Advances in Science and Technology Research Journal

Advances in Science and Technology Research Journal 2022, 16(3), 193–206
https://doi.org/10.12913/22998624/149934
ISSN 2299–8624, License CC-BY 4.0

Received: 2022.04.23
Accepted: 2022.05.14
Published: 2022.06.01

Network Intrusion Detection Using Machine Learning Techniques

Yasmeen S. Almutairi[1], Bader Alhazmi[1], Amr A. Munshi[1*]

[1] Computer Engineering Department, Umm Al-Qura University, Makkah 21961, Saudi Arabia

* Corresponding author's e-mail: aaamunshi@uqu.edu.sa

ABSTRACT

Intrusion detection systems (IDS) are essential for the protection of advanced communication networks. These systems were primarily designed to identify particular patterns, signatures, and rule violations. Machine Learning and Deep Learning approaches have been used in recent years in the field of network intrusion detection to provide promising alternatives. These approaches can discriminate between normal and anomalous patterns. In this paper, the NSL-KDD (Network Security Laboratory Knowledge Discovery and Data Mining) benchmark data set has been used to evaluate Network Intrusion Detection Systems (NIDS) by using different machine learning algorithms such as Support Vector Machine, J48, Random Forest, and Naïve Bytes with both binary and multiclass classification. The results of the application of those techniques are discussed in details and outperformed previous works.

Keywords: data mining; intrusion detection system; machine learning techniques.

INTRODUCTION

More and more devices are becoming internet-connected these days. According to Cisco, there will be 29.3 billion internet-connected gadgets by 2023 [1]. The Mirai botnet attack on Domain Name System provider affected the Internet-of-Things (IoT) devices, resulting in the unavailability of many major Internet companies such as Spotify, Twitter, and Netflix [2]. Intruders are able to easily defeat the firewall and its variants, for example, by exploiting a bogus source address. It has also failed to detect a large number of DoS and DDoS attacks [3]. To overcome the problems associated with traditional security methods, a new security mechanism called the Intrusion Detection System (IDS) has been created. IDS monitors inbound and outbound traffic for malicious activity. IDS can be classified based on where it is installed or the method it employs to identify anomalous actions [4].

IDS can be placed at terminals to protect them from being attacked, which is known as Host-based IDS (HIDS), or at the network's entry point to monitor incoming and outgoing packets for malicious content to defend the entire network, which is known as Network-based IDS (NIDS). As a result, a variety of methodologies, including evolutionary, information theory, statistical, and machine-learning techniques, have been employed to create a model that can detect abnormalities well.

This research looks into a variety of machine-learning techniques for evaluating intrusion detection systems by distinguishing attack patterns (signatures) or network traffic behavior. In order to achieve this, the IDSs techniques and a thorough literature review on state-of-the-art intrusion detection models, aiming at proposing a methodology to detect attack patterns and network traffic behaviors.

BACKGROUND

IDSs are security solutions that, like antivirus software, firewalls, and access control schemes, are designed to make information and communication systems more secure. IDS arose as a result of the inadequacy of traditional security methods.

The following subsections discuss the network security, firewalls and IDSs, respectively.

Network security

According to Cisco [5], network security involves any action that is tailored to ensure that there is usefulness and reliable integrity of the user's network and data. This activity incorporates both tangible and intangible innovations to computer systems. Accessing the network is usually under the control of active network security. It can detect and prevent a variety of threats from getting into or proliferating throughout the user's network at any given time. The majority of security threats are purposefully created by malicious people seeking a benefit, gaining publicity, or harming someone. Network security issues can be loosely classified into five interconnected areas, as noted by [6]:

1. Confidentiality: The contents of the transmitted communication should only be understood by the sender and the intended receiver because the message could be intercepted by eavesdroppers. Encryption is used to accomplish this.
2. Message integrity assures that the delivered message's content isn't tampered with, either intentionally or accidentally. Checksum and hash functions are used to accomplish this.
3. Verification: the party sending, and the one receiving the information, ought to have a way of verifying their identity. Each party should be able to verify the identity of the other.
4. Nonrepudiation deals with the possibility of someone denying sending a message or carrying out an action. It is achieved through digital signatures.
5. Operational security: this is a security process used to prevent important materials of a company or an institution from being accessed by unauthorized individuals.

Nearly all institutions, including banks and higher learning institutions, among others, possess or use a network that happens to be linked to the public Internet. At some point, the networks can easily be tampered with without the owner's consent.

Malicious people can introduce worms into the network's host, access the institution's confidential documents, change the organization's network configuration, and launch disk operating system attacks. For this reason, firewalls and IDSs are put into use to counter attacks that may arise against a company's network. Networks of companies or institutions are organized into two categories: internal networks and demilitarized zone (Fig. 1).

The internal network of the company or institution can only be accessed by the network administrators or the workers within the company. The demilitarized zone (DMZ) can be accessed by anyone. Having a demilitarized zone within any organization plays a very crucial role. It adds an extra layer of security to the company's internal network because the hosts that are the most susceptible to attacks are the ones that provide services to users who are not within the internal network, for instance, electronic mail, website, and domain name system servers.

