ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940
www.ijsem.org
Vol 14, Issuse.3 Aug 2023

NETWORK THREAT DETECTION USING MACHINE/DEEP LEARNING IN SDN-BASED PLATFORMS: A COMPREHENSIVE ANALYSIS OF STATE-OF-THE-ART SOLUTIONS, DISCUSSION, CHALLENGES, AND FUTURE RESEARCH DIRECTION

THEINMALAR GOKUL*, DR RAGHUNATH REDDY**

ABSTRACT: 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. Recently 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. In the first part of this survey paper, we offer an introduction to the NIDS theory, as well as recent research that has been conducted on the topic. After that, we conduct a thorough analysis of the most recent ML- and DL-based NIDS approaches to ensure reliable identification of potential security risks. Finally, we focus on the opportunities and difficulties that lie ahead for future research on SDN-based ML and DL for NIDS.

Keywords: software defined network; intrusion detection systems; machine learning; deep learning; security attacks

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, denialof-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.

PGSCHOLAR*, PROFESSOR**
DEPARTMENT OF CSE, CMR ENGINEERING COLLEGE, 1, MEDCHAL RD, MEDCHAL, KANDLAKOYA, SEETHARIGUDA, TELANGANA 501401

ISSN2454-9940
www.ijsem.org
Vol 14, Issuse.3 Aug 2023

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. The basic purpose of this review paper is to comprehensively review the current advancements and trends in ML/DL-based NIDS systems and, more significantly, to provide an overview of the work on SDN-based NIDS systems using ML/DL approaches. This paper covers the area of knowledge for people with basic to moderate knowledge related to ML and DL for network threat detection in SDN. The motivation is to provide an overall picture of the existing research outcomes in this area. In addition, we aim to identify future research directions that may be useful for new researchers who are interested in this field of study. We analyzed the scientific research carried out on network threat detection (NTD) in SDN, based on ML and DL mechanisms. We covered the main and sub-parts of the NTD paradigm to efficiently cover threat issues and their protection using ML and DL approaches to avoid adversary attacks and protect sensitive information during storage and transmission on a public network. There were various review papers covering different aspects of this domain, leveraging ML/DL approaches [5–9]. Much research work has been done on NIDS using ML and DL approaches, but there are few studies on SDN-based NIDS and very few on DL-based NIDS systems in SDN. We believe our study is different from existing studies

ISSN2454-9940
www.ijsem.org
Vol 14, Issuse.3 Aug 2023

for the following reasons (and thus, are the main contributions of our work):

• First, we conducted a comprehensive review on ML/DL-based network intrusion detection systems;

• Second, we reviewed each study on SDN-based NID systems using ML and DL algorithms; • We also explored recent advancements and trends in ML/DL approaches for NIDS, followed by the NIDS system leveraging SDN using ML/DL approaches, and research issues in NID systems using ML/DL approaches

LITERATURE REVIEW

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 [14]. 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 flowbased 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 [7]. The general architecture of the SDN is shown in Figure 1. It shows the three planes of SDN, named the application plane, control plane, and data plane, which are discussed in the following sections, respectively

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 [15], Indigo, and Open vSwitch [16]. 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

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN2454-9940
www.ijsem.org
Vol 14, Issuse.3 Aug 2023

[17]). Two open network hardware-based physical switches are ServerSwitch [18] and SwitchBlade [19]. 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.

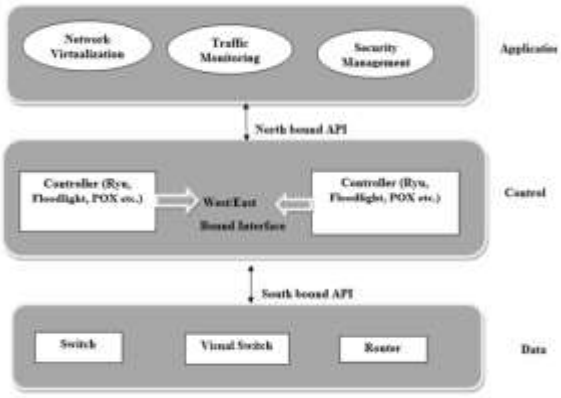

Fig: General architecture of SDN

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 [20], Ryu, Flood-light [21], POX, and NOX [22] 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

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.

