REPUBLIC OF TÜRKİYE
ALTINBAŞ UNIVERSITY
Institute of Graduate Studies
Information Technologies

DNS TRAFFIC BASED INTRUSION DETECTION USING MODIFIED KNN ALGORITHM AND FEATURE WEIGHTING

Haneen Kadhim Hwaidi Al-FADHLI

Master's Thesis

Supervisor

Prof. Dr. Osman Nuri UÇAN

İstanbul, 2024

DNS TRAFFIC BASED INTRUSION DETECTION USING MODIFIED KNN ALGORITHM AND FEATURE WEIGHTING

Haneen Kadhim Hwaidi AL-FADHLI

Information Technologies

Master's Thesis

ALTINBAŞ UNIVERSITY

2024

The thesis titled DNS TRAFFIC BASED INTRUSION DETECTION USING MODIFIED KNN ALGORITHM AND FEATURE WEIGHTING prepared HANEEN KADHIM HWAIDI AL-FADHLI and submitted on 17/01/2024 has been **accepted unanimously** for the degree of Master of Science in Information Technologies.

_____

Prof. Dr. Osman Nuri UÇAN

Supervisor

Thesis Defence Committee Members:

Prof. Dr. Osman Nuri UÇAN	Department of Electrical Engineering, Altınbaş University	_____
Asst. Prof. Dr. Abdullahi Abdu IBRAHIM	Department of Electrical and Computer Engineering, Altınbaş University	_____
Assoc. Prof. Dr. Adil Deniz Duru	Department of Biomedical Sciences Engineering, Marmara University	_____

I hereby declare that this thesis meets all format and submission requirements of a Master's thess

I hereby declare that all information/data presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Haneen Kadhim Hwaidi AL-FADHLI

Signature

ACKNOWLEDGEMENTS

I want to start by giving Allah, the Almighty, praise and thanks for giving me the knowledge, abilities, and opportunity to carry out and successfully complete my research. I genuinely thank my parents. Everything wonderful that has happened to me has been fueled by them, and their love is what makes everything possible. During this time, I'd like to express my gratitude to my sister and brother for their love and support.

ABSTRACT

DNS TRAFFIC BASED INTRUSION DETECTION USING MODIFIED KNN ALGORITHM AND FEATURE WEIGHTING

AL-FADHLI, Haneen Kadhim Hwaidi
M.Sc., Information Technologies, Altınbaş University,
Supervisor: Prof. Dr. Osman Nuri UÇAN
Date: 01 / 2024
Pages: 77

All computer networks can be the target of attacks, from the smallest, such as home networks, to the largest and most complex, such as corporate networks. To make the situation even worse, programs can easily be obtained through the Internet (or by other means), and then used for malicious actions, making any inexperienced person have the same skills as a professional It is necessary to propose a new database that contains a portion of the traffic that is present on an existing actual network environment in order to carry out testing in a real environment. This is necessary in order to accomplish the goal of conducting testing in a real environment The main objective is to create a network user control mechanism in order to recognize the user's legitimacy through their actions.

Keywords: MLP, Intrusion Detection , AI, ML, DL.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

ABBREVIATIONS

API	:	Application Programming Interface
CFA	:	Cuttlefish Algorithm
DL	:	Deep Learning
HMM	:	Hidden Markov Models
IDS	:	Intrusion Detection System
IoT	:	Internet of Things
ML	:	Machine Learning
NIDS	:	Network Intrusion Detection System
PDA's	:	Portable Device Adapter

1. INTRODUCTION

1.1 INTRODUCTION

With the exponential increase in the number of computer networks, there has been an increase in attacks. All computer networks can be the target of attacks, from the smallest, such as home networks, to the largest and most complex, such as corporate networks. To make the situation even worse, programs can easily be obtained through the Internet (or by other means), and then used for malicious actions, making any inexperienced person have the same skills as a professional. The question then arises: how can a computer network be protected? The solution may include the implementation of an Intrusion Detection System (IDS). An IDS inspects network traffic, identifies packets that could compromise the network, and then alerts the computer network administrator.

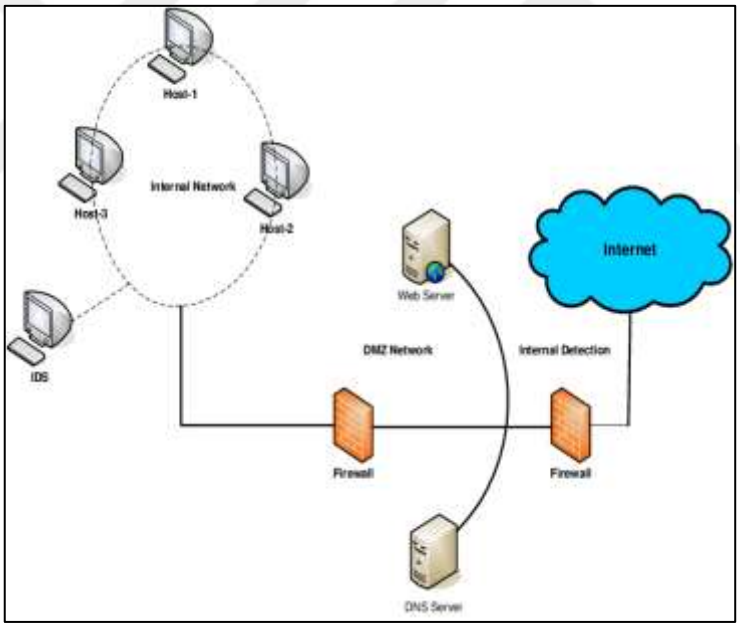


Figure 1.1: Intrusion Detection System (IDS).

An intrusion detection system (IDS) is intended to detect and alert administrators about a possible intrusion attempt that may occur in a network. In addition, it allows eliminating unauthorized access to a computer, in case it has a suspicious attitude or intends to damage a network device. The IDS is an important security mechanism, as it allows identifying possible vulnerabilities and weaknesses that compromise the security of a network [2] There are three classifications of IDS: network-based, host-based and distributed, and in this work

a host IDS is used to analyze the traffic destined to a certain group of machines in which sensors are installed. One of the ways to implement an IDS is through the Snort software. Snort has several features, being a sniffer, logger and a powerful intrusion detector. Security is an open question even today. Many researchers have developed work in the security area and are always looking for new mechanisms or methods to control the invasion of systems by malicious users. Security problems in current systems are mainly caused by application and protocol implementation errors and operating system vulnerabilities. In the field of applications, vulnerabilities appear continuously, because with the advent of the Internet, the number of applications has grown rapidly and are available to an extensive community of users. Currently, there is also a rapid growth in the use of mobile devices and the use of wireless networks, which in turn introduce another range of security problems, given the specific characteristics of the absence of cables and the mobility of equipment and users. These environments require greater control over users and their actions, as well as their credentials, in order to prevent the use of environment resources and user credentials by malicious and unauthorized users.

1.2 PROBLEM STATEMENT

The currently adopted security mechanisms, in general, comprise the use of encryption methods for data and user protection and authentication, access control mechanisms, and intrusion detection mechanisms in particular, intrusion detection systems are used to detect occurrences of suspicious situations, or real attacks, and also to prevent systems against possible attacks. Much research has been carried out in the area of intrusion detection systems in order to optimize their efficiency, with efficiency related both to the number and quality of detections and to their computational performance. Several mechanisms have been studied, such as the use of data mining methods. These mechanisms have been widely used in the anomaly detection area, which currently represents one of the biggest challenges in the area of intrusion detection systems, as it must identify situations that deviate from a behavior pattern considered normal. This work is focused on the area of anomaly detection based on the use of statistical methods to identify behavioral deviations and control access to network resources. The use of a user control system can be conveniently used in wireless networks to monitor network users and validate their activities, in order to protect the network and users from misuse of resources. In traditional networks, that is, wired, there is

also the possibility of using this detection system, but in these networks hosts usually use multiuser systems, which allow the remote connection of several users, which prevents the association of actions from a given host to a user. In wireless networks, with the use of PDA's (Portable Device Adapter) the association of users to these devices is facilitated because they are usually used by a single user. This allows user actions to be compared with expected (known) behavior in order to determine their legitimacy.

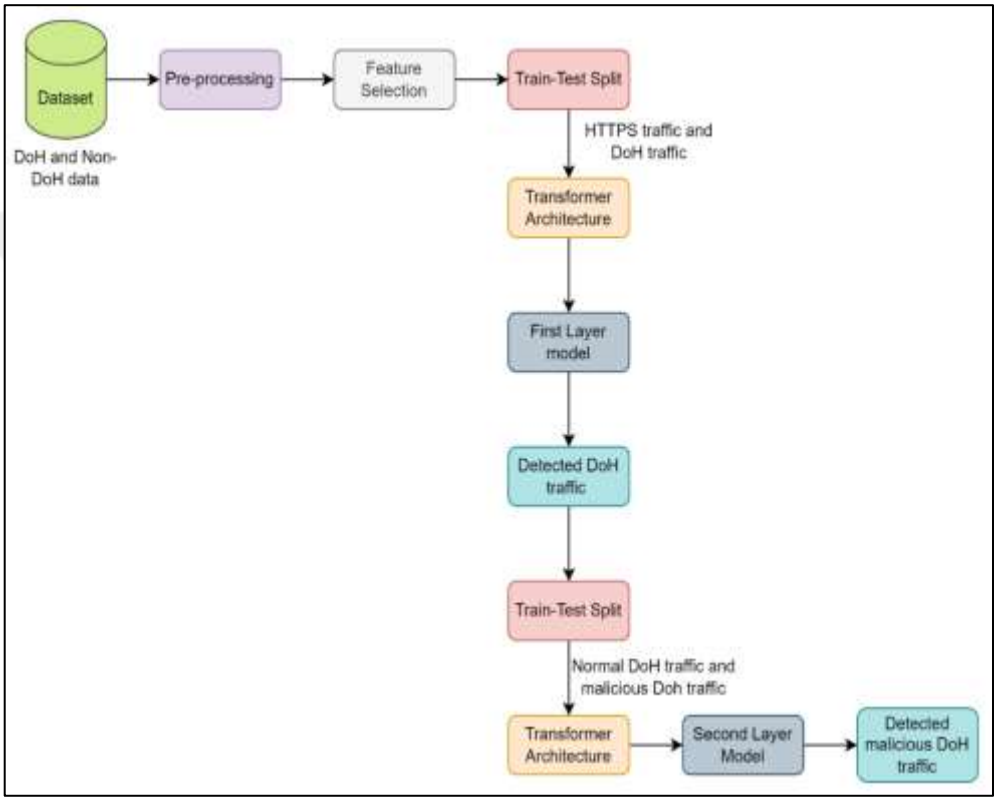


Figure 1.2: DNS Attack on PDAs.

The analysis of user behavior presupposes the construction of knowledge about their actions, that is, the construction of a user profile. Behavior profiles are widely used in anomaly detection systems [4], as the idea of an anomalous is something that deviates from a known pattern. The data that must compose the profile are strictly related to the object of protection of intrusion detection systems [5]. This system is going to monitor the activities of the user when they are connected to a network, which is the reason why it is being created in the first place. Identifying instances in which the activities of the user deviate from their typical patterns is the overall objective of the system that has been made available to the user. The usage of the statistical procedures that have been provided will be the means by which this objective will be realized. Because of the ease of use and, at the same time, the dependence

that it offers us with tools like as social networks, online personal and company management systems, and the ability to easily access information, the utilization of the internet has significantly increased throughout the course of the period of time that has passed. Today, almost everyone relies on the internet for their careers, their education, and even their day-to-day life at home. This includes both professional and personal purposes. This encompasses both educational and professional endeavors for the individual. The ability to connect to the internet is present in each and every one of these things. It has been observed that there has been an increase in the number of occurrences of traffic anomalies that result in failures, with denial of service being the most common sort of failure that has taken place. In order to guarantee that a network will continue to operate regularly, it is essential to possess the capability to identify, investigate, and rectify any irregularities that may emerge inside the network. The utilization of a technology that is able to offer aid in the protection of the data is required in order to ensure the safety of this enormous quantity of data. This is the only way to guarantee the data's safety. While there is a possibility that the apps and operating systems that are utilized in diverse portions of the Internet could pose a threat to the network's security, there is also the likelihood that this situation could occur. For the purpose of identifying and preventing the exploitation of vulnerabilities in computer systems, it is necessary to have a Network Intrusion Detection System (NIDS) in place. This is essential in order to stop the vulnerabilities from being exploited after they have been discovered. Network Intrusion Detection Systems, or NIDS for short, are a collection of software tools that are designed to investigate and identify any intrusions that may occur in data networks. These intrusions may be caused by malicious actors or malicious software. One of the distinguishing characteristics of network intrusion detection systems is their capacity to monitor and anticipate attacks. This ability has the impact of lowering the probability that an infiltration will take place. Obtaining a high hit rate while simultaneously obtaining a low false alert rate is one of the primary objectives of an Intrusion Detection System (IDS). At this time, there are three primary categories of intrusion detection systems that are easily accessible. The host-based, network-based, and hybrid systems are all included in this category. In addition, you have the option of classifying them according to signatures, outliers, or detection methods. Every one of these choices is up to you to choose from. There are a number of suggestions that have been put up in order to address the problem of intrusion detection in computer networks. The implementation of these

guidelines requires the application of methods that are linked with computational intelligence. Having said that, not a single one of these suggestions has been put into action in a context that is analogous to the real world.

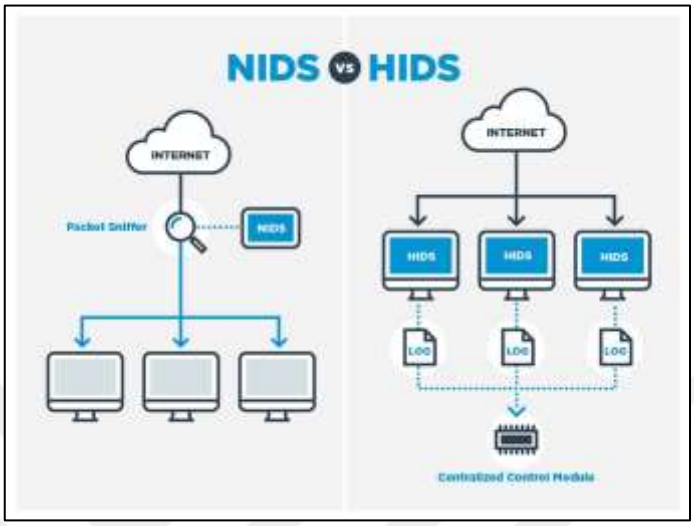


Figure 1.3: NIDS vs HIDS.