Due to the high number of organizations that are facing attacks, the organizations are placed within a subnetwork to protect the rest of the network within the organization from

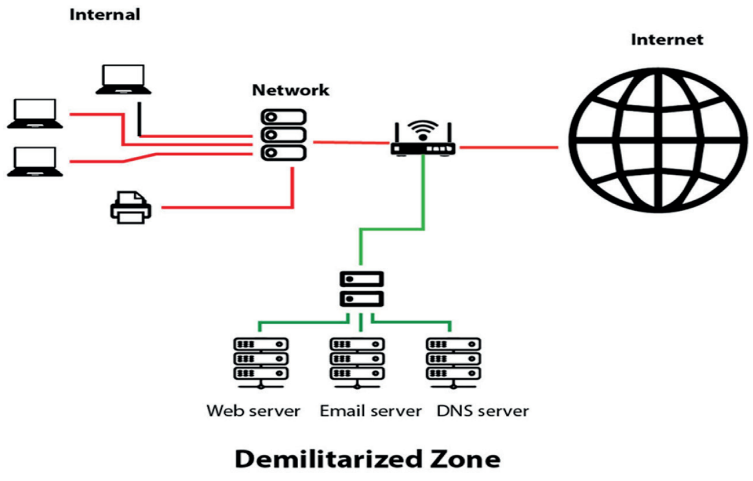

Fig. 1. Organization of network topology

receiving attacks. Only the information exposed in the DMZ within an organization can be accessed by an external host. The rest of the organization's network cannot be accessed by any means from an external host. Nevertheless, having a separation of the organization's network while not developing tactics that can control network traffic doesn't make any sense. Consequently, a common mechanism of security is the addition of a firewall.

Firewalls

A firewall is a feature that combines tangible and intangible computer systems to isolate an organization's network from the Internet, hence creating a situation where some packets will be allowed to pass while others will not. It functions as a filter of packets that are coming in and moving out of the organization's network. Only the packets that meet the criteria formulated by the organization's network firewall are allowed to flow normally and with ease. Those that don't comply with the criteria are blocked.

Firewall placement

Within the network of an organization, the firewall can have two common possible placements; at the gateway or between the internal network and the router (Fig. 2). To ensure proper network security, large institutions can use several firewall levels. Access to authorized traffic can be restricted by having the firewall within the company router in place. On the other hand, placing it in between the internal and the external networks make the demilitarized zone without any defense measure. Each placement is associated with various limitations.

Firewall categories

According to [6], there are only three categories of firewalls:

1. Traditional packet filters

In this category of firewalls, every datagram is examined separately based on specific rules, imposed by the administrator, to determine whether the datagram may legitimately be allowed to pass or be dropped. In this category, filtration of datagrams is mainly based on the following:

- The source address of an Internet packet, or the destination address.
- Protocol type in Internet protocol datagram field.
- The source of the transmission control protocol or user datagram protocol and the destination port.
- Transmission control protocol flag bits.
- The Internet control message protocol type of message.
- Various regulations for datagrams that leave and enter the network.

Based on how the policies of an organization are formulated, the network administrator knows how to install and do firewall configurations for the organization. For instance, if the company does not need any incoming transmission control protocol (TCP) synchronization sections, except those for its public Web server, it can consider blocking all in-coming transmission control protocol synchronization segments except those segments with destination port 80, and the destination internet protocol address corresponding to the Web Server. If the company's management does not want the internal network of the company to be mapped by an individual who is not within the company, it can block all Internet control message protocol

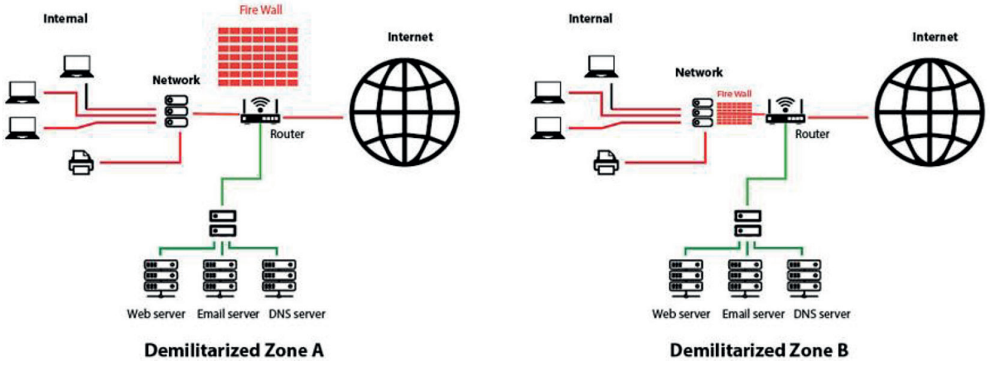

Fig. 2. Different firewall location

time-to-live expired messages that leave the network of the organization or company. Filtering can be done by combining addresses and port numbers. The major drawback of this firewall category is that there is no protection of addresses from datagrams in the case where the address sources have already been hoaxed. In addition, it is not easy to write rules that allow useful functionality and ensure that all traffic that is not needed is fully blocked [7].

2. Stateful packet filters

This category maps packets to links and uses the field of transmission control protocol or internet protocol (IP) header to ensure connections. By doing this it creates rules that, for instance, permit the access of external web servers to ensure packets are channeled to an internal host, but only after ensuring that the internal host has a connection with the external Web server. The major disadvantage is that, occasionally, this category can't function in circumstances where it can't forward traffic because the traffic exceeds the capacity of the connection table of the firewall.

3. Application gateway

This category involves a firewall examining the content of packets that is beyond the TCP header to know exactly what the application is doing. With this capability of the firewall, distinguishing between hypertext transfer protocol (HTTP) traffic that is used in peer-to-peer file sharing, and the one used for Web browsing can be done with ease. Web administrators can compose rules to safeguard an organization from peer-to-peer file sharing but allow Web browsing activities that are crucial and work to the advantage of the organization.

The company can ensure that the incoming and outgoing traffic is closely inspected. For instance, it can help in the prevention of information that is very sensitive to the organization/company from being sent through emails to an outside source. Like the other categories, the application gateway has its own limitations. They include that the company using this category needs a different application gateway for every application. This means that the company uses more resources to ensure that it is properly implemented. Also, there is always a penalty fee to be paid by the organization using this category because all the data are relayed through the gateway.

Firewall drawbacks

Various security issues that arise even though firewalls are usually configured properly. For instance, if a firewall is configured to filter packets, an intruder outside the firewall can use false sources of address to evade the process of filtering.

