A.; Gani, A.; Anuar, N.B.; Abdelaziz, A.; Khan, M.K.; Hayat, A.; Khan, S.U. Secure and Dependable Software Defined Networks. J. Netw. Comput. Appl. 2016, 61, 199–221.
44. Wang, Y.; Cai, W.; Wei, P. A Deep Learning Approach for Detecting Malicious JavaScript Code. Secur. Commun. Netw. 2016, 9, 1520–1534.
45. Sommer, R.; Paxson, V. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. In Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, USA, 16–19 May 2010; pp. 305–316.
46. Petroulakis, N.E.; Spanoudakis, G.; Askoxylakis, I.G. Patterns for the Design of Secure and Dependable Software Defined Networks. Comput. Netw. 2016, 109, 39–49.
47. Prete, L.R.; Shinoda, A.A.; Schweitzer, C.M.; Santos. Simulation in an SDN Network Scenario Using the POX Controller. In Proceedings of the 2014 IEEE Colombian Conference on Communications and Computing (COLCOM), Bogota, Colombia, 4–6 June 2014; pp. 1–6.
48. Wehrle, K.; Günes, M.; Gross, J. Modeling and Tools for Network Simulation; Springer Science & Business Media: Berlin, Germany, 2010.
49. Fontes, R.R.; Afzal, S.; Brito, S.H.; Santos, M.A.; Rothenberg, C.E. Mininet-WiFi: Emulating Software-Defined Wireless Networks. In Proceedings of the 2015 11th International Conference on Network and Service Management (CNSM), Barcelona, Spain, 9–13 November 2015; pp. 384–389.
50. Hande, Y.; Muddana, A. A Survey on Intrusion Detection System for Software Defined Networks (SDN); IGI Global: Hershey, PA, USA, 2021; pp. 467–489.

51. Elsayed, M.S.; Le-Khac, N.-A.; Dev, S.; Jurcut, A.D. Ddosnet: A Deep-Learning Model for Detecting Network Attacks. In Proceedings of the 2020 IEEE 21st International Symposium on “A World of Wireless, Mobile and Multimedia Networks” (WoWMoM), Cork, Ireland, 31 August–3 September 2020; pp. 391–396.
52. Zargar, S.T.; Joshi, J.; Tipper, D. A Survey of Defense Mechanisms against Distributed Denial of Service (DDoS) Flooding Attacks. IEEE Commun. Surv. Tutor. 2013, 15, 2046–2069.
53. Kolias, C.; Kambourakis, G.; Stavrou, A.; Voas, J. DDoS in the IoT: Mirai and Other Botnets. Computer 2017, 50, 80–84.
54. Su, A.-J.; Choffnes, D.R.; Kuzmanovic, A.; Bustamante, F.E. Drafting behind Akamai (Travelocity-Based Detouring). ACM SIGCOMM Comput. Commun. Rev. 2006, 36, 435–446.
55. Khalid, H.M.; Muyeen, S.M.; Peng, J.C.-H. Cyber-Attacks in a Looped Energy-Water Nexus: An Inoculated Sub-Observer-Based Approach. IEEE Syst. J. 2020, 14, 2054–2065.
56. Giotis, K.; Argyropoulos, C.; Androulidakis, G.; Kalogeras, D.; Maglaris, V. Combining OpenFlow and SFlow for an Effective and Scalable Anomaly Detection and Mitigation Mechanism on SDN Environments. Comput. Netw. 2014, 62, 122–136.
57. Lim, S.; Ha, J.; Kim, H.; Kim, Y.; Yang, S. A SDN-Oriented DDoS Blocking Scheme for Botnet-Based Attacks. In Proceedings of the 2014 Sixth International Conference on Ubiquitous and Future Networks (ICUFN), Shanghai, China, 8–11 July 2014; pp. 63–68.
58. Liu, Z.; He, Y.; Wang, W.; Zhang, B. DDoS Attack Detection Scheme Based on Entropy and PSO-BP Neural Network in SDN. China Commun. 2019, 16, 144–155.
59. Winter, P.; Hermann, E.; Zeilinger, M. Inductive Intrusion Detection in Flow-Based Network Data Using One-Class Support Vector Machines. In Proceedings of the 2011 4th IFIP International Conference on New Technologies, Mobility and Security, Paris, France, 7–10 February 2011; pp. 1–5.
60. Trung, V.; Huong, T.T.; Tuyen, V.; Duc, D.M.; Thanh, N.H.; Marshall, A. A Multi-Criteria-Based DDoS-Attack Prevention Solution Using Software Defined Networking. In Proceedings of the 2015 International Conference on Advanced Technologies for Communications (ATC), Ho Chi Minh City, Vietnam, 14–16 October 2015; pp. 308–313.
61. Jadidi, Z.; Muthukkumarasamy, V.; Sithirasenan, E.; Sheikhan, M. Flow-Based Anomaly Detection Using Neural Network Optimized with GSA Algorithm. In Proceedings of the 33rd International Conference on Distributed Computing System, Philadelphia, PA, USA, 8–11 July 2013; pp. 76–81.
62. Niyaz, Q.; Sun, W.; Javaid, A.Y. A Deep Learning Based DDoS Detection System in Software-Defined Networking (SDN). arXiv 2016, arXiv:1611.07400.

63. Dawoud, A.; Shahristani, S.; Raun, C. A Deep Learning Framework to Enhance Software Defined Networks Security. In Proceedings of the 32nd IEEE International Conference on Advanced Information Networking and Applications Workshops: IEEE WAINA 2018, Krakow, Poland, 16–18 May 2018; pp. 709–714.
64. Smith, R.J.; Zincir-Heywood, A.N.; Heywood, M.I.; Jacobs, J.T. Initiating a Moving Target Network Defense with a Real-Time Neuro-Evolutionary Detector. In Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion, Denver, CO, USA, 20–24 July 2016; pp. 1095–1102.
65. Narayanadoss, A.R.; Truong-Huu, T.; Mohan, P.M.; Gurusamy, M. Crossfire Attack Detection Using Deep Learning in Software Defined ITS Networks. Proceeding of the 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring), Kuala Lumpur, Malaysia, 28 April–1 May 2019; pp. 1–6.
66. Khalid, H.M.; Peng, J.C.-H. A Bayesian Algorithm to Enhance the Resilience of WAMS Applications Against Cyber Attacks. IEEE Trans. Smart Grid 2016, 7, 2026–2037.
67. Team, M. Mininet an Instant Virtual Network on Your Laptop (or Other PC). 2012. Available online: https://ic.unicamp.br (accessed on 26 August 2022).