There are a number of obstacles that need to be overcome before an unsupervised intrusion detection system (NIDS) that makes use of artificial intelligence techniques can be implemented in a real-world setting. The obligation to perform pre-processing on data that is discovered on the internet is one of the problems that must be overcome. The primary objective of this dissertation is to construct a Network Intrusion Detection System (NIDS) that is capable of identifying anomalies in a real-world environment. This will be accomplished by utilizing and testing the following Artificial Neural Network (ANN), Support Vector Machine (SVM), and Random Forest (RF) approaches. There are a variety of approaches that have been proposed to solve the issue of intrusion detection in computer networks. The majority of these ideas include the utilization of computational intelligence; however, these strategies have not yet been put into practice. There are a number of obstacles that need to be overcome before an unsupervised intrusion detection system (NIDS) that makes use of artificial intelligence techniques can be implemented in a real-world setting. The obligation to perform pre-processing on data that is discovered on the internet is one of the problems that must be overcome. The primary objective of this dissertation is to construct a Network Intrusion Detection System (NIDS) that is capable of identifying anomalies in a real-world environment. This will be accomplished by utilizing and testing the following Artificial Neural Network (ANN), Support Vector Machine (SVM), and Random Forest

(RF) approaches. There are a variety of approaches that have been proposed to solve the issue of intrusion detection in computer networks. The majority of these ideas include the utilization of computational intelligence; however, these strategies have not yet been put into practice. There are a number of obstacles that need to be overcome before an unsupervised intrusion detection system (NIDS) that makes use of artificial intelligence techniques can be implemented in a real-world setting. The obligation to perform pre-processing on data that is discovered on the internet is one of the problems that must be overcome. The major purpose of this dissertation is to develop a Network Intrusion Detection System (NIDS) that is capable of identifying abnormalities in a real-world situation. This will be accomplished by utilizing and testing the following methodologies: Artificial Neural Network (ANN), Support Vector Machine (SVM), and Random Forest (RF). It is important to note that these approaches are not applied in the real world. There are a number of obstacles that need to be overcome before an unsupervised intrusion detection system (NIDS) that makes use of artificial intelligence techniques can be implemented in a real-world setting. The obligation to perform pre-processing on data that is discovered on the internet is one of the problems that must be overcome. The major purpose of this dissertation is to develop a Network Intrusion Detection System (NIDS) that is capable of identifying abnormalities in a real-world situation. This will be accomplished by utilizing and testing the following methodologies: Artificial Neural Network (ANN), Support Vector Machine (SVM), and Random Forest (RF). It is important to note that these approaches are not applied in the real world. There are a number of obstacles that need to be overcome before an unsupervised intrusion detection system (NIDS) that makes use of artificial intelligence techniques can be implemented in a real-world setting. The obligation to perform pre-processing on data that is discovered on the internet is one of the problems that must be overcome. During the course of this dissertation, the exclusive focus will be on a Network Intrusion Detection System (NIDS) that is capable of recognizing irregularities. This particular system employs and evaluates a number of different machine learning strategies, including Random Forests, Artificial Neural Networks, and Support Vector Machines. An environment that is representative of the real world will be used for the implementation of the system.

1.3 THESIS AIMS

An investigation into intrusion detection strategies in DNS networks is the primary purpose

of this work. This will be accomplished by utilizing methods that are founded on artificial intelligence in a setting that is representative of the actual world. Implementation, testing, and evaluation of the techniques of KNN, Support Vector Machines, and Random Forests will be carried out with the use of connection information that has been gathered from the traffic on a computer network and saved in a database. It is necessary to propose a new database that contains a portion of the traffic that is present on an existing actual network environment in order to carry out testing in a real environment. This is necessary in order to accomplish the goal of conducting testing in a real environment. You should provide a last recommendation, which is to suggest an Application Programming Interface (API) for the building of a Network Intrusion Detection System (NIDS). This API should be compatible with these and other ways to CI, and it should also be able to construct new databases of network traffic.

1.4 THESIS OBJECTIVES

The main objective is to create a network user control mechanism in order to recognize the user's legitimacy through their actions. The choice for statistical methods is based on the fact that these methods are described in the literature as methods of good computational performance, with a good degree of intrusion detection and are more flexible and thus present a lower sensitivity to false events [3]

1.5 CONTRIBUTIONS

The contributions of this thesis are divided into the following topics.

i. Study the techniques of KNN, Support Vector Machines and Random Forests, their libraries and/or tools in DoH based attack detection
ii. Develop an API to capture and preprocess network traffic.
iii. Pre-process the UNB CIC database with the API.
iv. Use the libraries and/or tools of Computational Intelligence techniques, applying the UNB CIC database.
v. Create a database, in a real environment, for testing.
vi. Join the API to the Computational Intelligence module (via socket), thus forming a NIDS.
vii. Evaluate the results of each technique.

1.6 MOTIVATION

Because of the persistent exposure of application vulnerabilities, it is conceivable to assert without a shadow of a doubt that no server can be considered to be completely secure. This is because of the circumstances described above. If an individual is serious about ensuring that their network remains secure, they are required to make use of security technology such as firewalls. It is for this reason that an intrusion detection system is one of the most important things to have. It is possible that the firewall and other network security measures failed to detect an attempted attack; still, a Network attack Detection System (NIDS) is able to alert the administrator to the threat. As a result of this, the administrator of the network is able to reset the security equipment after examining the alarms that were generated by the NIDS. As a consequence of this, it is abundantly clear that the topic of information security has a significant opportunity for many forms of research and application. These methodologies have not been tested in a real-world setting and do not make use of the distinctive characteristics of Network Intrusion Detection Systems (NIDS), which depend on a variety of host-specific data rather than merely network traffic. Although there are related works that provide Intrusion Detection System (IDS) methodologies that are based on computational intelligence techniques, these methodologies have not been tested in a real-world setting. Because of this, it is not possible to evaluate the efficacy of the method on various types of infrastructure and traffic that are not contained within the database that is utilized for the purpose of training these algorithms. It is important to design an application programming interface (API) that pre-processes data online in order to run tests in a real-world context and with a range of network architectures. This is because it is necessary to perform tests.

1.7 PROPOSED MODEL

The architectures of new information and telecommunication technologies are developing towards systems based on easily accessible infrastructures, open to a wide range of users and with more and more services. These properties offer many advantages to users such as ease of access, flexibility of use, mobility. But for designers and system administrators, these advantages can lead to many disadvantages, such as problems with updates, user management and problems with information security and protection. Security issues are a key factor in assessing the reliability of systems. If these problems

are well managed then the reliability of the system increases considerably and the added value of the system will be considerable. Otherwise, the reliability is called into question and the system will not be usable and therefore without added value. So system protection has become a very important task for administrators. They can use several tools and protection mechanisms such as:

i. Encryption mechanisms to ensure data confidentiality and integrity.
ii. Firewalls to control access and filter network traffic.
iii. Vulnerability scanners to detect security flaws in the system.
iv. Antiviruses to protect the system against malicious programs.
v. Intrusion detection systems to detect any illegal use of the system.

These tools play complementary roles, they can be used separately as they can cohabit to carry out the system protection process. In this chapter we will focus on intrusion detection systems, which represent a very important tool in the security of information systems. We introduce the different basic concepts of the field, as well as the research work related to intrusion detection and which relates to the needs of this thesis.

1.7.1 Definition

An intrusion detection system (IDS) is a software or hardware device or a combination of both. It is responsible for monitoring an information system to detect any breach in the use of resources. Breaking or intrusion is often defined as illegal entry into the system, an attempt by a system user to obtain unauthorized privileges, or any attempt to compromise standard security services: confidentiality, integrity or availability of information. Therefore, an IDS can be defined as a control mechanism whose purpose is to detect threats that affect the proper functioning of the information system. These threats can be internal or external, related to actions that exploit known or unknown flaws in the system. The result of these actions is either an intrusion or an attack. Intrusion detection is a very complex task, due to the diversity of environments and the number of parameters involved in the detection process. Several research works have dealt with the problems related to IDS. There are several architectures, approaches and detection standards. In the following sections we discuss the different aspects related to the actions of IDS.

Architecture of an Intrusion Detection System:

Several architectures have been proposed to describe the different elements involved in an

intrusion detection system. The simplest architecture is composed of three modules: the data source, the data analyzer and the response module (figure 3.1).

i. The data source: also called capture probe or sensor, it takes care of the recovery of information and events related to the detection, to send them to the analysis module. The position of the capture probe plays a very important role in the quality of detection. Several probes can be used in the same IDS. These probes will be positioned at strategic points in the system.
ii. The data analyzer: This is the heart of the IDS, this module makes it possible to analyze the information collected by the capture probes. It uses a knowledge base related to attacks, and for the search for traces of malicious activities it applies analysis models. Several analysis models exist and will be detailed later.
iii. The response module: This is the module that provides IDS responses to detected malicious activities. The responses can be active or passive, these are the countermeasures necessary to counter the intrusions. It can be a simple alert message, a save in a log file or even interrupt a connection.

2. LITERATURE REVIEW

Critical national infrastructures are dependent on Industrial Control Systems as well as Supervisory Control and Data Acquisitions for the purpose of managing their output through the Internet of Things (IoT). This management is accomplished through the use of SCADA. Hospitals, water and gas utilities, and electrical transmission and distribution firms are just some of the potential targets of cyberattacks that are becoming more likely to target various sections of the nation's infrastructure. Because of the threats that were outlined earlier [1], it is of the utmost significance to protect Europe's networks, information, and electronic communications. [T]his should be a priority. In order to protect a network from any and all kinds of cyberattacks, it is vital to have a number of different lines of protection in place. Intrusion detection systems, more commonly abbreviated as IDS, are one type of defense that a company might employ [2]. IDSs determine whether a behavior is harmful or harmless based on rules that characterize attack patterns or typical system behavior [3]. These rules are used to detect whether a behavior is malicious or benign. In order to obtain increased performance, IDS must fulfill the qualities of high accuracy, minimal complexity, and efficient use of time [4]. Data mining demonstrates more resilient behavior in comparison to standard IDSs due to the knowledge it unearths about new sorts of attack [5]. This is because data mining uncovers information about previously unknown intrusions. Because information on previously undisclosed incursions can be found through data mining, this is the case. It is vital to make use of dependable datasets that contain instances of both normal activity and a wide variety of different kinds of attacks [6, 7] when evaluating the efficacy of various approaches for intrusion detection. This is because it is only possible to determine which methods are most successful. Every industry has been revolutionized as a result of the introduction of artificial intelligence into the future plans for the ease of the internet of things. This has made things intelligent that were once considered to be senseless. Without the help of other people, it is feasible to improve the level of living of humans [8]. An increase in the frequency of attacks on the Internet of Things has been ascribed to mathematical analytical methods and network protocols that are not secure [9]. We take a look at the challenges that various types of research have to surmount, and we emphasize the significant aspects of the method of discovery. Assaults that were carried out using IoT. To get at an accurate assessment of the gravity of these assaults, both the model and any relevant earlier research are subjected to an analysis that is based on the application of parameters that are regarded

as being statistically sound.

2.1 MALICIOUS INFORMATION ON THE INTERNET

The use of numerous enterprises, educational institutions, and social activities simultaneously led to an increased dependence on global network connections, which in turn led to the development of new technologies. The ever-increasing number of computer networks has resulted in an increase in the number of holes in the security of the Internet, many of which have only lately been discovered. As a consequence of this, ensuring the security of devices that are connected to the internet is very necessary in order to ensure the availability and integrity of the system [10]. Typical data collection methods for network traffic include the use of packet and stream formats. Normally, network traffic is captured at the packet level by replicating ports to different network devices. The data from this traffic consists of information regarding payloads. The network traffic is recorded as a continuous stream when using the stream format. Flow-based data only includes metadata about the connections that are made between networks [11]. [12] Methods like as administration of networks, analysis of user information networks and services, and early detection of security issues are all useful ways for passive traffic collecting and analysis. The systems are able to receive the necessary level of protection via firewalls and other authentication procedures, which can also prevent unauthorized users from having access to the systems. However, because these methods were used, we were no longer able to monitor the activity on the network. This was an especially challenging situation in the event of internal attacks carried out by dissatisfied personnel who had been authorized lawful network access and utilized that access to cause destruction [13]. Figure 2.1 presents an illustration of the processes involved in the process of ensuring network security.

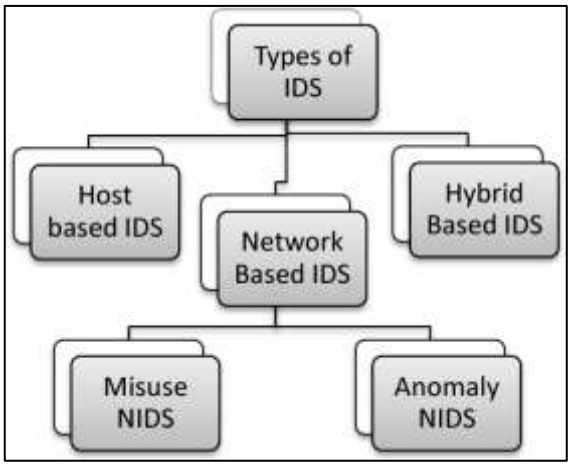


Figure 2.1: An Illustration of A Type of System That Detects Intrusions.

The first purpose of an intrusion detection system, also known as intrusion alerting, is to check the system for any signs that the user may be engaging in hostile activity. The second function, often known as the site security office SSO, is in charge of responding appropriately to any alarms that are triggered [14]. There are three distinct varieties of intrusion detection systems (IDSs), each of which is illustrated in the figure that follows (Figure 2) [15]: detection based on anomalies, detection based on signatures, and detection based on specifications. Following is a more in-depth discussion of each of these several ways of detection. IDSs are capable of being placed into any one of the categories that are outlined in Figure 2 (see there). The present subclassifications of this system are as follows, with the primary emphasis of this study being placed on intrusion detection systems (IDS) that are initiated in response to an anomaly: 1. "Anomaly-based detection" techniques rely on the concept of a normal or legitimate profile that is acquired in the regular settings of the network without any assaults [16]. The term "statistical data" is most usually used to refer to these normal conditions. This profile was acquired when there were no irregularities or anomalies found in the network. The "statistical-based anomaly IDS" takes regular snapshots of the network's statistical features and then compares those properties to a stochastic model of traffic generated under normal conditions [17]. Any divergence from any of the two patterns is viewed as suspicious and may be an indication that an attack is underway. The rule-based will be employed as inputs in this system [18], and heuristics or a UML may occasionally be utilized to explain the behavior of an assault under specific conditions. • In order to differentiate between normal behavior and attacks, the professionals who work on knowledge-based anomaly IDS use various laws in the form of an expert system or apply a definition to the behavior of the connection. In both cases, they focus on the connection's

behavior. [19] The "machine learning-based anomaly IDS" develops a model for the purpose of assisting with the efficiency of intrusion detection. This model is based on the analysis. This model is constructed regardless of whether the recurring patterns are conscious or unconscious. Misuse detection, also known as knowledge-based detection, entails conducting a comparison between the attack signature and the traffic that is now taking place in order to establish whether or not an attack is taking place; if there is no match, then there is no attack recorded. If there is a match, then there is an attack that has been recorded. This method differs from others in that it has a low rate of false alarms and requires the signature to be regularly updated [20]. Additionally, it has a high degree of accuracy. Type that determines the execution of a target program by comparing a memorized specification to a set of criteria in order to detect the program's execution and signal any deviations from the requirements [21]. The comparison is made in order to detect the program's execution and detect any deviations from the requirements. The primary focus of this research is on anomaly-based network intrusion detection systems. This is due to the fact that these systems have the potential to help in the identification of new threats posed by the Internet of Things (IoT). The Network Intrusion Detection System (NIDS) performs an analysis of the data that was collected from the network in order to find attacks that were not previously known.

2.2 THE DEVELOPMENT OF NIDS

The two primary categories that are utilized in this area are supervised algorithms [23] and unsupervised algorithms [23]. Supervised algorithms are ones that make judgments for the user based on known and labeled data that is provided by the user. These methods provide a description of the algorithms that are utilized the most in Machine Learning (ML) and Deep Learning (DL), indicating the efficacy of these algorithms in the development of efficient AI-based NIDS. On the other hand, unsupervised algorithms learn to benefit from unlabeled data by extracting useful information and characteristics. This is done in contrast to supervised algorithms, which learn to gain from labeled data. Unsupervised algorithms, on the other hand, are not monitored by a human in any way, in contrast to supervised algorithms.