If someone within an organization wants to send secret information or sensitive information from the organization he or she is working for, without being noticed, it is very easy. He or she can encrypt the information he or she wants to send, or they can scan them and transform them into JPEG files, and send them via email without being noticed. The majority of attacks that a company or organization experiences usually come from outside attacks. However, even though the attacks from the inside of the company or organization are minimal, they are the ones that cause more damage to the company or the organization involved [8].

There are other types of attacks that firewalls can't deal with at all. The main aim of firewalls is to make sure that intruders cannot get into a company's network and obtain important information or sensitive data about the organization. However, some individuals only want to use their information technology expertise to bring the organizations down. Some of them feel that it is a significant achievement when they bring an organization down. They do this by sending data packets to the target organization until they ensure the organization goes down. These kinds of attacks are referred to as Denial of Service (DoS). These intruders use packets that do not have any source address so it is not easy to trace them. There is another type of intruder that uses very many computers and commands them to attack a certain organization's network at the same time. This kind of attack is referred to as a Distributed Denial of Service (DDoS) attack. It is not easy to create a defense against this type of attack. Firewalls cannot cope with these two types of attacks.

Intrusion detection systems

As we saw in the previous section, a packet filter (firewall) inspects packets such as ICMP, TCP, IP, and UDP header fields when determining whether to allow them past the firewall. However, Deep Packet Inspection (DPI) is required to detect many attack types, particularly those that the packet filter cannot detect. A device that not only analyzes the headers of all packets traveling

through it (unlike a packet filter) but also does deep packet inspections has a place in intrusion prevention. An Intrusion Prevention System acts when a device detects a suspect packet or a suspicious series of packets and drops them to prevent them from accessing the organization's network. Intrusion Detection Systems are used when a device can let packets pass by it on their way to the corporate network but sends an alarm to the network administrator or logs the packets. In this section, we'll look at intrusion detection in further depth. Intrusion Detection Systems are computer-based security and defense systems that monitor, identify, and analyze harmful activity on hosts or networks.

The purpose of an intrusion detection system is to ensure that the security of a computer system or network based on integrity, confidentiality, and availability is maintained. The Intrusion Detection System, upon detecting that an intrusion has occurred and that the firewall failed to mitigate or stop the attack or intrusion [8]. The firewall is the first protection against intrusion. At the same time, using the Intrusion Detection System is based upon the certainty that an attack will occur that the firewall cannot eliminate or mitigate. The Intrusion Detection System can be classified in different ways, based on the monitored platform or the technique they employ to identify anomalous activity (Fig. 3).

IDS types by monitored platform

1. Network-based IDS (NIDS)

This type of IDS is typically implemented at the network's entry point to protect all hosts, such as at the organization's border router or the internal network's or DMZ's entry point. It's in a different place. In addition, if an organization receives gigabits/s of traffic from the Internet, it may deploy one or more IDS monitors in its organizational network to balance the load. Each IDS monitor observes only a small part of the traffic of the organization since the IDS monitors are placed at different points throughout the network. This form of IDS is usually easy to install in a network and is deemed secure against attacks [8]. However, they have significant drawbacks, such as the difficulty of analyzing all packets from a large and overburdened network, particularly when only a few IDS units are implemented. Fig. 4 and 5 show that a network-based intrusion detection system can be deployed at the three disparate areas simultaneously.

2. Host-based IDS (HIDS)

HIDS is most commonly used on single hosts. They are installed on the host as software. This type of IDS ensures the host's security by monitoring only the host's inbound and outbound packets and alerting the user or administrator if suspicious activity is discovered. It protects against undetectable attacks by firewalls and network-based intrusion detection systems. Another advantage of HIDS over NIDS can be determined if the attacker succeeds or fails quickly [9].

3. Hybrid IDS:

The Hybrid Intrusion Detection Systems are installed between a Host Intrusion Detection System and a Network Intrusion Detection System. It is created by obtaining data or information from both the host and the network for analysis and then applying a methodology. Because these systems solve the disadvantages of both HIDS and NIDS, they will prove to be exceedingly efficient and successful in the future.