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

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 [6]. 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. A summary of IDS deployment environments and implemented techniques of detection is shown in Figure 2. As Figure 2 describes, various techniques and methods can be used to implement intrusion detection systems, which are further divided into following groups: ML-based methods, data mining methods, and statistical techniques [8]. 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

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940
www.ijsem.org
Vol 14, Issuse.3 Aug 2023

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

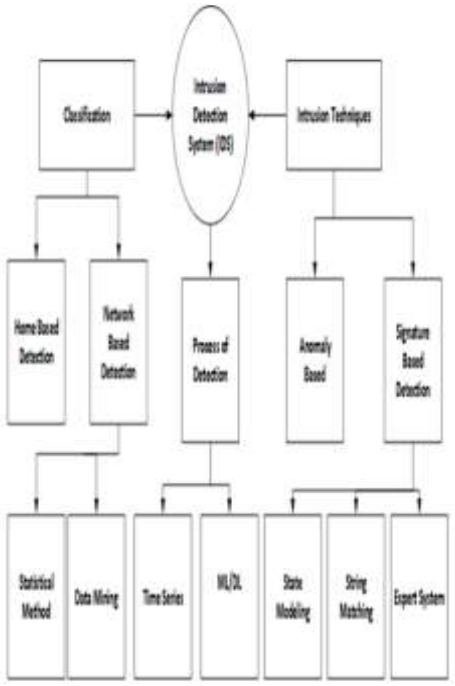

Fig: Overview of IDS.

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

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 [27]. In [28], an algorithm was designed for an anomalybased 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

- Hodo *et al.* [3] adopt variousMLalgorithms to detect cyberattacks against an emulated industrial environment using the IEC 60 870-5-104 protocol. To this end, the authors use a dataset consisting of 1) replay attacks, 2) DoS attacks, and (c) address resolution protocol spoofing attacks. Thus, they evaluate the classification performance of various ML classifiers, including Random Forest, OneR, J48, IBk, and Naive Bayes. According to the evaluation results, J48 achieves the best performance.
- Yang *et al.* [4] create Snort-compliant signature and specification rules to detect IEC 60 870-5-104-related cyberattacks. The difference between the signature and specification rules lies in the fact that the former category defines malicious patterns, while the second determines the normal behavior. The same authors in [7] introduce a

ISSN2454-9940
www.ijsem.org
Vol 14, Issuse.3 Aug 2023

specification-based intrusion detection system (IDS) capable of recognizing IEC 60 870-5-104 anomalies. The proposed IDS relies on a detection state machine, which relies on finite state machines. The experimental results confirm the efficiency of the proposed IDS.

- Lin [14] introduces an SDN-based in-network honeypot, which can mitigate the impact of a cyberattack by 1) isolating the cyberattacker and 2) spoofing the network communication, thereby establishing a connection with a cyberattacker via nonexistent nodes, called phantom nodes. This connection allows the defender to mislead the cyberattacker and gather useful information. Initially, the SDN controller (SDN-C) quarantines themalicious nodes by corrupting their communicationwith any legitimate node. Next, the SDN-C uses spoofed IP addresses that communicate with the cyberattacker by adapting appropriately the network packets' content at the network and application layers. To this end, statistic and physical models are utilized, respectively

IMPLEMENTATION

Several papers have investigated the cybersecurity issues in the healthcare sector. Some of them are listed in [1], [9]–[13]. In particular, in [1], Yaqoob *et al.* investigate the vulnerabilities of the smart medical devices and propose appropriate countermeasures. In [9], Chenthara *et al.* discuss the cybersecurity and privacy challenges of the e-health solutions in cloud-computing environments. Similarly, Wolker-Roberts *et al.* [10] discuss relevant countermeasures against internal threats in healthcare CIs. Vijayakumar *et al.* [11] provide an anonymous authentication framework for wireless body area networks. Finally, Sun *et al.* [12] provide a detailed survey about the IoMT security and privacy issues. Next, we elaborate on some similar works regarding 1) IEC 60 870-5-104 threat modeling, 2) detecting intrusions against IEC 60 870-5- 104, and 3) mitigating or even preventing cyberattacks through SDN. In [5], the authors conduct an abstract threat analysis of the IEC 60 870-5-104 industrial systems. Based on a colored Petri net (CPN) analysis, two cyberattack categories are specified: 1) physical attacks and 2) cyberattacks. The first category denotes those activities performed by an attacker having physical access to the target system. On the other side, the cyberattacks refer to those that exploit the IEC 60 870-5-104 vulnerabilities. In particular, based on the authors, the second category includes the following four kinds:

1) unauthorized access;
2) man-in-the-middle (MITM);
3) DoS;
4) traffic analysis.