2.2.1 Machine Learning Ml

The process of teaching computers how to automatically extract meaningful information from big datasets is referred to as machine learning, which is a subfield of artificial intelligence. Machine learning is also commonly referred to as ML. Once that is accomplished, one can make use of this information to develop predictions, either about newly gathered data or about data that has been collected in the past. There are a substantial number of security challenges that are posed by devices and networks that are connected to the Internet of Things (IoT), and there is a solution that can be found in approaches that are based on intelligence. Investigations have been carried out into modern approaches to machine learning and deep learning, both of which have been applied in relation to the implementation of the internet of things [24]. In response to the significant amount of computing expense that is associated with intrusion detection, a feature selection [25] method that possesses outstanding classification efficacy has been created. This action was taken in order to equalize the costs that were incurred. A database of Hidden Markov Models (HMM) templates and two architectures that are able to detect and trace the progress of assaults in real time was developed by Hidden Markov Models (HMM), which is one of the statistical machine learning approaches described in [26]. The level of success, on the other hand, differs according to the type of assault that is being carried out. 'Cuttlefish algorithm' (CFA) and 'Feature grouping based on linear correlation coefficient' (FGLCC) are both utilized by the DT classifier [27]. Both of these algorithms are acronyms for the same thing. In order to construct a hybrid strategy that included the dimensionality reduction approach, the 'information gain' (IG) method and the principal component analysis (PCA), which is an algorithm for intrusion detection, were applied. [28] This allowed for the development of a hybrid strategy. In order to successfully create a hybrid strategy, this was done in order to accomplish the goal.

2.2.2 Deep Learning Dl

Deep learning is an artificial neural network that simulates some of the properties of machine learning by deploying several hidden layers of the network. Deep learning is frequently abbreviated as DL, which stands for deep learning. DL outperforms ML in a variety of diverse use cases, including object detection, language translation, and speech recognition. These are just some of the use cases that benefit from DL's superior efficiency and accuracy. It is able to generate output on its own depending on the main qualities that it extracts from

the information, thanks to its intelligent design and the fact that it can do so on its own accord. The most recent research is discussed, and then several DL approaches are applied to the problem of NIDS. This is done with the intention of coming up with potential solutions. Because it is able to learn on its own from data and does not require any input from humans or coding, machine learning (ML) is a powerful technology that can be used to automate a broad variety of different occupations. ML is able to teach itself without any assistance from humans. Therefore, machine learning is different from deep learning in that it can read unprocessed input such as text, but the accuracy of deep learning's predictions increases as more data is provided [29]. This gives machine learning its own distinct advantage over deep learning. Deep learning is an essential tool for improving the performance of intrusion detection systems (IDS) [30], as it clarifies the many types of IDS as well as their goals and gives corresponding definitions for IDS. In the area of anomaly detection, the benchmark network intrusion dataset NSL-KDD [31] has been utilized in NIDS to test the efficacy of the sparse auto-encoder and softmax regression. This was done in order to find out which approach is more successful than the others. [32], which claimed that Software-Defined Networks (SDN) may be leveraged to construct highly secured networks while simultaneously boosting the danger of assaults, offered clarity regarding the potential application of DL for anomaly detection systems based on flows. According to the conclusions of a survey [33], experts have found new weaknesses in the security of a variety of components that make up the architecture of the internet of things (IoT). The IDS runs into difficulties while processing huge amounts of data; however, it is feasible to circumvent this issue by making use of deep learning techniques. Deep learning is able to automatically extract features, in contrast to machine learning, which needs to have its features designed before it can be used [34]. Deep learning has the ability to automatically extract features. As a consequence of this, the recent work that has been done with DL and network anomaly detection [35] has brought attention to and addressed the possibility of DL in network traffic analysis. Some intrusion detection systems (IDSs) that used deep learning approaches for security purposes demonstrated the benefits, limitations, and drawbacks of using such a methodology [36–39]. The combination of RNNs with 'Restricted Boltzmann Machines' (RBM) is shown here as an example of a hybrid modeling approach. It's possible that this hybrid approach can spot fake messages [40]. It accomplishes this by making use of a classification task and eliminating the need for feature engineering as a prerequisite. It is

commonly agreed that the automatic execution of sophisticated features in dynamic settings is crucial for network intrusion detection [41]. As a result, a solution to intrusion detection that is based on CNNs has been developed because of this widespread consensus. Because the DNN operates in a black box, it is impossible for the DNN-IDS to be transparent, which is something that is necessary for the development of confidence. Despite this, the DNN-IDS demonstrated significantly higher levels of communicativeness and user trust. During the training process for DNN-IDS [42], the user demonstrated the input properties that are most essential for identifying any kind of intrusion. These attributes were exemplified by the user's actions. A recently developed intrusion detection system has been given the moniker "hierarchical spatial-temporal features-based IDS." This name is also written as "HAST-IDS" in some instances. Deep CNNs were utilized in order to acquire knowledge regarding the low-level spatial characteristics of HAST-IDS, whilst LSTM was utilized in order to acquire knowledge regarding the high-level temporal elements [43]. In this work, we use a variety of deep learning approaches that have been applied to IDS in order to evaluate the accuracy and precision of three models: a 'vanilla deep neural net' (DNN), a 'Self-Taught Learning' (STL) method, and an RNN-based LSTM. Our goal is to determine which of these models provides the most accurate results. The DNN is also referred to as a "vanilla deep neural net" since its architecture is similar to that of a normal neural network. Within the youth network, a new deep learning system referred to as 'Bi-directional LSTM RNN' (BLSTM RNN) [45] was offered as a possible solution for attack detection. The novel structure that has been suggested for the DBN network makes use of the 'Particle Swarm Optimisation' (PSO) approach, which combines a learning component and an adaptive inertia weight. This contributes to making certain that the structure is as effective as it really can be. As a result, the PSO is built with the schooling behavior of fish as its foundation [46]. With the assistance of Deep Learning, the strategy that was suggested was able to detect attacks with a greater degree of accuracy while simultaneously lessening the impact of irrelevant features. The 'Random Forest' (RF) Algorithm was combined with DT Classifiers in order to achieve this [47]. It has been shown that the 'Scale-Hybrid-IDS-Alert Net' (SHIA) DNN hybrid architecture can successfully monitor both network traffic in real time as well as host-level events. This was shown by the fact that it was able to do so. This SHIA has the ability to determine whether or not a potential cyberattack is taking place [48]. The unique architecture of an AE-based deep neural network [49] incorporates many autoencoders into

the convolutional and RNN layers of the network. This helps to elicit linked knowledge of the anticipated linkages between the significant aspects (spatial features) and the timely evolution of those features (temporal features). Because of the usage of 'deep-convolutional neural network-based' and steady traffic features, in addition to nonlinear multiclass classification procedures, a brand new classification strategy known as Deep CNN has been proposed [50, 51].

2.2.3 Different Methods of Ids

In addition, the process of designing NIDS makes use of a variety of different approaches, such as data mining, genetic algorithms, and swarm intelligence. In this section, the findings of research that were conducted on these subjects are broken down in further detail. When data mining strategies are combined with swarm intelligence, highly effective new ways are produced. These approaches can detect and categorize data flow in an effective and efficient manner. The next sentences provide further explanation of these procedures. Authentication and encryption have helped improve the safety of IoT networks, but despite these advancements, these networks are still vulnerable to cyberattacks. Authentication and encryption have helped improve the safety of IoT networks. As a direct result of this, methods of anomaly detection are applied in order to lessen the likelihood of a wide variety of different kinds of attacks [54]. In order to identify properties for the classifier, the firefly approach was applied. Additionally, C4.5 and 'Bayesian Networks' (BN) were made available in order to categorize attacks. An original intrusion detection model that was built on a 'Deep Belief Network' (abbreviated as DBN) was enhanced with the assistance of repeated iterations of a Genetic Algorithm (abbreviated as GA). [56] We used this model in our investigation.

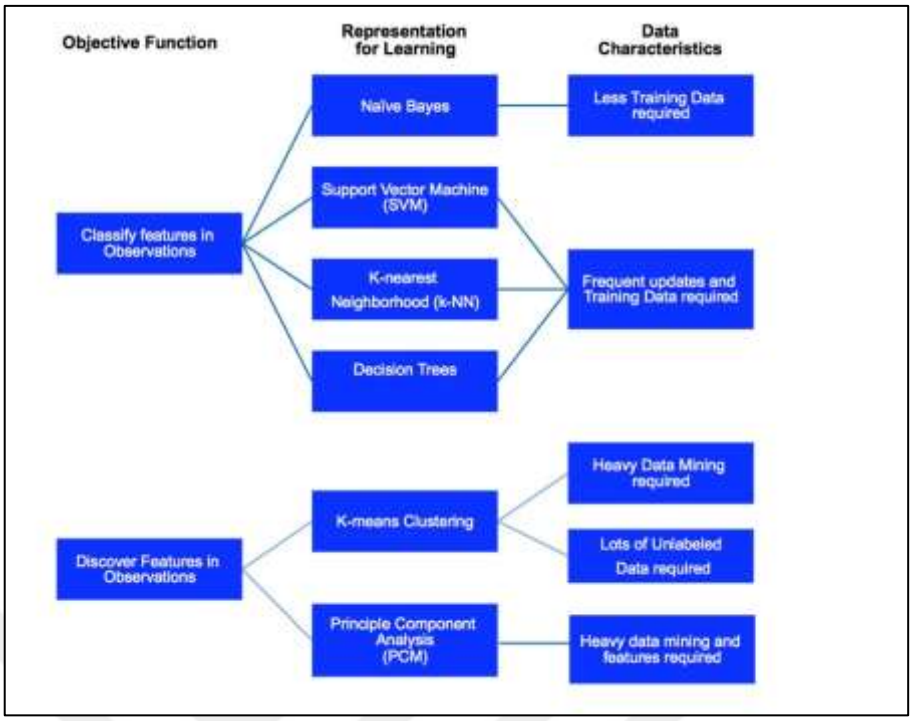


Figure 2.2: A Comprehensive Categorization of Deep Learning in All of Its Aspects[5].

A 'method' that makes use of a 'Artificial Bee Colony' (ABC) has been shown to have a high 'detection rate' (DR) and a low 'false positive rate' (FPR) [58].

2.3 THE DENIAL OF SERVICE AND THE INTERNET OF THINGS

Research on intrusion detection systems (IDS) for the internet of things may be found on this page. We provide a semi-supervised Fuzzy learning model for a distributed threat detection system for the Internet of Things in the article that has the reference number [59]. This model is supplied as a supplementary perk for the purchase. It was recommended by certain intrusion detection systems (IDSs) that learning algorithms based on AdaBoost be used in order to identify possibly harmful actions. This collection of intrusion detection systems, also known as IDSs, was put to use in order to determine the particular botnet attacks that were carried out against the "Domain Name System," the "Hypertext Transfer Protocol," and the "Message Queue Telemetry Transport" [60]. The 'Hybrid IDS' (HIDS) group utilizes a C5 as a classifier to detect well-known incursions and an SVM classifier with One-Class [61] to identify unexpected intrusions. Both of these classifiers are used by the HIDS group. The HIDS group makes use of both of these different types of classifiers. Extensive study shown how essential it is to find a happy medium between performance overheads and accuracy requirements in order to successfully deploy accurate and effective

intrusion detection for the internet of things (IoT) [62]. In order to detect IoT botnets in an efficient manner using the model that was provided, DL was applied. By utilizing a network-based methodology and deep packet checking techniques, we were able to recognize abnormal patterns of behavior in network traffic that were associated with botnet attacks [63]. These patterns were indicative of botnet attacks. The IoT intrusion detection model that was based on the specification offered improved accuracy detection in addition to efficient performance (memory and communication overhead) [64]. This was carried out with the goal of warding against attacks of a kind that have never been seen before. This article presents a novel technique to the detection of intrusions in the Internet of Things that makes use of blockchain technology. The approach was developed by the authors of this article. The strategy relies on a network of regional agents and a centralized hub that plays the role of a coordinator for the information (alerts) that is transmitted by the agents [65]. The identification of physical-layer assaults in Internet of Things (IoT) networks has recently been made possible with the use of a technology that is founded on deep learning. This work is centered on assaults in which the attacker tries to take the identity of the Internet of Things (IoT) device that they are targeting while operating within the RF footprint [66]. When doing an analysis of intrusion detection systems (IDS), there are three aspects that must be taken into consideration: computational cost, privacy, and power utilization [67]. An alternative intrusion detection system (IDS) that makes use of machine learning techniques to recognize potentially harmful behavior in the Internet of Things has been proposed as an alternative. Utilizing the "security as a service" that is supplied by the platform in order to do detection has allowed for the streamlining of protocol collaboration within the Internet of Things [68]. As part of the risk assessment, every potential danger was cataloged and given a careful evaluation before being eliminated from consideration. There is discussion [69] of the benefits as well as the drawbacks associated with the utilization of IoT protocols. The dimensions of the dataset have been significantly reduced by the newly developed model thanks to the application of principle component analysis (PCA), a method of machine learning that is utilized for the detection of intrusions. This has been accomplished by paring down the amount of features from the previously extensive list to a more manageable level. This research makes it abundantly clear that the lack of satisfactory results can be traced back to three primary issues with the current data sets (KDD99 and NSLKDD): a lack of modern attack patterns, a lack of modern scenarios of traffic streams, and the difficulty of

training and testing on distributed sets. [Citation needed] This research makes it abundantly clear that the lack of satisfactory results can be traced back to three primary issues with the current data sets [Citation needed]. As a direct result of this, we came to the conclusion that the best way to investigate these concerns was to make use of the UNSW-NB15 dataset that we had created [71]. The 'variantGRU' is the component that is accountable for the automatic learning of the features of packets' network headers in addition to their payloads. Both the 'E-GRU' and the 'E-BinGRU' are brand new methods for the detection of in-network intrusions that have never been implemented before [72]. Figure 2.3 illustrates a typical installation of an intrusion detection system (IDS) in an internet of things context.

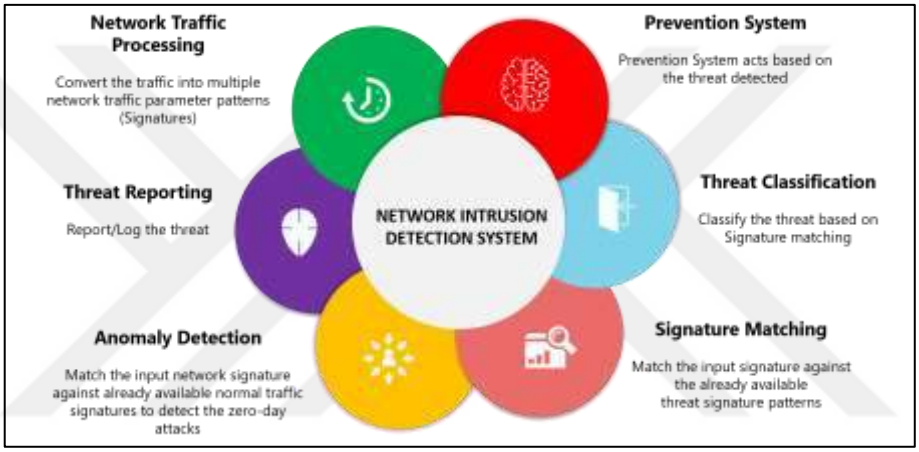


Figure 2.3: Method That is Standardized for the Detection of Intrusions in the Internet of Things (Iot).