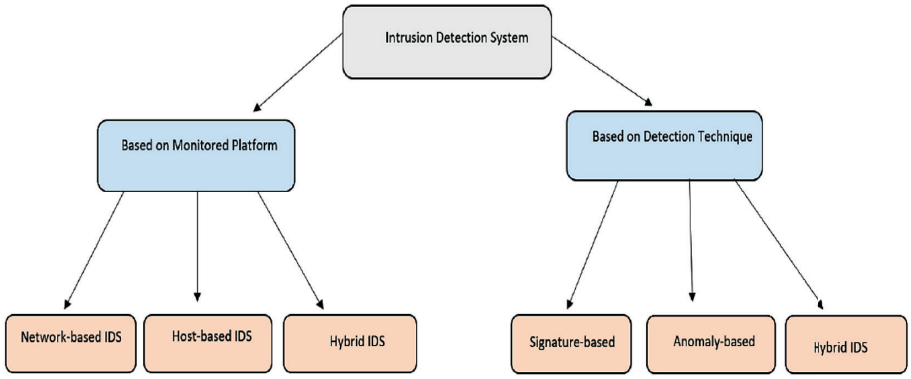

Fig. 3. Different firewall location

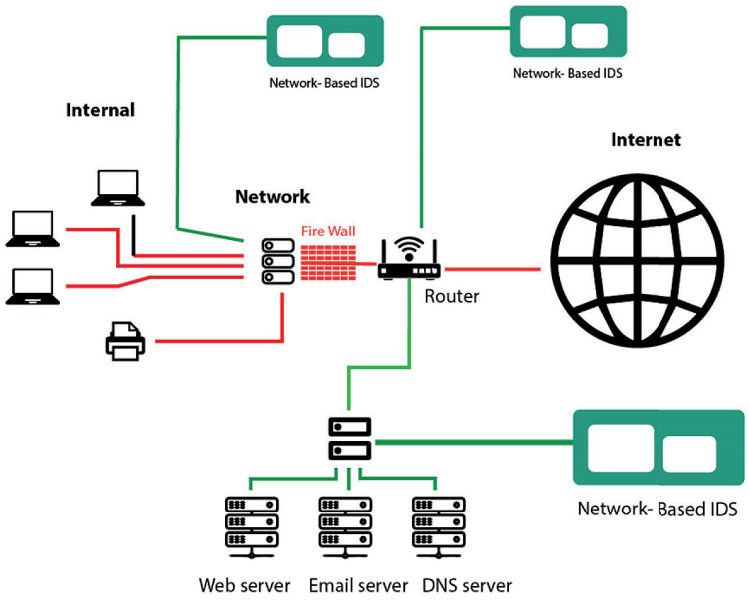

Fig. 4. NIDS sensor placement

Types of IDS based on the technique used in the detection

1. Misuse based – IDS based on signature

Signatures of attacks, for instance, file fingerprints and metadata (MD5 or SHA1 hash), are used to assess whether an attack is occurring or whether the activity under system observation is legitimate. The legitimacy test and attack assessment can compare signatures against prior assaults' signatures [10].

2. Anomaly based

Any network behavior that deviates from the standard network operating baseline is suspected of abnormality by anomaly-based detection systems. Under certain circumstances, this strategy is recommended over a signature-based system because it has a significantly better probability of detecting fresh (zero-day) attacks. Because each network has its baseline, attackers will find it difficult to carry out an undetected attack without creating an anomaly.

Despite the method's stated benefits, it is prone to false positives, as some of the identified changes are actual system updates. The two types of anomaly detection systems are the statistical and the knowledge-based. In contrast, the knowledge-based technique entails noting and recording behavior from normal network traffic examples and other relevant system data.

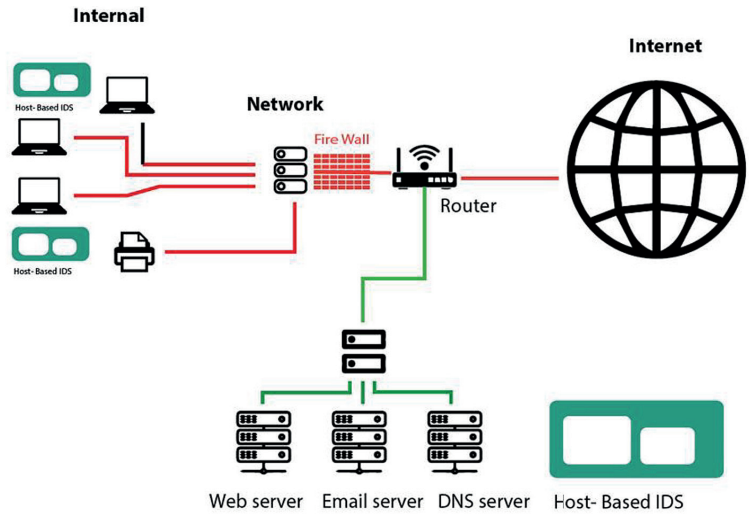

Fig. 5. HIDS sensor placement

3. Hybrid based

This strategy combines the signature and anomaly methods to optimize the benefits and minimize the downsides of both methods, thus increasing the detection rate of zero-day and new or unknown threats while lowering the number of false alarms. According to a survey by [11], no system is solely based on signatures or anomalies, and most intrusion detection systems are deployed in a mixed configuration.

IDS state-of-the-art works

Network intrusion detection has been investigated extensively. As a result, to gain a better understanding of the field, different methodologies must be classified and evaluated. Because Machine Learning (ML) and Deep Learning (DL) are continuously growing research topics, we look at several models and the techniques they have used in these two fields for less than a decade.

In [11], deep learning approaches for cybersecurity were discussed. They looked at thirty-five datasets and divided them into seven groups: datasets based on electrical network traffic, network traffic, Internet traffic, virtual private network traffic, application of Android, Internet-connected device traffic, and IoT traffic. They use the CSE-CIC-IDS 2018 and Bot-IoT datasets to train and analyze several deep learning models, as well as look at existing DL-based intrusion detection algorithms. Deep Belief Networks, Deep Boltzmann Machines, Deep Neural Networks, Convolutional Neural Networks, Restricted Boltzmann Machines, Recurrent Neural Networks, and Deep Autoencoder models appear to have similar kinds of detection performances.

A study of the application of machine learning to several domains of networking, including network security, is presented in [12]. They look at 36 techniques to evaluate misuse-based, anomaly-based, Deep and Reinforcement Learning-based, and hybrid intrusion detection using the KDD Cup 1999 (a dataset developed during a competition called the Third International Knowledge Discovery and Data Mining Tools Competition), and NSL-KDD (Network Security Laboratory Knowledge Discovery and Data Mining) datasets. They also discovered that there is a lack of recent datasets, real-world anomaly-based detection systems, insufficient real-time implementations, and a general lack of systems that meet other unique requirements. Finally, they offer a broader view of networking and point to the need for real-world data rather than generated datasets, as well as uniform evaluation metrics to make comparisons easier.

The researchers [13], investigate the use of machine learning and data mining to identify intrusions. Decision Trees, Clustering, Bayesian Networks, Artificial Neural Networks, Association Rules, Ensemble Learning, Evolutionary Computation, Hidden Markov models, Naïve Bayes, Sequential Pattern Mining, Inductive Learning, and Support Vector Machines are some of the methodologies available.

In each of these solutions, they take into account both misuse-based identification and anomaly detection. The next sections go through the computational complexity and streaming capabilities of each approach. They conclude their paper by discussing IDS performance, the challenges of comparing multiple detection techniques, and more. They conclude by discussing IDS performance, the challenges of comparing different detection algorithms, and model (re)trainability, as well as providing some recommendations.