Each of the aforementioned cyberattacks is assigned to the CPN transitions. Next, the authors emulate the four IEC 60 870- 5-104 cyberattacks and quantify their risk based on the Alien- Vault OSSIM risk model.

Hodo *et al.* [3] adopt variousMLalgorithms to detect cyberattacks against an emulated industrial environment using the IEC 60 870-5-104 protocol. To this end, the authors use a dataset consisting of 1) replay attacks, 2) DoS attacks, and (c) address resolution protocol spoofing attacks. Thus, they evaluate the classification performance of various ML classifiers, including Random Forest, OneR, J48, IBk, and Naive Bayes.

According to the evaluation results, J48 achieves the best performance. Yang *et al.* [4] create Snort-compliant signature and specification rules to detect IEC 60 870-5-104-related cyberattacks. The difference between the signature and specification

ISSN2454-9940
www.ijsem.org
Vol 14, Issuse.3 Aug 2023

rules lies in the fact that the former category defines malicious patterns, while the second determines the normal behavior. The same authors in [7] introduce a specification-based intrusion detection system (IDS) capable of recognizing IEC 60 870-5-104 anomalies. The proposed IDS relies on a detection state machine, which relies on finite state machines. The experimental results confirm the efficiency of the proposed IDS.

- The system is not implemented SDN-BASED MITIGATION: PROBLEM FORMULATION AND METHODOLOGY.
- The system is not implemented enough method for testing and training for large datasets.

The proposed IEC 60 870-5-104 threat modeling combines both ADT and CVSS that determine the cyberattack paths and their risks, respectively. In particular, an ADT [16] comprises two antagonistic nodes: 1) attacking nodes and 2) defending nodes. The attacking nodes describe the goal and the actions that a cyberattacker may adopt in order to compromise the security of the target system. The defending nodes correspond to the defences that can be used by the defender in order to address or mitigate a cyberattack. Each node can have one or more children of the same type (i.e., attacking node or defending node), thus reflecting a refinement into specific subgoals and actions. If a node does not have any refinement (i.e., children of the same type), then it constitutes a nonrefined node, which indicates a basic action. Moreover, a node can have children of the opposite type, thus defining a countermeasure. A refinement can be classified into two types: 1) conjunctive and 2) disjunctive. In the first case (i.e., conjunctive refinement), the goal of a refined node is achieved, whether all of its children accomplish their goals. Thus, a conjunctively refined node is characterized by an AND operator. On the other side, a disjunctively refined node is characterized by an OR operator, i.e., its goal is achieved if at least one of its children achieves its goal. On the other side, CVSS is an open vulnerability assessment framework, which quantifies the severity of each vulnerability or attack between 0 and 10 [17].

The proposed system implemented 1) detection performance and 2) mitigation performance which are enough operations on datasets. The proposed system developed notification and response module (NRM) for datasets prediction.

ALGORITHMS

DECISION TREE CLASSIFIERS

Decision tree classifiers are used successfully in many diverse areas. Their most important feature is the capability of capturing descriptive decision making knowledge from the supplied data. Decision tree can be generated from training sets. The procedure for such generation based on the set of objects (S), each belonging to one of the classes C1, C2, …, Ck is as follows:

Step 1. If all the objects in S belong to the same class, for example Ci, the decision tree for S consists of a leaf labeled with this class

Step 2. Otherwise, let T be some test with possible outcomes O1, O2,…, On. Each object in S has one outcome for T so the test partitions S into subsets S1, S2,… Sn where each object in Si has outcome Oi for T. T becomes the root of the decision tree and for each outcome Oi we build a subsidiary decision tree by invoking the same procedure recursively on the set Si.