2.4 AI AND THE INTERNET OF THINGS FOR NIDS!

As a result of the advancements that have been made in neural network algorithms, DL is today more applicable than it has ever been in the past. Deep learning has shown to be the most adaptable method for identifying new types of attack because of its capacity to extract information at a relatively high level. This is due to its ability to learn from examples. Experiments conducted with a model of deep learning [73] showed that the performance of a distributed attack detection system was superior to that of centralized detection approaches. This was determined by comparing the two systems' respective levels of detection accuracy. This research utilized a data set from the University of California, Irvine (UCI) that was made up of temperature sensors as well as images of personal walkways in order to highlight some of the difficulties that are encountered by smart cities [74]. Emulation was used to illustrate the unique detection system's scalability, and actual network traffic was utilized in

the real world to confirm the concept. The concept was validated as a result of the utilization of actual network traffic. The detection options are provided as a service with the goal of promoting interoperability amongst the numerous protocols that are utilized by the Internet of Things [75]. Artificial intelligence was applied in order to create a solution to the ever-increasing issues that botnet assaults provide to the banking service sector and the financial sector. The 2018 IDS dataset is a real-time dataset (CSE-CIC-IDS2018), which was developed by the Canadian 'Institute of Cyber Security' (CIC) exclusively for the 'Amazon Web Services' (AWS) environment [76]. In order to perform analysis, this system makes use of the dataset. In this research, we explain how to construct a system for detecting attacks by making use of BLSTM RNN, a type of deep learning [77]. Specifically, we focus on the development of the system rather than its actual implementation. The architecture of the Internet of Things was built with the help of the lightweight distributed security solution [78]. Additionally, ML and DL approaches to IoT and Cyber Security were researched, and networks (LSTM and GRU) were tested for each layer of the IDS dataset. These procedures were carried out in their entirety.

2.5 DATASETS

When it comes to deciding the level of success that the detection model will have, one of the most significant factors to consider is the choice of database that will be utilized for information extraction. Figure demonstrates that there are many different kinds of cyber security datasets that may be used for intrusion detection. These datasets can be broken down into seven basic categories, as the figure also demonstrates. 1 [78]: 'internet-connected devices-based' dataset, 'android apps-based' dataset, 'IoT traffic-based' dataset, 'network traffic-based' dataset, 'electrical network-based' dataset, 'internet traffic-based' dataset, and 'virtual private network-based' dataset. 5 and the countless guises in which they can present themselves. The CSE-CIC-IDS 2018 dataset and the Bot-IoT dataset, both of which have only just recently become accessible, will serve as the primary foci of this survey. Both datasets have only just recently been made available. In Table 1, we give an analytical comparison of the findings of research that studied whether or not NIDS could be effective for the Internet of Things. This research evaluated whether or not NIDS could be useful for the Internet of Things. [79] The CICIDS2018 dataset was the product of a collaborative effort between the Communications Security Establishment (CSE) and the Institute for Cybersecurity (CIC). The CICIDS2018, which consists of seven different kinds of assaults,

uses calculations in both forward mode and reverse mode. These include brute force, Heartbleed, infiltration, Web assaults, denial-of-service attacks, distributed denial-of-service attacks, botnets, and distributed denial-of-service. The CIC Flow Meter [81] gathers information from the network traffic that is produced by the CIC in order to incorporate a total of 80 new attributes into the CICDS2017 dataset [80]. This data collection gives you the option of working with PCAP files as well as CSV files. CSV was the foundation upon which AI was developed, and PCAP is the tool that it utilized to summarize newly discovered features [82]. The database gives a representation of regular activities on an Internet of Things network, combined with a number of malicious invasions. Some of the threats that are modeled in the Bot-IoT database include data exfiltration, denial of service, distributed denial of service, keylogging, and operating system scanning. Koroniotis et al. [83] presented the Bot-IoT dataset, which is a new dataset that employs the MQTT protocol as a lightweight communication channel. This was done in order to compare IoT environments with datasets that were already in existence. The Bot-IoT collection has more than 72,000,000 individual records [84].

3. MATERIALS AND METHODS

3.1 INTRODUCTION

Computers have evolved due to the Internet. While on one hand, the potential for good is expanding exponentially, the likelihood of bad actors gaining access is growing. The ever-increasing number of people using the Internet and the management of corporate networks have both made security a big concern. Preventing unwanted access to system and data resources should be a top priority when designing security methods. There is no practical way to eliminate infractions entirely at this time. If these intrusion attempts are detected, then the necessary actions can be done to fix the harm. Intrusion detection is the name of this field of study. The purpose of this paper is to lay out the groundwork for intrusion detection systems and to discuss the many approaches, methodologies, and resources that can be used to build better systems. To properly present the concepts of the intrusion detection system, we will first give an overall vision of the firewall and the IDS. We will explain why intrusion detection systems are necessary. Next, we will present the different kinds of IDS. We will study the important criteria for its construction and finally, we will cite some examples with brief descriptions of the systems already existing in the market.

3.1.1 Security Concepts and Tools

In the years following the publication of Anderson's work in 1980 [Cisco, 03] [Stef, 99], a number of different approaches to intrusion detection have been the focus of subsequent research. There have been many different kinds of cutting-edge detection techniques that have been put into use. Successful outcomes have been achieved as a consequence of a variety of research activities that have been undertaken respectively. Intruders are individuals who get access to a system through the Internet without authorization. This is considered to be a breach of security. In spite of the fact that the individuals in question are already authorized users, this is still the case. When authorized users abuse the powers that they have been given, this is still another type of intrusion. The deployment of intrusion detection technology, which offers an additional layer of protection, is a proven method for enhancing the security of a network. Intrusion detection is becoming increasingly vital for enterprises to defend their systems from potential attacks as the number of devices that are linked to the internet continues to increase. Intrusion detection systems, which are more commonly referred to as IDSs, have developed into a standard piece of equipment that is

required for every company that gives serious consideration to the maintenance of a safe online presence. Even while intrusion detection systems are not a panacea for network security, they do contribute to the current trend toward a defense-in-depth strategy with their aid. This is despite the fact that they are not a silver bullet. In the end, but certainly not least, we will talk about the qualities that set intrusion detection devices apart from other types of equipment. We will also talk about the relevance of these devices, the benefits they provide, and the ways in which they improve network security.

3.1.2 Firewall

This network security device follows a predefined and configured security policy while it works. It is a security tool that helps put the policy into action. The fact that it cannot provide 100% security against harmful traffic is also critical. When comparing firewalls with intrusion detection systems, the main distinction is that firewalls actively prevent assaults, whereas IDS products can only detect and inform about them. Similar to firewalls, intrusion detection systems are not a panacea and will not be able to ward off intrusions on their own.

3.1.3 Features

i. Both internal and external Internet traffic can be controlled by the firewall.
ii. It is actively guided by the chosen security policy and ensures that only genuine traffic is allowed. As an example, a firewall can be set up such that only certain ports, such the one used by the email server, are let through. Here, it's easy to see that the firewall isn't checking the validity of valid traffic.

3.1.4 Limitations

For those unfamiliar with intrusion detection, this is a typical question. Considering the following facts, it is clear that the common belief that a firewall is sufficient to safeguard a network is incorrect:

i. The firewall filters all incoming and outgoing Internet traffic.
ii. Not all dangers originate from outside the network's defenses.
iii. An assault could target the firewall itself.
iv. It skips over stuff that is actually relevant to traffic.
v. Without encryption, the firewall provides no security.

vi. Not all assaults, and notably not all variations of particular attacks, may be blocked by the firewall.
vii. No forensic analysis services are provided by them.

3.2 DEFINITION OF THE INTRUSION DETECTION SYSTEM

An Intrusion Detection System (IDS) is the first thing to know. Devices that keep tabs on a network or a specific host's operations can identify and respond to intrusion attempts [Baud & Mar, 04]. The term "intrusion" refers to any attempt to breach the security of a computer system or network in some way, whether it be by compromising its confidentiality, integrity, or availability or by going beyond what is required by law. Intrusion detection is the act of keeping an eye on all network and computer events and evaluating them for indicators of intrusions.[23]

3.2.1 Several Interesting Reasons for Using IDS

i. To aid in the detection, remediation, and repair of underlying causes by providing information regarding attempted and actual intrusions.
ii. To take action and regulate the high standard of administration and security, particularly in big organizations.
iii. To identify the goals of assaults (iii).
iv. We improve security for identified threats and penalize anyone who would misuse or assault the system in order to forestall problems.
v. To identify intrusions and security breaches that other safeguards have failed to detect.

3.2.2 The Actual Appearance

It is also possible to state that the operation of intrusion detection systems is analogous to that of home security installations. This is still another meaning of the phrase. Just for a moment, try to picture the Authentication Control Device, which is situated at the entry to the door, as being analogous to the Firewall. Due to the fact that he is the owner of the property, the only individuals who are permitted to enter are those who possess his identification and password. Through the utilization of this technology, you are able to move around in a variety of different ways. There is a possibility that the burglar will be able to break the window and enter the house without being found and without being caught. The theft of the identity card and the password is yet another possibility that they could have

achieved. With regard to the two instances that were discussed earlier. Nobody knows who the burglar is or when he attempted to break in, and there is no record of any damage or how it was taken. Additionally, there is no record of how it was taken. In addition to this, there is no evidence of any previous injury. As we compare it to burglar alarms, let's look at the similarities and contrasts between the two tools. In the event that an unauthorized person makes an attempt to enter the residence, the burglar alarms are equipped with the capability to sound. The system has the power to notify the owner or guard of the house, notify any individuals related with the burglary, record the time of the burglary, and maybe provide further information in the event that the thief is successful in getting admission. The architecture of the security policies and the processing capabilities of the intruder alarm system are both factors that will determine whether or not this functionality is available. The usage of intruder alarms, which provide complete information and clues that may be utilized for forensic investigation and the prosecution of perpetrators, is therefore a means by which the impact of a break-in can be reduced. When it comes to the identification of intrusions, functional processes that are same can help decrease risk, offer support to forensic investigations, and contribute to the reporting of management events. In the following lines, a compilation of many misconceptions that are frequently held about intrusion detection systems will be provided.

3.2.3 Features

i. It portrays the state of the system and network in a clear light.
ii. Intrusion detection systems are able to identify threat actions and notify the appropriate parties.
iii. It strengthens the existing security architecture while making it more flexible.
iv. version iv. It has the capability to store sessions in a format that is specified.
v. With the help of session logs, event correlation, and graphical interfaces, it offers and improves the forensic investigation process.
vi. It provides real-time analysis when monitoring the network or systems.
vii. It has the ability to notify security personnel of certain patterns.
viii. Active responses are possible, such as the banning of IP addresses and the termination of connections.
ix. Management can get effective reports from it.

x. First and foremost, intrusion detection systems (IDS) offer recommendations that may be useful when crafting a company's security policy.

3.2.4 Limitations

i. It's not proactive or optimistic; it's just active.
ii. He is helpless to stop the assault.
iii. It's not mechanized; managing them calls for a lot of people.
iv. It is not able to provide total security for assets. It's merely an additional layer. There is no magic bullet.
v. It won't fix problems caused by flawed network protocols.
vi. It is not capable of preventing every kind of assault. There are boundaries he must adhere to.
vii. It isn't up to the task of handling extremely fast and massive amounts of internet traffic.

3.3 CLASSIFICATION OF IDS

Intrusion detection techniques will be presented here in a taxonomic fashion. Different intrusion detection systems (IDS) use different methods for monitoring and analysis. Every strategy has its own set of benefits and drawbacks. An IDS model is applicable to every strategy.

i. The location of IDS
ii. Detection methods
iii. Types of response
iv. Frequency of use

We will first study host-based intrusion detection, then application-based, before focusing on Network IDS (NIDS).

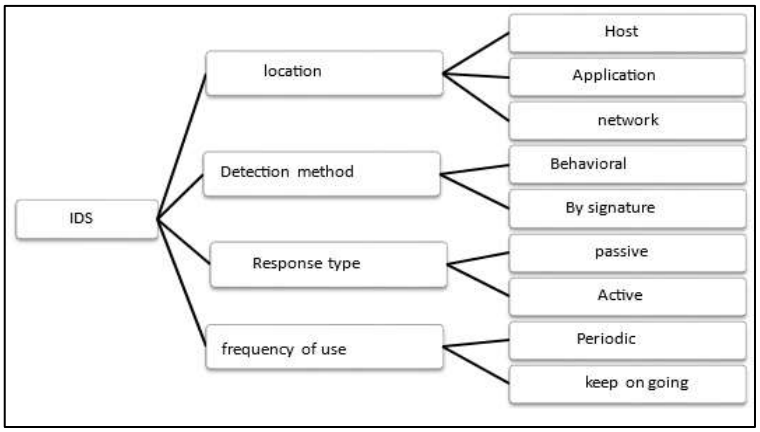


Figure 3.1: The Different Types of IDS.

3.4 IDS LOCATION

Sorting intrusion detection systems (IDSs) according to the physical location of the data source is the de facto standard practice. Primary data comes from things like operating system logs, crucial files, and packet captures from local or segmented networks.

3.4.1 Host-based Intrusion Detection

When it comes to intrusion detection systems, host-based intrusion detection systems (HIDS) are the ones that concentrate entirely on the data that is associated with the host. Instead of being responsible for the management of network traffic, they are in charge of monitoring "only" host activity when it comes to monitoring network traffic. With a high degree of precision and reliability, intrusion detection systems, which are more commonly referred to as IDS, are able to identify the processes or individuals who are responsible for an assault. The utilization of these technologies allows for the possibility of making a forecast regarding the outcome. On the other hand, high-level intrusion detection systems (HIDS) provide for direct access to and administration of the data files and processes that are typically the targets of attacks. This is in contrast to network intrusion detection systems (NIDS), which are designed to detect network intrusions. The examination of audit trails generated by operating systems is a frequent approach that these intrusion detection systems employ in order to gather information regarding the activities carried out by machines. One of the many benefits that are linked with it is the ability to view the effects of an attack in real time, which permits quicker responses. This is only one of the many advantages that are associated with it. It was able to improve the system in accordance with the behaviors that were noticed on the host as a result of the substantial amount of data that was evaluated. This

was made feasible as a result of the process of analyzing the information. Moreover, NIDS and HIDS are complementary to one another in terms of their capabilities. In point of fact, they improve the identification of attacks known as "Trojan Horse" attacks, which are notoriously difficult for network intrusion detection systems (NIDSs) to recognize. The fact that they are a part of encrypted communication makes it simpler for a high-fidelity intrusion detection system (HIDS) to recognize threats that a network intrusion detection system (NIDS) could miss. There are, on the other hand, a few drawbacks connected to this particular kind of intrusion detection system (IDS), which are a direct consequence of the advantages that it provides. An example of this would be the fact that this specific kind of intrusion detection system (IDS) is particularly vulnerable to denial-of-service (DoS) assaults, which can lead to a considerable increase in the size of files. Additionally, the security manager is having a difficult time because there are excessively large alert report files that need to be analyzed. This is further complicating the situation. There exists a possibility that the sizes of files could potentially approach the region of megabytes. It is because of the enormous amounts of data that they need to analyze that they have a high CPU usage rate. This high utilization rate does have the potential to occasionally have an influence on the performance of the machine that is hosting them. HIDS, or high-security intrusion detection systems, are often installed on machines that are vulnerable to intrusions and that carry valuable company data. These machines are typically the target of investigations. It is possible that hidden files could provide an additional degree of security for servers that are responsible for hosting websites or applications.

3.4.2 Application-based Intrusion Detection

AIDS, which is an application-based intrusion detection system, is one sort of host intrusion detection system. They monitor the method in which a user interacts with a software by adding log files that provide additional information regarding the activities of an application. This allows them to do so. When one places themselves in the middle of a user and a piece of software, it is much simpler to overlook any behavior that looks to be suspicious. The region in which an ABIDS is most effective is the one in which the interaction that takes place between the user and the application that is being monitored takes place. In addition to being able to recognize and prevent specific user requests, this intrusion detection system is also capable of monitoring all transactions that take place between users and apps. This is one of the advantages of the system. The analysis of the data is also more complete and

accurate as a result of the fact that the data is decoded within a context that is previously known. The protection that is given, on the other hand, is not as robust, particularly against attacks of the "Trojan Horse" sort. This is due to the fact that this intrusion detection system does not function at the kernel level. As an additional point of interest, in contrast to system audit trails, the log files that are produced by this kind of intrusion detection system are totally accessible to malicious actors. This particular form of intrusion detection system (IDS) is a great instrument for monitoring the actions of a highly sensitive application. It is often deployed in conjunction with a high-level intrusion detection system (HIDS) in order to achieve optimal results. It will be important to handle the management of the CPU utilization rate of the intrusion detection system (IDS) in order to avoid any unfavorable effects on the performance of the machine.