A study by [14] provides an overview of NIDS-based IoT security, including IoT threats, publicly available datasets and tools, and current open-source NIDS. They also consider the effects of machine learning-based algorithms that have been evaluated on large-scale network intrusion detection datasets. One of their primary concerns is the necessity for a real-world IoT IDS dataset, with a stronger focus on the semantic relationships between detection performance and learning.

The authors of [15] give an overview showing the use of deep learning in several cybersecurity problems, including network intrusion detection. They explore recurrent neural networks, convolutional neural networks, deep neural networks, deep belief networks, autoencoders, and other methods in their study of network intrusion detection. Restricted Boltzmann machines, autoencoders, and recurrent neural networks were the most popular options across all security issues, according to them. Furthermore, they claim that the number of classes, types of attack, and benign-malicious ratio in the training data all have a significant impact on a system's intrusion detection ability. Finally, they analyze the consequences of false alarms and missed attacks, as well as the possibility that adversaries will actively try to circumvent security mechanisms.

A look at how Deep Learning can be used for intrusion detection using spam detection, phishing detection, website defacement detection, and malware detection, among other things has been presented by [16]. They focused on studies that use generative deep learning technology for intrusion detection rather than discriminative or mixed deep learning approaches. The authors also propose a general architecture or framework for using deep learning for cybersecurity based on the surveyed papers. Given the availability of unlabeled data in the area, they also mention the prospect of semi-supervised learning. They conclude by highlighting that deep learning should only be employed in industries that require complex nonlinear models and have sufficient data, as well as other learning-related features.

Several anomaly detections approaches are considered by [17]. Misuse-based algorithms like Support Vector Machines and rule-based approaches are among their anomaly detection techniques. They also explore IDS datasets and evaluate anomaly detection algorithms based on computational complexity, output format, and attack priority. However, because this analysis appears to be limited to DARPA/KDD Cup attacks, it has limited application to more current datasets.

In a study by [18], a comprehensive overview of the network is provided. Because they provide both a simplified overview and a large, in-depth table and discussion, their work can be used as a reference in selecting relevant public datasets for a certain goal. They also consider the impact of other data sources, such as data repositories and traffic producers. Finally, they draw some implications that could be useful in future studies using NIDS datasets. They address the impossibility of ever having a perfect dataset for example and recommend evaluating multiple datasets. While their analysis does not provide insight into the performance of certain algorithms or techniques, it does provide useful information.

For detecting network anomalies, [19] provide a study that includes statistical, classification-based, knowledge-based, soft computing, clustering-based, ensemble-based, fusion-based, and hybrid techniques. They also look at how to evaluate detection techniques and look at some of the tools, such as Nmap and Wireshark that can be used to discover network anomalies. Finally, they offer recommendations for network anomaly detection as well as a list of challenges.

A study by [20] gives an overview of Machine Learning and Deep Learning approaches in cybersecurity, focusing on network intrusion detection. They look at methods like k-Nearest Neighbor, Support Vector Machines, Deep Belief Networks, Decision Trees Recurrent Neural Networks, and Convolutional Neural Networks. They take notice of three issues in this overview: The scarcity of benchmark datasets and the non-uniformity of evaluation metrics, which makes comparison difficult, and finally, the lack of focus on algorithm efficiency. They also point out several trends in intrusion detection research, including hybrid model research, deep learning's prospects and problems, the expanding number of papers evaluating different algorithms and their applicability, and the potential for new benchmark datasets.

METHODOLOGY

In this section, the methodology of the research is discussed. According to the literature studies, there is a critical need for the creation of effective machine learning and deep learning models for identifying attacks in datasets. The dataset NSL-KDD was analyzed and trained using four Machine Learning algorithms Random Forest (RF), Naïve Bayes (NB), (J48), and Support Vector Machine (SVM). The general layout of the methodology is shown in Fig. 6.

Dataset

NSL-KDD is a condensed version of the original KDD dataset that was acquired from the Canadian Institute for Cybersecurity [21]. It has the same features as KDD. Each record has 41 features and one class attribute. Each connection is classified as either an attack or a normal connection. NSL-KDD has a total of 39 attacks, each of which is classified into one of four categories: DOS, R2L, U2R, and Probing. For building our models, we used 25,192 instances as training. Next, these trained models were evaluated and tested using 11851 instances. Finally, the rest of the dataset was used for validation.

1. DOS: denial-of-service, which means preventing authorized users' access to a service, such as syn flooding.
2. R2L: This refers to breaking into a remote machine to get access to the victim's machine, such as guessing passwords.

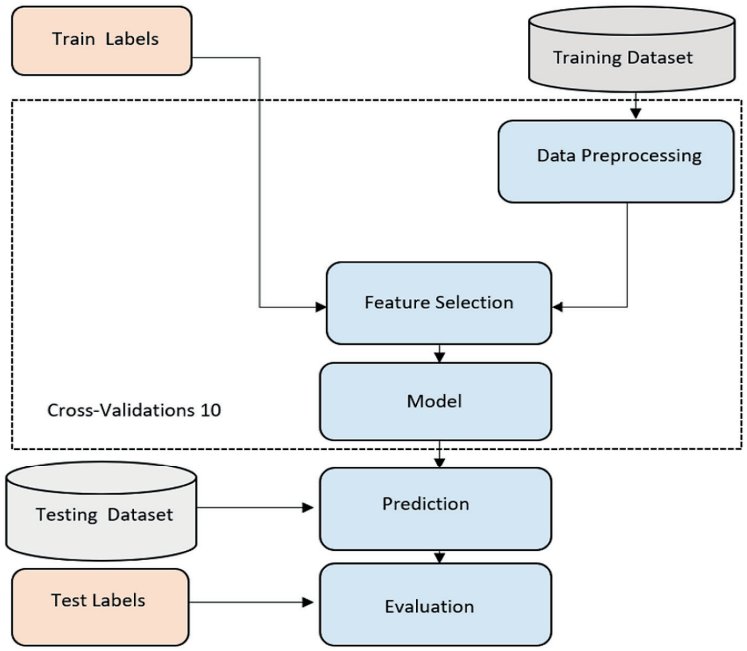

Fig. 6. Flowchart sequence of steps for build IDS models

3. U2R: When a normal account is used to log into a victim machine and tries to gain root privilege, using a technique such as a buffer overflow.
4. Probing: examining and scanning the victim's machine for vulnerabilities in terms of learning more about it, such as port scanning.