GRADIENT BOOSTING

Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees.[1][2] When a

ISSN2454-9940
www.ijsem.org
Vol 14, Issuse.3 Aug 2023

decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest.A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

K-NEAREST NEIGHBORS (KNN)

- Simple, but a very powerful classification algorithm
- Classifies based on a similarity measure
- Non-parametric
- Lazy learning
- Does not "learn" until the test example is given
- Whenever we have a new data to classify, we find its K-nearest neighbors from the training data

Example

- Training dataset consists of k-closest examples in feature space
- Feature space means, space with categorization variables (non-metric variables)
- Learning based on instances, and thus also works lazily because instance close to the input vector for test or prediction may take time to occur in the training dataset

LOGISTIC REGRESSION CLASSIFIERS

Logistic regression analysis studies the association between a categorical dependent variable and a set of independent (explanatory) variables. The name *logistic regression* is used when the dependent variable has only two values, such as 0 and 1 or Yes and No. The name *multinomial logistic regression* is usually reserved for the case when the dependent variable has three or more unique values, such as Married, Single, Divorced, or Widowed. Although the type of data used for the dependent variable is different from that of multiple regression, the practical use of the procedure is similar.

Logistic regression competes with discriminant analysis as a method for analyzing categorical-response variables. Many statisticians feel that logistic regression is more versatile and better suited for modeling most situations than is discriminant analysis. This is because logistic regression does not assume that the independent variables are normally distributed, as discriminant analysis does.

This program computes binary logistic regression and multinomial logistic regression on both numeric and categorical independent variables. It reports on the regression equation as well as the goodness of fit, odds ratios, confidence limits, likelihood, and deviance. It performs a comprehensive residual analysis including diagnostic residual reports and plots. It can perform an independent variable subset selection search, looking for the best regression model with the fewest independent variables. It provides confidence intervals on predicted values and provides ROC curves to help determine the best cutoff point for classification. It allows you to validate your results by automatically classifying rows that are not used during the analysis.

NAÏVE BAYES The naive bayes approach is a supervised learning method which is based on a simplistic hypothesis: it assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature .

Yet, despite this, it appears robust and efficient. Its performance is comparable to other supervised learning techniques. Various reasons have been advanced in the literature. In this tutorial, we highlight an explanation based on the representation bias. The naive bayes classifier is a linear classifier, as well as linear discriminant analysis, logistic regression or linear SVM

ISSN2454-9940
www.ijsem.org
Vol 14, Issuse.3 Aug 2023

(support vector machine). The difference lies on the method of estimating the parameters of the classifier (the learning bias).

While the Naive Bayes classifier is widely used in the research world, it is not widespread among practitioners which want to obtain usable results. On the one hand, the researchers found especially it is very easy to program and implement it, its parameters are easy to estimate, learning is very fast even on very large databases, its accuracy is reasonably good in comparison to the other approaches. On the other hand, the final users do not obtain a model easy to interpret and deploy, they does not understand the interest of such a technique.

Thus, we introduce in a new presentation of the results of the learning process. The classifier is easier to understand, and its deployment is also made easier. In the first part of this tutorial, we present some theoretical aspects of the naive bayes classifier. Then, we implement the approach on a dataset with Tanagra. We compare the obtained results (the parameters of the model) to those obtained with other linear approaches such as the logistic regression, the linear discriminant analysis and the linear SVM. We note that the results are highly consistent. This largely explains the good performance of the method in comparison to others. In the second part, we use various tools on the same dataset (Weka 3.6.0, R 2.9.2, Knime 2.1.1, Orange 2.0b and RapidMiner 4.6.0). We try above all to understand the obtained results.

Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

The first algorithm for random decision forests was created in 1995 by Tin Kam Ho[1] using the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.

An extension of the algorithm was developed by Leo Breiman and Adele Cutler, who registered "Random Forests" as a trademark in 2006 (as of 2019, owned by Minitab, Inc.).The extension combines Breiman's "bagging" idea and random selection of features, introduced first by Ho[1] and later independently by Amit and Geman[13] in order to construct a collection of decision trees with controlled variance.

Random forests are frequently used as "blackbox" models in businesses, as they generate reasonable predictions across a wide range of data while requiring little configuration.

SVM

In classification tasks a discriminant machine learning technique aims at finding, based on an *independent and identically distributed* (*iid*) training dataset, a discriminant function that can correctly predict labels for newly acquired instances. Unlike generative machine learning approaches, which require computations of conditional probability distributions, a discriminant classification function takes a data point *x* and assigns it to one of the different classes that are a part of the classification task. Less powerful than generative approaches, which are mostly used when prediction involves outlier detection, discriminant approaches require fewer computational resources and less training data, especially

ISSN2454-9940
www.ijsem.org
Vol 14, Issuse.3 Aug 2023

for a multidimensional feature space and when only posterior probabilities are needed. From a geometric perspective, learning a classifier is equivalent to finding the equation for a multidimensional surface that best separates the different classes in the feature space.