3.4.3 Network Intrusion Detection (NIDS)

When it comes to network intrusion detection, the most widely used solutions in the commercial sector are those that are currently commercially accessible. As a result of the fact that these systems listen on the network segment or switch, as well as capture and analyze data, they are able to detect intrusions. Two of the approaches that they employ in order to achieve this objective include protocol decoding and comparing multiple packets to a database of known attack signatures.

It is possible for the Network Intrusion Detection System (NIDS) to swiftly block connections and generate notifications in the event that it discovers suspicious activity. When it comes to the operation of goods, the "promiscuous mode" is the mode that is generally recognized. During this time, they keep a close eye on each and every packet that is received from the local segment.

The placement of sensors at key points throughout the network is a component of network intrusion detection systems (NIDS), which are designed to meet the requirements of their intended purpose. The activation of these sensors will result in the transmission of warnings in the event that they identify an attack from occurring. A secure console is the recipient of these notifications, which it then analyzes and, depending on the results of its investigation, may even take some kind of action. The placement of this console on a separate network that enables direct communication between the sensors and the console is a common practice that is often adopted by a large number of people.

a. The location of the sensors

The network sensors are configured to operate in stealth mode, allowing them to remain undetected by other machines. This is accomplished by configuring their network card in "promiscuous" mode, which reads all traffic, without configuring an IP address.

The majority of sensors come with dual network cards; one is for network stealth and the other is for connecting to the control panel.

Due to their invisibility on the network, it is much more difficult to attack them and know that an IDS is used on that network.

b. The location of the sensors

It is possible to place the sensors in different locations, depending on what you want to observe. The sensors can be placed before or after the firewall, or in a sensitive area that we want to specially protect.

If the sensors are located after a firewall, it is easier for them to tell if the firewall was misconfigured or to know if an attack came through that firewall. Their mission is to detect intrusions which have not been stopped by the latter. This is a common use of a NIDS.

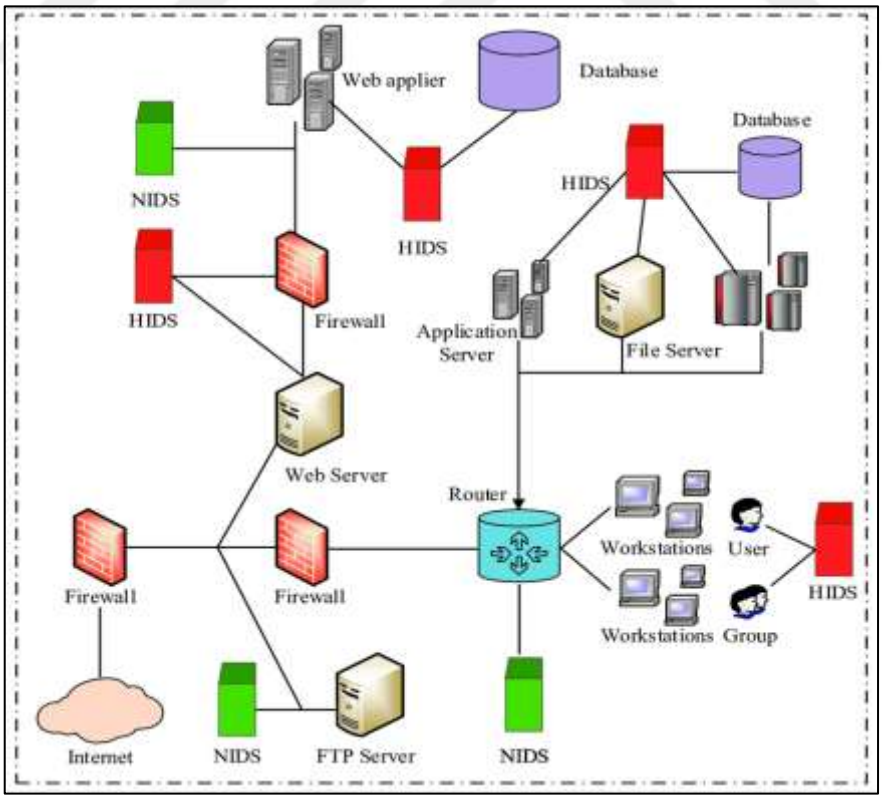


Figure 3.2: Installation of N-IDS.

It is also possible to place a sensor outside the firewall (before the firewall). The advantage of this position is that the sensor can thus receive and analyze all Internet traffic. If we place the sensor here, it is not certain that all attacks will be filtered and detected. However, this location is the favorite of many experts because it offers the advantage of writing to logs and analyzing attacks (towards the firewall...), so the administrator sees what he must modify in the firewall configuration. The sensors placed outside the firewall are used to detect all attacks directed towards the network, their task here is therefore more to control the operation and configuration of the firewall than to ensure protection against all detected intrusions (some being processed by the firewall). It is also possible to place one sensor and another after the firewall. In fact, this variant combines the two cases mentioned above. But it is very dangerous if you configure the sensors and/or the firewall incorrectly, in fact you cannot simply add the advantages of the two previous cases to this variant. IDS sensors are sometimes located at the entrance to particularly sensitive network areas (server parks, confidential data, etc.), so as to monitor all traffic towards this area. A few of the benefits of network intrusion detection systems include the fact that they simplify the process of filtering traffic, that they make it simpler to recognize scan assaults, and that sensors can be monitored in a discrete manner because they simply monitor the traffic. These are just a few of the advantages. Network intrusion detection systems, commonly known as NIDS, are typically passive devices that listen on the network connection without interfering with the normal operation of a network. This is the case the majority of the time. Using several network intrusion detection systems (NIDS) that are properly positioned, it is possible to operate a big network. Despite the fact that they are widely utilized and perform an essential role, network intrusion detection systems (NIDS) continue to have a number of shortcomings. False negatives, which are attacks that are not detected by the system, are possible, without a doubt, and there is a significant potential of them occurring. If a network intrusion detection system (NIDS) has trouble processing packet fragments, then it is probable that an intrusion detection system (IDS) will not function properly. They are unable to deal with the high speeds and the large amount of traffic that they regularly face. When it comes to interpreting material that has been encrypted, they are completely incapable. On the other hand, in contrast to host-based intrusion detection systems, these technologies are unable to determine the repercussions of an assault. Even though we make it a point to differentiate between HIDS and NIDS, the distinction between the two is becoming increasingly difficult

to discern. HIDS is rapidly embracing the fundamental characteristics of NIDS, which is the reason for this.

3.4.4 Detection Modes

Two techniques of detection are available: signature recognition and anomaly detection. Note that the majority of commercially available intrusion detection systems use signature recognition as their primary mode of operation. On the other hand, most recent offerings improve intrusion detection by combining the two approaches.

3.4.5 Anomaly Detection

It entails looking for deviations from a "usual traffic" profile. During the learning phase of any implementation, the intrusion detection system will "discover" how the monitored parts typically operate. As a result, they might disclose deviations from the standard procedure. In the case of HIDS, this type of detection can be based on information such as CPU usage rate, disk activity, connection times or use of certain files (office hours, etc.). Let us cite, as an example, a project was sponsored by DARPA (in 98 and 99), in collaboration with the Lincoln laboratory at MIT and which was one of the most ambitious projects [Lippmann & al, 00]. The goal was to provide a meaningful set of training data, including background traffic and intrusive activities (i.e., intrusive traffic or system events caused by attacks). Background traffic was inferred from statistical data collected on the Air Force base network while attacks were generated by specially created scripts, but also by scripts collected through specialized sites and mailing lists. The data collected concerned both HIDS (e.g. audit data from Solaris stations, disk dump from UNIX machines and BSM “Basic Security Monitoring”) and NIDS. The DARPA'98 test set used around 300 attacks (classified into 38 attack types), while DARPA'99 used around 50 attack types.

a. Advantages:

i. Without knowing the specifics, intrusion detection systems based on anomaly detection can identify symptoms of both known and unknown attacks since they detect unexpected activity.
ii. This approach makes it possible to produce information useful for defining signatures for signature-based intrusion detection systems.

b. The Inconvenients:

i. The downside of this approach is the large number of false alarms due to the unpredictable behavior of network users.

ii. It often requires the long-term history of recorded events in order to characterize normal patterns of behavior. Systems based on this approach must have some intelligence for machine learning reasons.

3.4.6 Signature Recognition

This method involves looking for telltale signs of recognized assaults in the monitored element's activities. This intrusion detection system is reactive in nature; it can only identify threats for which it has a signature. Because of this, it needs to be updated often.

In addition, the accuracy of the signature base is crucial to the performance of this detection system. Because of this, hackers employ what are known as "evasion" techniques, which include masking the attacks, to get past these measures. These methods change the attack signatures such that the IDS can't identify them.

It is possible to develop more generic signatures, which make it possible to detect variants of the same attack, but this requires good knowledge of the attacks and the network, in order to stop the variants of an attack and not hinder normal network traffic A signature makes it possible to define the characteristics of an attack, at the packet level (up to TCP or UDP) or at the protocol level (HTTP, FTP, etc.).

At the packet level, the IDS will analyze the different parameters of all packets passing through and compare them with known attack signatures. At the protocol level, the IDS will check at the protocol level if the commands sent are correct or do not contain an attack. This functionality has mainly been developed for HTTP currently.

This example seemed very educational to us on the role of signatures and their development. You should know that IDS vendor sites offer signature updates based on new attacks identified.

However, the more different signatures there are to test, the longer the processing time will be, the use of more elaborate signatures can therefore provide an appreciable time saving.

However, a poorly crafted signature can ignore real attacks or identify normal traffic as an attack. It is therefore advisable to handle the development of signatures with caution, and with good knowledge of the monitored network and existing attacks.

a. Advantages

i. Identifies assaults with high accuracy and low false alarm rate.

ii. Is able to accurately and rapidly identify the tool or attack technique in question.

iii. Security managers can use this to prioritize efforts to fix vulnerabilities.

b. The Inconvenients

i. Can only detect known attacks, whose signatures are introduced into the system, so the detection system must be constantly updated with the signatures of new attacks.

ii. Many systems taking this approach are designed to employ a limited number of signatures that can be defined, which prevents them from detecting variants of these attacks. Once an attack is detected, an IDS has the choice between several types of responses, which we will now detail.

3.4.7 Response Types

It all depends on the IDS, but there are two possible answers. All IDS have passive response capabilities, and active response is generally used.

a. Active Response

The active response, on the contrary, aims to stop an attack at the moment of its detection. There are two techniques available for this: reconfiguring the firewall and interrupting a TCP connection. Reconfiguring the firewall makes it possible to block malicious traffic at the firewall level, by closing the port used or prohibiting the attacker's address. This functionality depends on the firewall model used, not all models allow reconfiguration by an IDS. Furthermore, this reconfiguration can only be done according to the capabilities of the firewall. The IDS can also interrupt a session established between an attacker and their target machine, so as to prevent the transfer of data or modification of the attacked system. To do this, the IDS sends a TCP reset packet to both ends of the connection (target and attacker). A TCP reset packet has the RST flag set, which indicates a disconnection from the other end of the connection. With each end on the receiving end, the target and attacker think the other end has disconnected and the attack is interrupted. In the case of an active response, you must be sure that the traffic detected as malicious really is, otherwise normal users will be disconnected. Typically, IDSs do not actively respond to all alerts. They only respond to alerts when they are positively certified as attacks. The analysis of the alert files generated is therefore essential to analyze all the attacks detected.

b. Passive Response

An intrusion detection system's passive response is to keep track of any incursions it finds in a log file, which the security manager can then review. You can record a whole malicious connection with some intrusion detection systems. While this does help fix security holes that allowed previously recorded assaults to happen, it cannot stop attacks from happening in the first place.

c. Frequency of Use

The usage characteristic is the last point of differentiation, and there is only one way to implement it: ongoing (web-based) or periodic (offline).

d. Continued Use

Whenever an attack happens, it is immediately detected. Scanning the surroundings to obtain information is usually the first step for an attacker before attacking a network. Thus, by observing this, one can identify an impending attack in advance, allowing for a prompt response.

e. Periodic Use

Having a more dependable defense in terms of calculation time is preferable than for the initial use, especially since this method is executed frequently and we only observe the attack result. There is no way to identify the attack in its early stages to stop it, unlike online IDS. Damage increases as attack detection times increase.

3.5 VALUATION CRITERIA

Despite intrusion detection systems' pervasiveness as a defensive tool in modern computer systems, no thorough and rigorous scientific methodology has been developed to evaluate their efficacy. [25]. To evaluate IDS effectiveness, we provide a set of partial metrics below.

3.5.1 Classification Errors

The power of a detector can be affected, to varying degrees, by several kinds of mistakes. When security policies are violated, an alarm is raised, which is known as a true positive. In a true negative, neither an alert nor any other abnormality occurs. When an alarm goes off even when nothing out of the ordinary is occurring, this is called a false positive. When anything out of the ordinary occurs but the alarm doesn't sound, this is called a false negative. A false positive could seem to be less of a threat than a false negative at first look. [26]

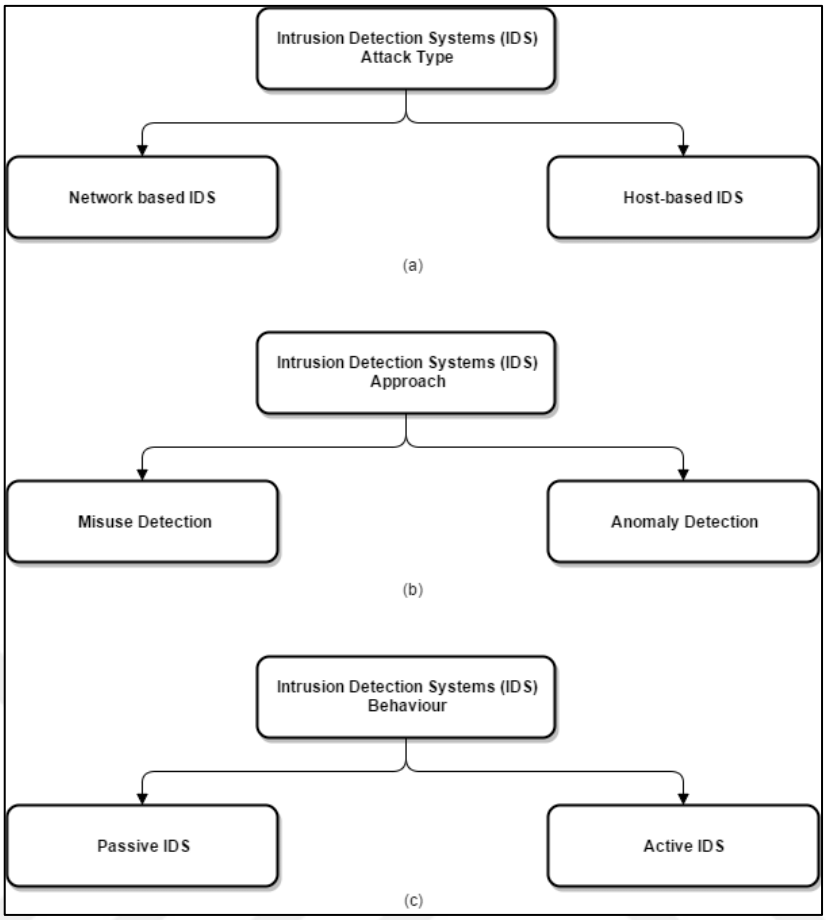


Figure 3.3: The Different Types of IDS [8].