Data pre-processing

Pre-processing the data is a very important step in preparing the data to be fed into the algorithm. The goal of data preparation is to eliminate ambiguity in the dataset and provide IDS with accurate data. It unifies feature selection and normalization. Many symbolic attributes in the dataset, such as flags and protocol types, have nominal values. These values must be converted to numeric values for the dataset to perform better. Multi-class classification problems (4 attack classes and normal classes) and binary classification problems (normal or attack) have been transformed using discretized datasets in bin 10.

Table 1. Labels of dataset for binary class

Label (Class)	No. of instances
Normal	15601
Attack	21441

Feature selection

Feature Selection produces more enhanced and efficient subsets by eliminating redundant and unrelated features. Correlation is a popular and successful strategy for identifying the most closely linked characteristics in any dataset; it defines the strength of the relationship between features, based on the assumption that features are conditionally independent given the class. A good feature subset contains features that are highly correlated (predictive of) the class yet uncorrelated and not predictive of one another. The table shows the result of CFS SubSetEval-BestFirst was chosen for feature selection used in WEKA.

Table 2. Labels of dataset for multiclass

Class	No. of instances	Attack type
Normal	15601	-
DOS	13576	Apache2, Netpune, Pod, Land, Smurf, Mailbomb
PROBE	4691	Satan, Saint, Ipsweep, Portsweep, Msan, Nmap
U2R	411	Buffer Overflow, Httptuneel, Rootkit, LoadModule, Xtern, Perl, Sql Attack
R2L	2763	Phf, WarezMaster, Guess, Password, Imap, Spy, Xsnoop, Sendmail,

Table 3. CFS SubSetEval-BestFirst selection feature

(CFS SubSetEval – BestFirst)	Duration, Service, Flag, Source bytes, Destination bytes, Servor rate, Diff srv rate, Dst host srv diff host rate, Logged in, srv error rate, Same srv rate, Dst host srv count

Split and discretization

The main objective of discretization is to improve the overall classification performance while reducing storage space because discretized data takes up less space. An important step before classification is considered using several classifiers employing discrete data and classifiers using discrete data discretization. Discretization is numeric attributes that were discretized by use of a discretization filter using unsupervised 10 bin discretization on Weka. Also, one of the most important steps for building any machine learning model is splitting the dataset into training and testing modules. In this study, the dataset was split into three, 70% of data for training, 20% for testing, and the rest for validation, which is 10% of the data. Then, we renamed every attack label for binary and multi-classifications as normal traffic or attacks and determined the type of main categories of attacks on the datasets DOS, Probe, R2L, and U2R.

Classification process

For the supervised machine learning algorithms used to evaluate the performance of NIDS over the NSL-KDD dataset in this study, we used Support Vector Machines (SVM), J48, Random Forest (RF), and Naïve Bayesian (NB) algorithms for each type of feature selection method. In general, every process of classification in machine learning is divided into five steps:

1. Data collection

Data that is specific to the research being conducted is collected and stored in the memory [22].

2. Data preprocessing

At the data processing stage, data that has been collected is organized and translated to a format that is compatible for entry in the machine learning algorithm. Features of the data are extracted, and relevant information is acquired while irrelevant data is discarded.

3. Training

The machine learning algorithm receives a training dataset that originates from the preprocessing data stage. At this stage, a model is created, and the training stage is completed. The considered algorithms are: J48 (C4.5) [23], Support Vector Machines (SVM) [24], Naïve Bayesian [25], and Random Forests [25].

4. Testing

The dataset from the training phase is transferred to the machine learning algorithm of choice at this point, and a model is created. This stage may be completed only once or several times.

5. Deployment

At this phase, the best model for categorization or prediction is made depending on the need at hand. The selection is made after an intensive comparison of the models [22].

Evaluation metrics

The evaluation of the produced classification models is an important phase. It's also done through the use of a variety of evaluation metrics. The following are used on evaluation metrics:

- True Positives (TP) the total number of malicious packets correctly classified.
- True Negatives (TN) the total number of correctly classified as normal.
- False Positives (FP) the total number of malicious packets incorrectly classified as attacks.
- False Negatives (FN) the total number of malicious packets incorrectly classified as normal.

Classification accuracy is the most commonly used statistic for evaluating a model, however, it is not a reliable predictor of its performance. The appropriate classification ratio is the proportion of correctly classified samples to the total number of input samples. It is calculated using the following formula:

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

Precision: It's the number of successfully classified positive samples divided by the number of samples that the classifier predicted as positive (i.e. the proportion of positive samples correctly classified to the all predicted as positive). Its formula is as follows:

$$\text{Precision} = TP/(TP+FP) \quad (2)$$

Recall: It is calculated by dividing the number of correctly classified positive samples by the total number of positive samples passed.

$$\text{Recall} = TP/(TP+FN) \tag{3}$$

Mathews Correlation Coefficient (MCC): It represents the relative correlation between observed and predicted binary classifications.

$$MCC = (TP*TN - FP*FN). / / \text{sqrt}[(TP+FP)*(TP+FN)*(TN+FP)*(TN+FN)] \tag{4}$$