SVM is a discriminant technique, and, because it solves the convex optimization problem analytically, it always returns the same optimal hyperplane parameter—in contrast to *genetic algorithms* (*GAs*) or *perceptrons*, both of which are widely used for classification in machine learning. For perceptrons, solutions are highly dependent on the initialization and termination criteria. For a specific kernel that transforms the data from the input space to the feature space, training returns uniquely defined SVM model parameters for a given training set, whereas the perceptron and GA classifier models are different each time training is initialized. The aim of GAs and perceptrons is only to minimize error during training, which will translate into several hyperplanes' meeting this requirement.

CONCLUSIONS AND FUTURE WORK

In this paper, various approaches based on the classical techniques of ML and DL were discussed in detail, by which attacks were detected in SDN. The paper also included explanation about the weaknesses of traditional methods based on ML and their poor performance. It was emphasized that vulnerabilities are detected and the network is monitored using approaches of ML/DL in a platform known as SDN. In SDN, specific methods are needed to define data by classification into frameworks and techniques to implement ML approaches. The need for updated datasets was identified, as the latest attacks are used to learn models. Based on this study, the usage of methodologies based on DL were shown to improve NIDS effectiveness and performance in terms of FAR reduction and accuracy of detection. DL approaches were used in about 80% of the proposed solutions with AE and DNN, and these algorithms were used frequently. It is worth noting that the high speed infrastructure and critical infrastructure of the network did not use any approaches where SDN-based NIDS was implemented. In this area, the future research scope should be to propose an efficient framework of NIDS using fewer complex DL algorithms for a more effective detection mechanism. This knowledge could be used to design an effective, lightweight, and innovative NIDS-based DL. Moreover, intruders could be effectively detected by using these within the network.

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. [**Google Scholar**]
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. [**Google Scholar**] [**CrossRef**]
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. [**Google Scholar**]
4. Dawoud, A.; Shahristani, S.; Raun, C. Deep Learning and Software-Defined Networks: Towards Secure IoT Architecture. *Internet Things* **2018**, *3*, 82–89. [**Google Scholar**] [**CrossRef**]

ISSN2454-9940
www.ijsem.org
Vol 14, Issuse.3 Aug 2023

5. 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. [Google Scholar]
6. 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. [Google Scholar] [CrossRef]
7. 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. [Google Scholar] [CrossRef]
8. Hodo, E.; Bellekens, X.; Hamilton, A.; Tachtatzis, C.; Atkinson, R. Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey. *arXiv* **2017**, arXiv:1701.02145. [Google Scholar]
9. Tiwari, S.; Pandita, V.; Sharma, S.; Dhande, V.; Bendale, S. Survey on Sdn Based Network Intrusion Detection System Using Machine Learning Framework. *IRJET* **2019**, *6*, 1017–1020. [Google Scholar]
10. Xie, J.; Richard, Y.F.; Huang, T.; Xie, R.; Liu, J.; Wang, C.; Liu, Y. A Survey of Machine Learning Techniques Applied to Software Defined Networking (SDN): Research Issues and Challenges. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 393–430. [Google Scholar] [CrossRef]
11. Chalapathy, R.; Chawla, S. Deep Learning for Anomaly Detection: A Survey. *arXiv* **2019**, arXiv:1901.03407. [Google Scholar]
12. Aldweesh, A.; Derhab, A.; Emam, A.Z. Deep Learning Approaches for Anomaly-Based Intrusion Detection Systems: A Survey, Taxonomy, and Open Issues. *Knowl.-Based Syst.* **2020**, *189*, 105124. [Google Scholar] [CrossRef]
13. Ahmad, Z.; Khan, S.; Shiang, W.; Abdullah, J.; Ahmad, F. Network Intrusion Detection System: A Systematic Study of Machine Learning and Deep Learning Approaches. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4150. [Google Scholar] [CrossRef]
14. Injadat, M.; Moubayed, A.; Nassif, A.B.; Shami, A. Systematic ensemble model selection approach for educational data mining. *Knowl. -Based Syst.* **2020**, *200*, 105992. [Google Scholar] [CrossRef]
15. Singh, D.; Ng, B.; Lai, Y.-C.; Lin, Y.-D.; Seah, W.K. Modelling Software-Defined Networking: Software and Hardware