3.6 THE ATTACKS

3.6.1 The Different Stages of an Attack:

For the great majority of attacks, which range from the most fundamental to the most complex sort of attack, there is a consistent pattern that can be observed:

The identification of the target is the initial step in any coordinated attack. This step permits the gathering of as much public information as possible concerning the target, which is necessary for the attack to be successful. The example of visiting DNS servers is likely to be one of the examples that we will use, in addition to the other examples that we will use.

It is imperative that you make certain that you get all of the facts that you require concerning your target when you are in the process of scanning. It is possible to retrieve accessible services, IP addresses that are currently being utilized, and a broad variety of topological characteristics (such as operating system, service versions, subnet, firewall rules, and so on) by utilizing this method. All of these things can be obtained.

It is possible to make use of the data that has been obtained during this phase in order to exploit the vulnerabilities that have been discovered in the components of the target environment. This step is known as the actualization phase. It makes no difference whether the components in question constitute operating systems, services and applications, or protocols; these vulnerabilities are always present.

The next phase will demand the attacker to comprehend the reasons behind the activities that he took in the previous steps. This will be a must for the attacker to be successful. The end goal is to be able to root a system and do whatever you want with it, including inspecting it, extracting data, installing backdoors, removing traces, and so on. Rooting a system gives you the ability to do all of these things. Having the ability to root a system grants you the ability to perform all of these tasks.

3.6.2 Classification of Attacks

An assortment of attack types have been advanced, each with its own set of proposed criteria for categorization. These include attack severity, vulnerability exploitation, and methodology. Following is a presentation of the five types of assaults recognized by the Massachusetts Institute of Technology (MIT) [MIT, 99], the most popular categorization in the literature.

a. Denial of Service DOS

A denial of service attack occurs when an attacker overwhelms a system's processing or storage resources, making them inaccessible or unable to handle requests from users who are actually authorized to use them. Denial of service (DOS) assaults come in several forms [Kendall, 99]. Attacks such as "smurf" are an excellent example of how to legitimately exploit gadgets. Some "ping of death" attacks cause the target machine's TCP/IP stack to get confused, and the machine will subsequently attempt to reconstruct these packets. Still more "apache2, dos, syslogd" exploit network vulnerabilities.

b. User to Administrator Attacks (U2R)

Are a form of exploit in which the attacker first obtains access to a normal user account on the system (obtained by password sniffing, for example, by utilizing social engineering), and then makes an effort to exploit the vulnerability on this system in order to obtain administrator access. This type of exploit is known as a "password sniffing" exploit.

Kendall (99) asserts that there are a great deal of different kinds of attacks that can be

successfully carried out by utilizing U2R. A "buffer overflow" attack is the type of attack that occurs the most frequently across the board. An example of this type of attack is when a piece of software transfers a considerable amount of data into a static buffer area without first deciding whether or not the size of the buffer area is sufficient. This ultimately leads to an overflow, which is the result of the attack. On top of the instructions that were supposed to be carried out in the system stack, the data that was accidentally leaked is stored. When there is an excessive amount of data, this is what occurs. In order for an adversary to achieve their goals, it is required for them to carefully modify data that overflows into the stack and then force the operating system to carry out a few commands that are completely random. This is important for the adversary to be successful. U2R attacks can be broken down into a few different types, one of which is the "load module" attack. This type of attack takes advantage of programs that offer information about the environment in which they are currently functioning. Another type of U2R attack takes use of software that does not manage temporary files in the appropriate manner. This type of attack is known as a side-channel assault. Garfinkel et al. (96) state that there are vulnerabilities in U2R attacks that take use of exploitable concurrency circumstances. These flaws can be exploited by malware. These conditions have the ability to have an effect on the execution of a single program, two programs, or even more programs that are taking place at the same time. In spite of the fact that controlled programming has the potential to solve all of these problems, they are nonetheless present in each and every version of UNIX and Windows that is now available.

3.7 THE SECURITY AUDIT

When it comes to system maintenance and security, auditing is widely regarded as one of the most essential components [25]. Uncovering the activities that took place on a system is the objective of an audit, which is accomplished by the examination of the activity log created by that system. The term "audit trail" refers to a detailed record of actions that have been taken in the past, including the timings, authors, and consequences of such actions. These logs, which are also known as audit files, are the locations where audit reports are stored in a binary format. The gathering of information concerning the consumption of the system's resources is something that must be done during the course of an audit. The audit data can be utilized to determine the source of incidents that occur with regard to system security. With the help of this information, we are able to ascertain who is to blame for every terrible occurrence that takes place on a host. The identification and authentication processes

are the two security functions that make up the foundation of an efficient audit. If the authentication process is successful, two distinct identities are established when a user logs in: an audit user ID that cannot be changed, and a one-of-a-kind audit session ID that is connected to the user's behavior. Both of these identities are completely separate from one another. The user is the only one who can use either of these identities. During the time that the user is logged in, the audit session ID will be propagated to any processes that are started. Auditing certain actions, such as starting up and shutting down the system, is always left as the default configuration without being changed.

The audit service is accountable for carrying out the following responsibilities, which are as follows:

3.7.1 Auditing and Security

Auditing can uncover suspicious or unusual patterns of system usage, which can help detect potential security infractions. Additionally, auditing allows for the tracking of questionable actions, which can be traced to an individual user and serves as a deterrent. Users are less inclined to engage in malevolent behavior when they are aware that their actions are being monitored.

Mechanisms to govern actions prior to the start of system processes or user processes are necessary for the protection of any computer system, but especially a network-based system. Tools to monitor activity in real-time are essential for security. For security reasons, it is sometimes necessary to notify actions after they have occurred.

Unauthorized hackers can still get into the system despite auditing's best efforts to monitor activities and generate reports.

On the other hand, reports might be generated by the audit service showing that a certain user carried out specific actions on a specific day. Possible user identification in the audit repor.

3.7.2 How the Audit Works

a. Following the occurrence of specified events, auditing creates audit records. The most typical events that trigger audit records are:

i. system startup and shutdown;

ii. login and logout; and

iii. process creation and thread creation and destruction.

Operations involving object creation, deletion, or renaming v. Operations involving object identification and authentication.

Process or user-level permission changes vii. Package installations and other administrative tasks viii. Applications tailored to certain websites.

b. There are three main places where audit records are created:

c. Through an application ii. After an audit iii. After a process system call

d. An audit record is the final shape that pertinent event data takes after collection. The incident, its cause, the time it occurred, and any other pertinent details are detailed in every audit document.

3.7.3 Audit Features

During the initial connection, the audit is defined according to characteristics, those important for our project are the following:

i. Audit session ID: When you log in, you'll be assigned an audit session ID. This identifier is passed down to all subsequent threads.

ii. An audit user ID is assigned to a process when a user logs in. This ID cannot be changed. Any subsequent processes launched by the user's original process will also have this ID. Enforcing accountability is made easier with the audit user ID. The audit user ID does not change regardless of the user's role. Every audit report includes an audit user ID, which allows for easy action tracing to the login user.

3.8 CLASSIFICATION ALGORITHMS

Data mining offers a very wide variety of data mining techniques and algorithms. These algorithms have diverse origins; Some come from statistics (regression, etc.), others from artificial intelligence (neural networks, decision trees, etc.), and some still draw inspiration from the theory of evolution (genetic algorithms, etc.). This combination of technologies facilitates the resolution, understanding, modeling and anticipation of problems.

Data mining is a set of complementary techniques dedicated to different tasks. These techniques are shared, mainly, between automatic classification (supervised and unsupervised) and association search.

Supervised classification is a task widely applied in everyday life. Indeed, there are a multitude of problems that fall within this framework, including handwritten character recognition, speech recognition, text categorization, spam detection, medical diagnosis

assistance, Bioinformatics, etc. etc. Unlike unsupervised classification where groups of objects are discovered a posteriori, supervised classification techniques apply when we want to attach a new object (observation) to a class which is chosen from a set of known classes. This step generally follows the clustering step. Another approach which is no less interesting is reasoning based on cases. These algorithms do not seek to calculate the model but to find, for the object to be classified, one or more similar cases already resolved to deduce the class; the kNN algorithm adopts this principle. In the following, we will describe and detail the principle of the KNN classification algorithm.

3.8.1 The KNN Algorithm

A classification-only approach that has found a specific application in the field of estimating concerns is the KNearest Neighbor (PPV Plus Proches Voisins) algorithm. This algorithm is an example of a classification-only method. As an illustration of a line of reasoning that is predicated on an examination of examples, the PPV technique is given as an example. One of the essential premises is that decisions can be made by retrieving from memory prior outcomes that are comparable to the situation that is presently being analyzed. This is one of the fundamental premises.

In contrast to other classification methods, such as decision trees, neural networks, and evolutionary algorithms, this method does not involve a learning stage that involves the construction of a model from a training sample. Rather, this method does not entail such a step. The training set, a distance function, and a class selection function are the components that make up the model. The class selection function takes into account the classes of the neighbors that are located in the closest proximity to the model. Take into consideration the following as an example of a general algorithm for the categorization of PPV-based examples:

3.8.2 Definition of Distance

The choice of distance is essential to the proper functioning of the method. We chose the cosine measure: which defines the similarity between the vectors.

$$\cos(V1, V2) = \frac{V_{1J} \cdot V_{2J}}{||V1||. ||V2||} \quad (3.1)$$

We consider a system call sequence composed of n items as an n-dimensional vector. The cosine makes the angle between vectors normalized by their length.

3.8.3 Class Selection

The idea of the method is to search for cases similar to the case to be solved and to use the decisions of close cases already solved to choose a decision. The simplest method is to search for the closest case and make the same decision. This is the 1-PPV (1-NN) nearest neighbor method. If this method can provide good results on simple problems for which the points (records) are well distributed in dense groups of records of the same class, as a general rule, it is necessary to consider a larger number of neighbors to obtain good results. results.

3.9 CRITICISMS OF THE METHOD

3.9.1 No Learning

It is the sample that constitutes the model. The introduction of new data makes it possible to improve the quality of the method without requiring the reconstruction of a model. This is a major difference from methods such as decision trees and neural networks.

3.9.2 Clarity of Results

Although the method does not produce an explicit rule, the class assigned to an example can be explained by showing the nearest neighbors that led to that choice.

3.9.3 Any Type of Data

The method can be applied as soon as it is possible to define a distance on the fields. However, it is possible to define distances on complex fields such as geographic information, texts, images, sound. This is sometimes a criterion for choosing the PPV method because other methods have difficulty processing complex data. It can also be noted that the method is robust to noise.

3.9.4 Number of Attributes

The method makes it possible to deal with problems with a large number of attributes. But, the greater the number of attributes, the greater the number of examples must be. Indeed, for the notion of proximity to be relevant, the examples must cover the space well and be sufficiently close to each other. If the number of relevant attributes is low relative to the total number of attributes, the method will give poor results because the proximity on the relevant attributes will be drowned out by the distances on the irrelevant attributes. It is therefore sometimes useful to first select the relevant attributes.

3.9.5 Classification Time

If the method does not require training, all calculations must be performed during classification. This is the consideration to pay compared to methods which require learning (possibly long) but which are fast in classification (the model is created, it is enough to apply it to the example to be classified). Some methods make it possible to reduce the sample size by retaining only the examples relevant to the PPV method, but in any case a sufficiently large number of examples is required relative to the number of attributes.

3.9.6 Store Model

The model is the sample, so significant memory space is required to store it as well as fast access methods to speed up the calculations.

3.9.7 Distance and Number of Neighbors

The performance of the method depends on the choice of the distance, the number of neighbors and the mode of combining the neighbors' responses. Generally, single distances work well. If simple distances do not work for any value of k, we must consider changing the distance, or changing the method! our work involves implementing an intrusion detection system that applies these algorithms to a sufficient amount of normal or abnormal audit data to generate a classifier capable of labeling it as belonging to the normal or abnormal category of new audit data.

3.10 IMPLEMENTATION OF THE METHOD FOR INTRUSION DETECTION

Sometimes we need to classify the data into two groups, for example normal group and abnormal group. Among the learning mechanisms, we have the behavioral approach which proposes to describe the “normal” or “usual” behavior of a user. Any deviation from this normal behavior is considered an intrusion. In this project, data is represented by vectors of system calls.

3.11 INTRUSION DETECTION BY SIGNATURE

The attack signature detection approach is based on the creation of a knowledge base containing a description of known attacks. The intrusion detection module then analyzes the logged activity traces in search of different events corresponding to the description of the attacks in the knowledge base. These events represent the signature of the attack.

a. Learning Data

i. The attack signatures for the creation of the IDS database by signature were selected as follows:

ii. The experiment carried out considered two variants, the test attacks are of the same signatures saved in the database, the second variant proposes new attacks; Test data also includes normal processes. The following table summarizes the data involved in this experiment:

Table 3.1: Data Selected for IDS by Signature.

	Signatures	T is S	
		attacks	**Normal**
Experiment 1	81 instances	81	383
Experiment 2	61 instances	81	383

Two parameters can be manipulated to adjust the classification: the number of neighbors k, and the decision threshold if the process is normal or not.

The similarity is compared to the threshold, if it is greater then the IDS decides that the process is abnormal, and vice versa.

4. PROPOSED METHOD

The main idea of this work is to implement and analyze two distributed machine learning systems for detecting network intrusions. The first is the system with a parameter server communication topology and the second, with a point-to-point communication topology. Different machine learning models are used along with different model aggregation methods. The models tested are the neural network with federated average aggregation and decision trees with average aggregation. The data set used in this work corresponds to the access traffic of 373 fixed broadband users of a large telecommunications operator in the South Zone of the city of Rio de Janeiro The analyzed database was created from the capture of raw packets with the tcpdump tool, containing real IP (Internet Protocol) traffic information from residential users who were treated by an intrusion detection system (Intrusion Detection System - IDS) of network, and later summarized in network flow using the flowtbag tool. The dataset initially had a total of 590 attack flow classes and one normal flow class. In this work, the data set was treated to have only two classes, class 0 referring to normal flow, and class 1 corresponding to all types of attacks detected and recorded in the data collection. This treatment was carried out because it allows the models to achieve better performance in predicting flows due to the fact that binary classification reduces the complexity of prediction calculations and reduces the chances of overfitting models as the classes have greater generalization of data characteristics. There is a very important issue to be addressed in a real application that refers to the diversity of participants' local data. This issue refers to the fact that the heterogeneity of devices and the statistical heterogeneity of the data generate a set of data with different statistical distributions among the system participants. The features in a non-IID (non Independent and Identically Distributed) dataset are dependent and are not identically distributed, that is, each feature has distinct distribution probability functions and there may be statistical dependence between different features. This fact is important, as it directly determines the generalization capacity of machine learning models trained on non-IID data, as these models are strongly conditioned by the characteristics of these data.

4.1 INTRUSION DETECTION SYSTEM WITH COMMUNICATION TOPOLOGY

This section presents the implementation of Intrusion Detection System with Parameter

Server Communication Topology. This system has two main components, the client and the parameter server. The client is the device that has the data and the local model initially trained with only the local data. The parameter server is the device that aggregates clients' local models and distributes the model resulting from the aggregation among all clients. In a real application, customers are carefully selected to participate in model training. This is due to the fact that malicious clients can train the model with false data or profiles and thus bias the model, thus affecting the quality of the classification. However, this work considers that all clients are honest, as the data used in training is known and reliable. This system is made up of two different types of entities, servers and clients. Each client has a local dataset and the server has the global model. Initially, the server sends the weights of the global model to all clients participating in the training. Customers, as soon as they receive the weights from the global model, start training the machine learning model on the local dataset they have. With the model trained, each client sends the model weights and the size of the data set used in training to the server, which has all the.