Experimental results

This section shows the results obtained from training ML models on the NSL-KDD train-21 set and evaluating them using the NSL-KDD test-20 set, in both cases using binary and multi-class classification. Machine learning-based techniques such as J48, RF, NB, and SVM are used to organize the evaluation results. The experiment is conducted using a Windows 10 operating system on an HP Pavilion computer operating at 1.80 GHz and with 16 GB RAM. The environment is built on free software that is called Weka, which was developed at the University of Waikato in New Zealand. The performance of these classifiers was tested on 13 features of the dataset for first, detections if the network flow is normal or attacks, and second if the detections show if there are any type of attacks on this flow. It can also determine attack types such as DOS, Probe, U2R, and R2L. Tables 4 and 5 represent the performance of the four classifiers in binary and multi-classification using different evaluation metrics such as precision, recall, FPR, and MCC. In both binary and multi-classifications, the RF classifier achieves the highest score.

Comparison of intrusion detection models

The performance of each Random Forest, Naïve Bayesian, J48, and Support Vector Machines classifier used to develop our model is compared to the performance of an existing intrusion model on the same dataset. Given below is the detailed comparison:

Comparing random forest classifiers

In the approach proposed by [26] the RF algorithm yields an accuracy of 99.9% for DOS, 99.9% for Probe, 99.8.8% for R2L, and 99.0% for U2R. Compare to our model result, which is shown in Table 6. We can see that our model provided higher accuracy for DOS and the Probe attack compared to another model.

Comparing J48 classifiers

We compared the accuracy results of our model with the performances of the paper (Bhumgara and Pitale, 2019). Our model had a DOS accuracy of 99.7%, a Probe accuracy of 99.2%, and an R2L accuracy of 98.9% for U2R. Furthermore, it showed that our model performs better on DOS, Probe, R2L, and U2R as shown in Table 7.

Comparing the results of the Naïve Bayesian (NB) classifier

The accuracy of the types of attacks are 87.5%, 87.7%, 90,2%, and 93,7 (Kumar and Singh, 2016). However, our model provides better results for accuracy with DOS at 98.1%, Probe

Table 4. ML-based approach results using binary classification

Classifier	Acc. (%)	Precision (%)	Recall (%)	MCC (%)
RF	98.7	98.8	98.8	97.5
J48	98.2	98.2	98.2	96.3
SVM	97.7	97.8	97.8	95.5
Bayesian	94.9	95.0	95.0	89.7

Table 5. ML-based approach results using multi-class classification

Classifier	Acc. (%)	Precision (%)	Recall (%)	MCC (%)
RF	97.9	98.0	98.0	96.9
J48	97.4	97.5	97.4	95.9
SVM	96.4	96.5	96.5	94.3
Bayesian	87.4	88.8	87.4	79.0

Table 6. Comparing the results of accuracy of the RF model with the other model

Parameter	Accuracy (%)	
	[26]	Our model
DOS	99.7	99.9
R2L	99.7	99.8
Probe	99.7	99.9
U2R	99.7	99.0

Table 7. Comparing the results of the accuracy of the J48 model

Parameter	Accuracy (%)	
	[27]	Our model
DOS	98.1	99.7
R2L	97.7	99.2
Probe	97.6	99.2
U2R	97.5	98.9

Table 8. Comparing the results of the accuracy of the NB model

Parameter	Accuracy (%)	
	[28]	Our model
DOS	87.5	98.1
R2L	87.7	97.5
Probe	90.2	98.2
U2R	93.7	95. 1

Table 9. Comparing the results of the accuracy of the SVM model

Parameter	Accuracy (%)	
	[29]	Our model
DOS	98.7	98.7
R2L	92.5	98.0
Probe	91.4	97.5
U2R	94.6	98.2

at 98.2%, U2R at 95.1%, and R2L at 97.5% as shown in Table 5.5.

Comparing the results of the SVM classifier

Model validation

In the final step, the model will be implemented and trained based on the decisions made in the previous processes, and then validated to see if it meets all of the preconditions and to see how accurate it is at predicting with new data. The model's flaws and limitations are recognized as a result of these assessments, allowing the required measures to be taken to address them. In comparison to other algorithms, the experiment shows that RF has the highest accuracy, followed by the J48 algorithm. Table 10, shows that selecting 13 features for each algorithm provides high accuracy in the binary class. The models have a closer accuracy of 98.92% and F-measure 98.9%, respectively. It shows that the model is the best for detecting DOS attacks. Table 11, shows the same results for multi-class, with slight changes in accuracy, which was high in model RF.

CONCLUSIONS

In this paper, a Network Intrusion Detection System was presented utilizing machine learning techniques. A thorough evaluation on the performance of the proposed detection system using multiple machine learning algorithms on the NSL-KDD dataset. The results show that Random Forest performed well compared to the other models in predicting the malicious packets, especially in terms of accuracy, recall, and the Mathews correlation coefficient. Moreover, the RF classifier outperformed state-of-the-art intrusion detection systems. Although, the NSL-KDD dataset suffers from several issues, such as imbalanced classes,

Table 10. Cross-validation and test result of binary classification

Classifier	Test options	Accuracy (%)	Precision (%)	Recall (%)	F- measures (%)	MCC (%)	Roc area (%)
RF	Cross Validation	98.92	98.9	98.9	98.9	97.8	99.9
	NSL-KDD Test	98.77	98.8	98.8	98.8	97.5	99.9
J48	Cross Validation	98.00	98.0	98.0	98.0	95.9	99.5
	NSL-KDD Test	98.20	98.2	98.2	98.2	96.3	99.6
SVM	Cross Validation	97.26	97.3	97.3	97.3	94.4	97.2
	NSL-KDD Test	97.79	97.8	97.8	97.8	95.5	97.8
Bayesian	Cross Validation	94.12	94.1	94.1	94.1	88.1	98.6
	NSL-KDD Test	94.97	95.0	95.0	95.0	89.7	98.7