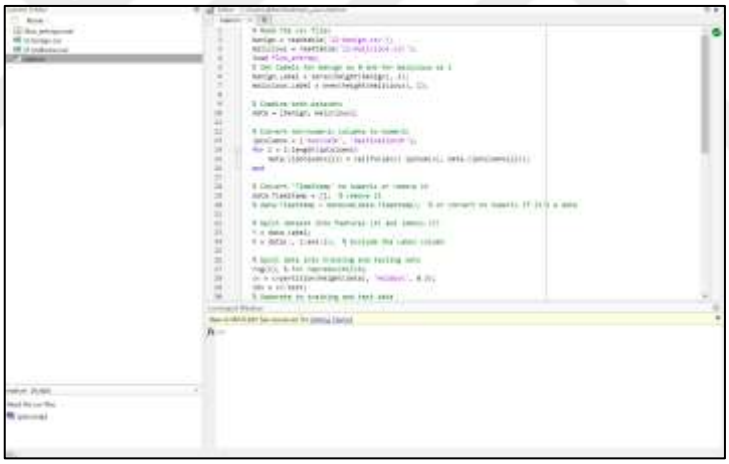


Figure 4.1: UML of the Machine Learning System With Parameter Topology.

Training data from all participating clients begins the model aggregation process. This process consists of applying the Federated Average algorithm (FedAvg) which will calculate a weighted average of the models' weights based on the size of the data set used in training each client. Once the aggregation is complete, the server sends the resulting model weights to all clients.

4.1.1 Initializing Simulation

When starting the program, the structures of the system components will be defined. The definition of the structures of the system components are made based on the declaration of

classes in the context of the object-oriented paradigm used in the simulation project. They are the Net class that defines the structure of the neural network used in learning, the Server class that defines the attributes and functions of the parameter server and the Client class that defines the attributes and functions that the system's clients have. After defining the classes of each member of the system, the simulation parameters will be defined and the objects that will make up the simulation will be instantiated. First, the Net class is declared, which refers to the system's neural network model. This class requires two attributes to initialize the neural network, they are the number of columns that the dataset has and the number of classes in which the flows in the dataset are labeled. To build this neural network, the Sequential class from the tensorflow library was used, which allows the addition of layers and activation functions to the model in an intuitive way. When initializing the Sequential class, the number of nodes in the input layer of the neural network is defined, which is equal to the number of columns in the dataset, the number of nodes in the first and second hidden layers equal to 200 with ReLU activation function. The ReLU activation function converts negative numbers to a value of 0 and maintains the value of positive numbers that pass through this function. The output layer is added with the number of output nodes equal to the number of classes to be predicted and applying the softmax activation function that serves to return the resulting probability of each class in decimal format and which when added together results in the value 1. The binary-crossentropy function was chosen as the loss function due to the fact that there were only two classes to be predicted. The cross-entropy function (crossentropy) is a function that calculates the value of the difference between two probability distributions for a given random variable or set of events. For the optimization function, the adam function was chosen because it applies the stochastic gradient descent (SGD). A definition of the Net class can be seen in Figure 4.1. In a global scope, the parameters and functions relating to the model training simulation are defined. Firstly, the method that will split and send the data set to clients in a non-IID way is defined. Each client will have data from only one IP in this data set. For each participating client, a random IP is selected from this set that will be used to filter the data that will make up the client's data set.

4.1.2 Training

In this part, the training and aggregation of the model will be carried out. For the simulation, 30 participating customers were used and 10 rounds of model training were carried out. First,

clients preprocess their respective datasets before starting the training round, then each participant uses the train-model function to compile and train their model locally. Once training is complete, the client sends its weights and the size of the data set used in training and calls the test-model function to display and store the metrics of the model resulting from this round of training. At the end of each round, the server, already having all the client training data, calls the aggregate-models function which will calculate the federated average on the models generated by the training to aggregate them, and then sends the resulting weights of aggregation for each client, which will create a new local model with the new weights sent by the server.

This process is repeated for each training round. The number of 10 rounds was chosen because after the tenth round there were no further changes in the metrics, and with this, it can be deduced that the model reached stability

4.2 DISTRIBUTED INTRUSION DETECTION SYSTEM WITH TOPOLOGY

Unlike the Intrusion Detection System with Parameter Communication Topology, in the Distributed Intrusion Detection System with Point-to-Point Communication Topology, clients communicate with each other directly and share the machine learning model. In this implementation, the Decision Tree (DT) was used as a machine learning model for network intrusion detection. After each client's model is trained, the process of sharing these models begins, where clients store other clients' models in a list, which will then be used to predict the type of network data flow using the Bagging method This list of Decision Trees is called KNN in this work. The figure 4.2 explains how the proposal works. The Decision Tree model was chosen because it is one of the most practical models for approximating functions with discrete values, and compared to the neural network, it is a simpler model in terms of mathematical calculations and therefore requires less computational power to train. it. This system is composed of several nodes, where each node has a set of data

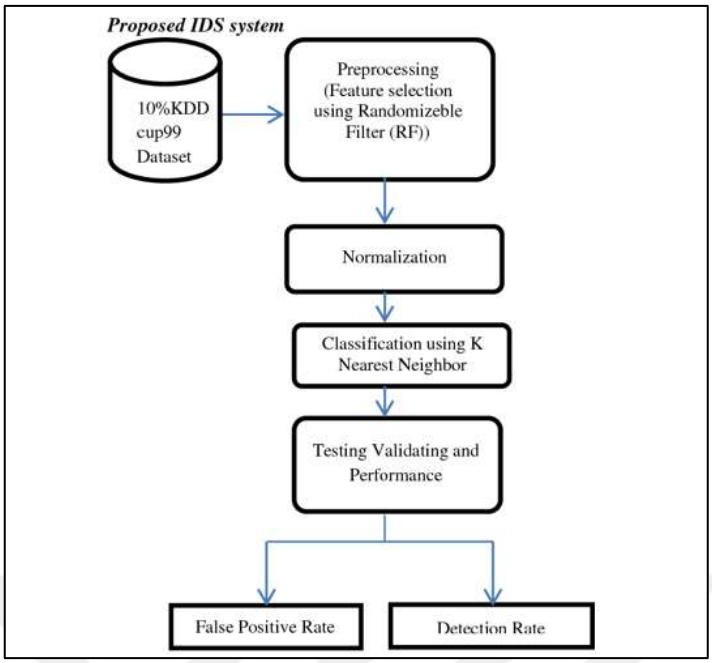


Figure 4.2: Intrusion Detection System Based on Distributed KNN.

The nodes participating in the intrusion detection system establish a peer-to-peer overlay network. Local models are trained on private data shared across the peer-to-peer network. Network packets are analyzed by the model composed of the k best trees and classified according to the voting of the model trees. Local Initially, each node trains the Decision Tree model on the data set it has. As soon as the model is trained, the model sharing procedure begins, where each node sends its model to all other nodes participating in the training. In possession of the neighboring nodes' models, each node will build a set of Decision Trees, called Distributed KNN, which contains all the Decision Trees of the neighboring nodes that have the highest prediction accuracy on the data set it has. It then uses its Distributed KNN to predict network flows using the concept of Bagging, where all Decision Trees in the Distributed KNN predict the flow and the class most predicted by the Trees is the one that will be chosen to label this flow.

4.2.1 Initializing Simulation

To simulate this system, it was necessary to define the Node class that defines the responsibilities and attributes of the nodes that make up the machine learning model training and sharing network.

This class is initialized by the constructor method that contains two parameters. The first parameter refers to the identifier of this node, which in this simulation is used only for the

purpose of analyzing the results, and the second refers to the local data set that this node has. The values of the attributes of this class are then defined. The local-decision-tree attribute is the attribute that will store the decision tree model for this node. This model is instantiated using the KNN Classifier class from the MATLAB library. In this class that instantiates the model, two parameters are passed, max-depth with a value equal to 7, which defines the maximum depth of the decision tree in order to avoid or reduce overfitting of the model with the training data, when the model becomes too fit only to the training data and does not generalize enough to

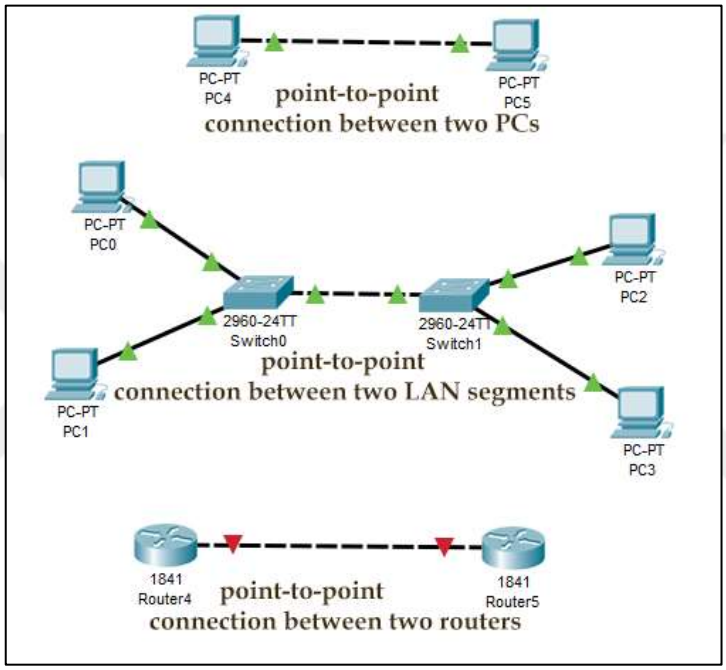


Figure 4.3: UML of the Machine Learning System with Point-to-Point Communication Topology.

Predict other data. Values 5, 6, 7, 8, 9, 10, 11, 12 and 15 for this parameter were tested using the GridSearch method from the MATLAB library and depth 7 had better results. The other parameter of this class called criterion receives the value 'gini' which relates to the calculation function to measure the quality of the branch separation of this tree. The gini impurity function was chosen because it generated better results compared to the entropy function. Soon after, the local-forest attribute, which represents the KNN of that node, receives a list initially containing the model instantiated by the node. This list is the set that will store the models of other nodes in the network in the part where models are shared between network nodes. A description of the Node class can be seen in Figure 4.3

Another method defined for this class is the model training method. In this method,

the fit function of the KNN Classifier model is basically invoked, where the node's training data is passed to perform learning on that node's data.

4.2.2 Training

Once the model training process is completed, the network nodes share their models with each other. In this case, all nodes communicate with each other, but in a real application this could be very costly or even unfeasible depending on the scale of this network and the distance between the nodes. When models are shared, each node adds to its set of decision trees the models that obtain the best accuracies on the node's test data. This set is called Distributed KNN and has a size equal to 5 for all nodes in the network. Furthermore, other metrics could be used as decision factors for inserting these trees into the sets, such as precision and recall metrics.

Then, each node tests its Distributed KNN on data from the dataset reserved for testing. The network flow data is classified by the KNN using the bagging algorithm, which consists of a vote of the Forest's Decision Trees, where each tree predicts the class of that flow and the class most predicted by the trees in the set is the class that will label this flow. With this, we can calculate the accuracy, precision, recall and f1-score metrics for the KNN, tested on the same set of data that the decision tree used for testing, for comparison purposes.

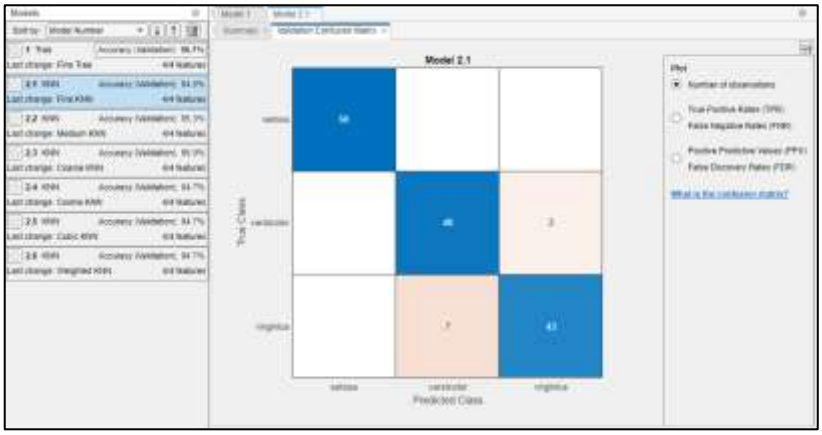


Figure 4.4: Confusion Matrix of the Proposed Method.

5. RESULTS AND DISCUSSION

The system simulations were carried out on a computer with an Intel Core i7-11700 CPU @ 3.00GHz, 16 GB of RAM and GeXForce 4 GB GPU. A total of 4 GB of data was used to train the models, which corresponds to 5 days of collected data streams and 23 million data streams. The metrics used to measure the training quality of the models used in the systems, were the loss function, accuracy, precision, recall, F1 measure and area under the ROC curve (AUC score).

5.1 EVALUATION OF INTRUSION DETECTION SYSTEM WITH TOPOLOGY

Firstly, the test results of training the model from a high-performing client will be presented. Then the results of the model training test from a low-performing client will be presented for comparison purposes. Finally, an overview of the final training test results of all participating customers is presented.

The system simulation was carried out using 30 participating clients and a parameter server. In the figure 5.1 you can see the online graph of the values of the accuracy, precision, recall and AUC metrics of the final test of a client's neural network machine learning model in which the model obtained an accuracy and recall of 91%, precision of 92% and AUC of 92.5%. This client was selected because it achieved the best performance compared to other participating clients. The dispersion of this customer's model metrics values was low, which indicates that the aggregated model performed well when predicting this customer's test sets. Furthermore, Figure 5.2presents the ROC curve referring to the final test of this customer's model. It is observed that precision and recall are well balanced, indicating that the model performed well in predicting the flows of this client's data set.

In the figure 5.3 the values of the accuracy, precision, recall and AUC metrics of the final test of the neural network machine learning model from another participating client are presented, in which the model obtained an accuracy and recall of 62%, precision and AUC of 63% . This client was selected because it had the worst performance compared to other clients. The dispersion of the metrics values of this client's model was high, indicating that the model did not fit the data that this client has. Furthermore, in Figure 5.4, the ROC curve referring to the final test of this customer's model,

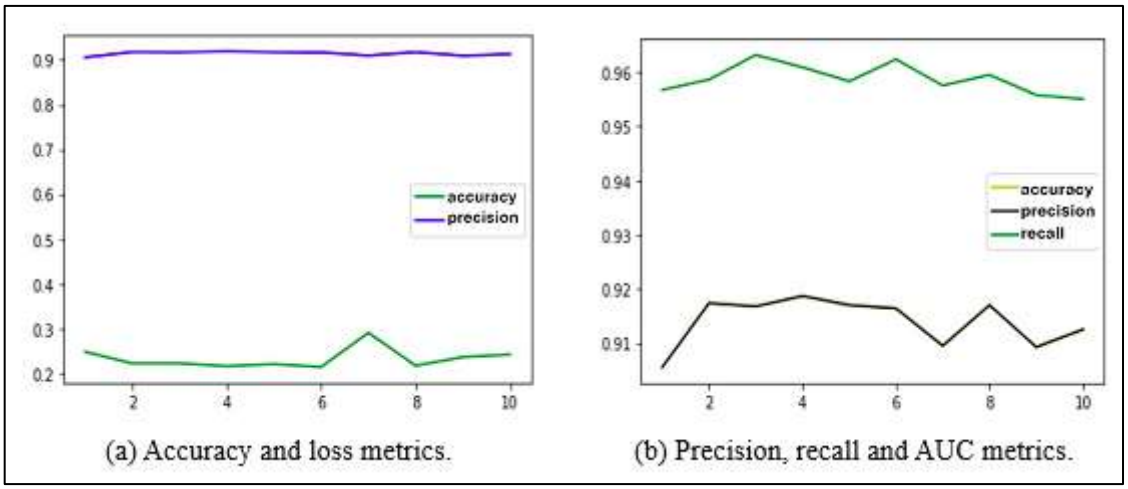


Figure 5.1: Values of the Accuracy, Loss Function, Precision, Recall and Auc Metrics in Each Training Round For A Client Participating in Federated Learning With A Parameter Server Topology.