Table 11. Cross-validation and test result of multi-classification

Classifier	Test options	Accuracy (%)	Precision (%)	Recall (%)	F- measures (%)	MCC (%)	Roc area (%)
RF	Cross Validation	99.9	97.2	97.3	97.3	97.3	95.3
	NSL-KDD Test	97.9	98.0	98.0	98.0	96.9	99.9
J48	Cross Validation	96.9	97.0	97.0	97.0	95.3	99.6
	NSL-KDD Test	97.4	97.5	97.4	97.4	95.9	99.5
SVM	Cross Validation	96.2	96.3	96.2	96.2	94.0	98.4
	NSL-KDD Test	96.4	96.5	96.5	96.5	94.3	98.4
Bayesian	Cross Validation	86.2	88.0	86.3	86.7	77.2	97.2
	NSL-KDD Test	87.4	88.8	87.4	87.8	79.0	98.0

and the recorded malicious traffic are synthetic, it does not reflect real-world attacks. The classifiers have presented satisfactory results and are capable of detecting network intrusions.

REFERENCES

1. Cisco Annual Internet Report (2018–2023) White Paper. (2022, January 23). Cisco. https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11–741490.html
2. Dyn Analysis Summary of Friday October 21 Attack (2022, February 20). https://web.archive.org/web/20200620203923
3. Dartigue, C., Jang, H.I., Zeng, W. A new data-mining based approach for network intrusion detection. In Seventh Annual Communication Networks and Services Research Conference. 2009; 372–377.
4. García-Teodoro, P., Díaz-Verdejo, J., Maciá-Fernández, G., Vázquez, E. Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security. 2009; 28(1–2); 18–28.
5. Cisco. What Is Network Security? (2022, February,8). Cisco. https://www.cisco.com/c/en/us/products/security/what-is-network-security.html
6. Kurose, J.F., Ross, K.W. Computer Networking: A Top-Down Approach (6th Edition). Pearson, 2012.
7. Tanenbaum, A., Wetherall, D. Computer Networks (5th Edition). Pearson, 2010.
8. Fernandes, G., Rodrigues, J.J.P.C., Carvalho, L.F., Al-Muhtadi, J.F., Proença, M.L. A comprehensive survey on network anomaly detection. Telecommunication Systems. 2018; 70(3): 447–489.
9. Othman, S.M. Alsohybe, N.T., Ba-Alwi, F.M., Zahary, A.T. Survey on intrusion detection system types. 2018; 7(4): 444–463.
10. Pal Singh, A., Deep Singh, M. Analysis of Host-Based and Network-Based Intrusion Detection System. International Journal of Computer Network and Information Security, 2014; 6(8): 41–47.
11. Ferrag, M.A. Maglaras, L. Moschoyiannis, S., Janicke, H. Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications. 2020; 50.
12. Boutaba, R. Salahuddin, M.A., Limam, N., Ayoubi, S., Shahriar, N., Estrada-Solano, F., Caicedo, O.M. A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. Journal of Internet Services and Applications. 2018; 9(1).
13. Buczak, A.L., Guven, E.A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. IEEE Communications Surveys & Tutorials. 2016; 18(2): 1153–1176.
14. Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C., Faruki, P. Network Intrusion Detection for IoT Security Based on Learning Techniques. IEEE Communications Surveys & Tutorials. 2019; 21(3): 2671–2701.
15. Berman, D., Buczak, A., Chavis, J., Corbett, C. A Survey of Deep Learning Methods for Cyber Security. Information. 2019; 10(4): 122.
16. Mahdavifar, S., Ghorbani, A.A. Application of deep learning to cybersecurity: A survey. Neurocomputing. 2019; 347: 149–176.
17. Ahmed, M., Naser Mahmood, A., Hu, J. A survey of network anomaly detection techniques. Journal of Network and Computer Applications. 2016; 60: 19–31.
18. Ring, M., Wunderlich, S., Scheuring, D., Landes, D., Hotho, A. A survey of network-based intrusion detection data sets. Computers & Security. 2019; 86: 147–167.
19. Bhuyan, M.H., Bhattacharyya, D.K., Kalita, J.K. Network Anomaly Detection: Methods, Systems and Tools. IEEE Communications Surveys & Tutorials. 2014; 16(1): 303–336.
20. Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., Gao, M., Hou, H., Wang, C. Machine Learning and Deep Learning Methods for Cybersecurity. IEEE Access. 2018; 6: 35365–35381.

21. UNB (2021, Novamber 15). https://www.unb.ca/cic/datasets/nsl.html
22. Chumachenko, K. Machine learning methods for malware detection and classification., 2017.
23. Zou, J., Han, Y., So, S.S. Overview of artificial neural networks. Methods in molecular biology (Clifton, N.J.). 2008; 458: 15–23.
24. Dong, B., Wang, X. Comparison deep learning method to traditional methods using for network intrusion detection. In 2016 8th IEEE International Conference on Communication Software and Networks (ICCSN). 2016; 581–585.
25. Mahesh, B. Machine Learning Algorithms – A Review. International Journal of Science and Research (IJSR). 2020; 381–386.
26. Farnaaz, N., Jabbar, M.A. Random forest modeling for network intrusion detection system. Procedia Computer Science. 2016; 89: 213–217.
27. Bhumgara, A., Pitale, A. Detection of Network Intrusions using Hybrid Intelligent Systems. 1st International Conference on Advances in Information Technology (ICAIT). 2019; 500–506.
28. Kumar, K., Batth, J.S. Network Intrusion Detection with Feature Selection Techniques using Machine-Learning Algorithms. International Journal of Computer Applications. 2016; 150(12): 1–13.
29. Dhanabal, L., Shantharajah, S.P. A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. International journal of advanced research in computer and communication engineering. 2015; 4(6): 446–452.