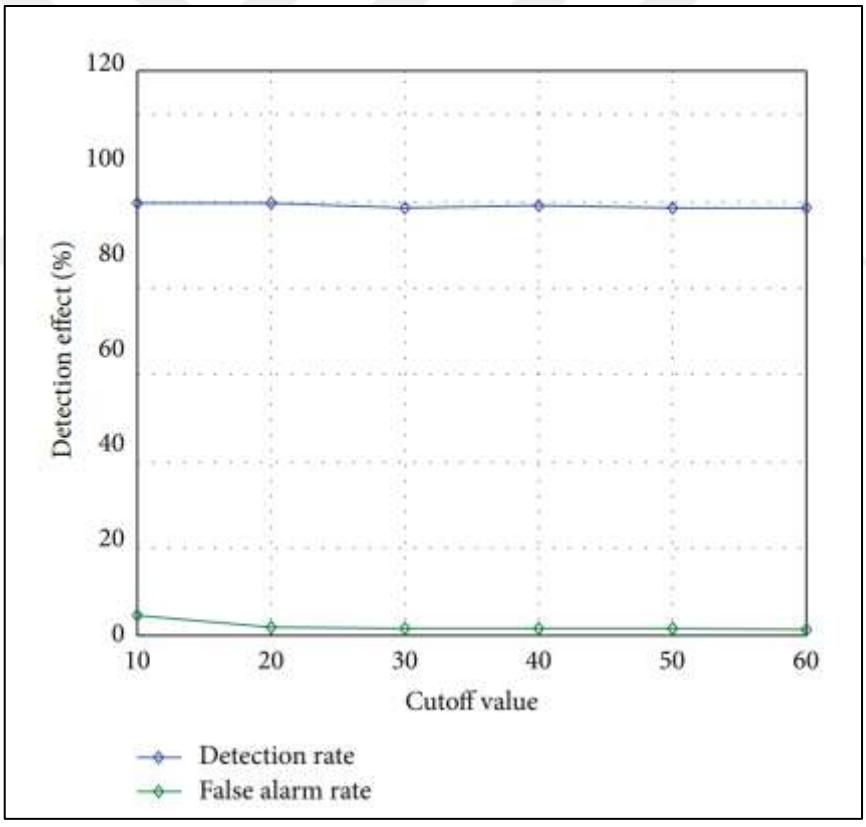


Figure 5.2: ROC Curve of a Client Participating in Federated Machine Learning with Parameter Server Topology.

Where it can be seen that precision and recall are unbalanced, resulting in a low-performance model for predicting the flows of this client's data set.

In a general overview, the simulation results of the system with Parameter Server topology show that the aggregated model obtained a median accuracy, precision and recall of 86% and a median AUC equal to 93% in the final test of customer models participants, as shown in Figure 5.3 These results indicate that the aggregated model achieved high performance in predicting the flows of the customer test dataset, despite there being outliers with the metrics in the range between 60% and 70%.

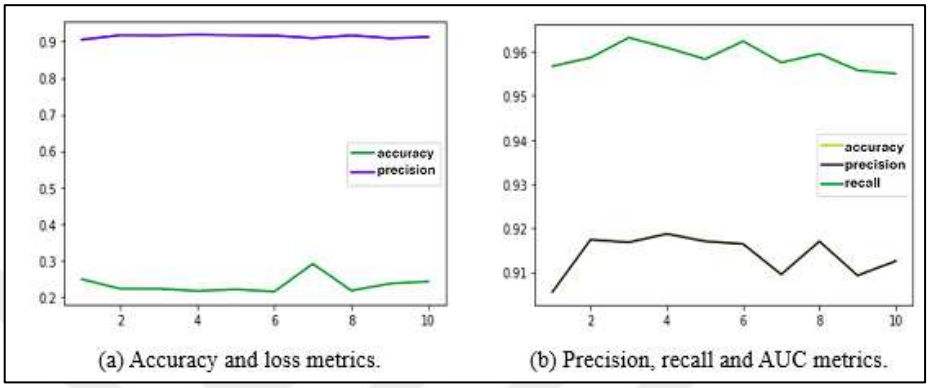


Figure 5.3: Values of the Accuracy, Loss, Precision, Recall and Auc Metrics in Each Training Round of a Client Participating in the Federated Machine Learning System With Parameter Server Topology.

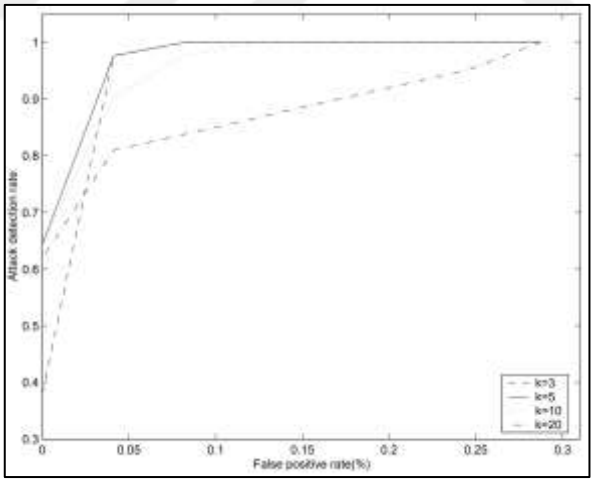


Figure 5.4: ROC Curve of a Client Participating in the System's Distributed Machine Learning with Parameter Server Topology.

5.2 EVALUATION OF THE DISTRIBUTED INTRUSION DETECTION SYSTEM WITH

Peer-to-Peer, each client performs one round of model training (Decision Tree DT). Once the models have been trained, the accuracy, precision, recall, and F1 measure metrics for the

local DTs of each participating node are calculated. The figure 5.5 presents the values of the metrics accuracy, precision, recall and F1 measure referring to the Decision Tree prediction of the nodes participating in the system on the data set intended for testing. The value of the median accuracy of these models is equal to 93%, the median precision is equal to 91%, the median recall is equal to 96% and the median

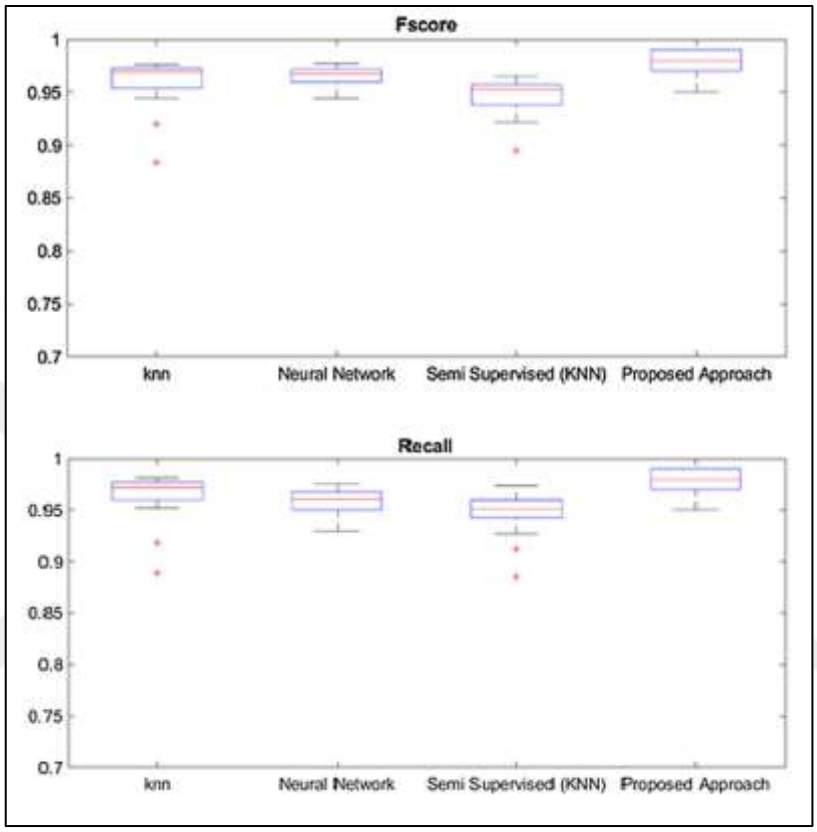


Figure 5.5: Final Metrics of the Loss Function, Accuracy, Precision, Recall and Auc of the Clients Participating in the Distributed Machine Learning of the System with Parameter Server Topology.

of measurement F1 equal to 93%. In the figure 5.5 The values for the metrics referring to the prediction of the Distributed KNN of the participating nodes on the data set intended for testing are represented. The Decision Tree of a node is a model trained only on the data set of that node, whereas the Distributed KNN is a model that corresponds to a set of Decision Trees from several nodes in the network. The median accuracy value of these models is equal to 79%, the median precision is equal to 83%, the median recall is equal to 79% and the median F1-score is equal to 80%.

The difference in the values of the Decision Tree and KNN metrics is due to the fact that the node data sets are non-IID among themselves, which causes the prediction of the

Decision Trees that were trained with the set of data from neighboring nodes have lower performance when predicting the test data set of a participating node, since an aggregation of all models into one is not performed, as is done with Federated Average. However, using Decision Trees trained on distinct non-IID datasets reduces the chances of predictions being biased, increasing the reliability of predictions made by KNNs.

The size of the Distributed KNN was selected based on observation of the model's performance for each size tested, as can be seen in Figure16. It can be observed that as the size increases the average of the precision metrics and F1 decreases, however the average of the recall metric increases until the size equals 6. The decrease in the average of the precision metrics and F1 can be explained by the fact of generalization of the model in predicting flows with a distribution different from the distribution of flows on which the model was initially trained, however this generalization allowed the model to acquire greater recall when predicting these flows.

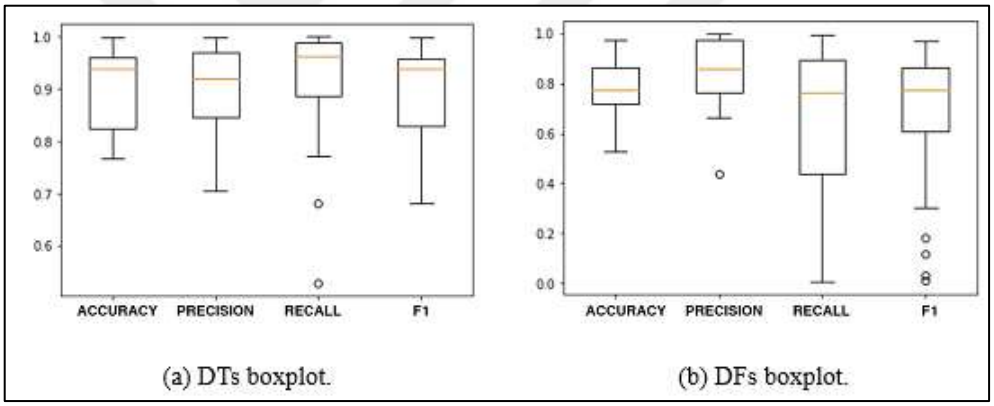


Figure 5.6: Final Metrics of Accuracy, Precision, Recall and F1-Score of the Decision Tree (A) And Distributed Knns (B) of the Nodes Participating in the Distributed Machine Learning of the System with Point-To-Point Topology.

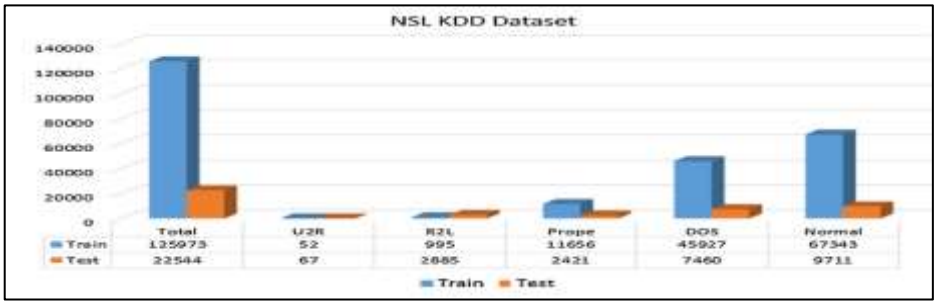
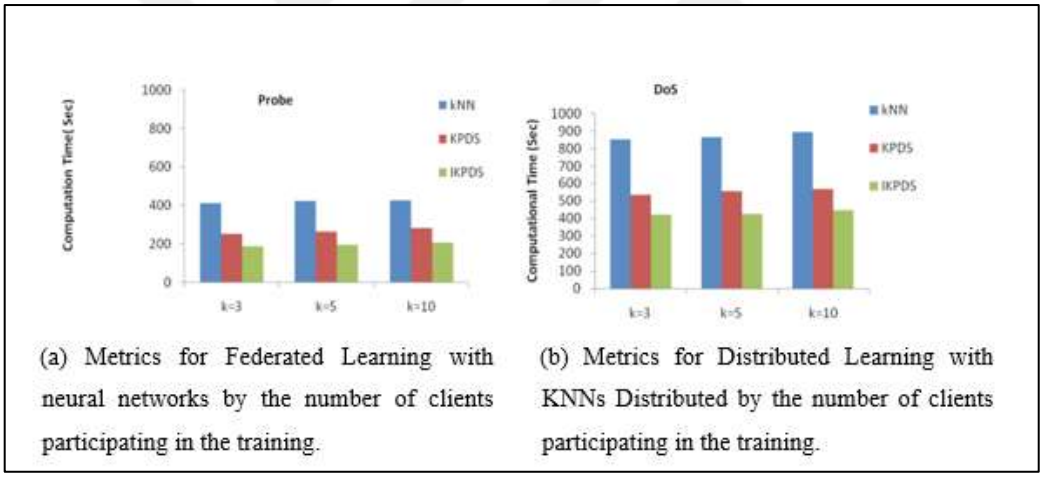


Figure 5.7: Average of the Final Metrics of Accuracy, Precision, Recall and F1 of the Distributed Knn Models of the Nodes Participating in The Distributed Machine Learning of the System with Point-To-Point Communication Topology by The Size K of the Distributed Knn.

5.3 COMPARISON OF SYSTEM RESULTS

From the results shown in Figure 5.7 one can compare the training performance of the machine learning models of the implemented systems. Although the performance of the Neural Network of systems with Parameter Server topology is greater than the performance of the Distributed KNN, according to the metrics presented, it is worth highlighting that the time it took for each model to achieve such results were quite different. The KNN model The distributed model proposed in this work achieved such results in approximately 99.3 seconds, compared to the Neural Network model that achieved such results in approximately 952.5 seconds, both simulations carried out under the same conditions and with the same amount of data. However, Decision Trees require a large amount of training data compared to other models to achieve optimal performance with little bias Neural Networks allow the model to be trained repeatedly on the same set of data, resulting in optimal performance with little bias, thus reducing the amount of data needed to achieve such levels of results/



(a) Metrics for Federated Learning with neural networks by the number of clients participating in the training.

(b) Metrics for Distributed Learning with KNNs Distributed by the number of clients participating in the training.

Figure 5.8: Accuracy, Precision, Recall, F1-Score and Auc-Score Metrics of Neural Networks (A) And Distributed Knns (B) As A Function of the Number of Participants in The Systems.

The increase in the number of participants tends to imply an improvement in the performance of the Distributed KNN. The figures 5.8 compare metrics of neural networks and Distributed KNNs by the number of participants in the training of system models with parameter server communication topology and system with peer-to-peer communication topology. The values presented are the averages, with a 95% confidence interval. It appears that, as the number of participants increases, the performance of the neural network model tends to decrease. On the other hand, the performance of the Distributed KNN model tends to increase, which shows that the Distributed KNN, used in the system with peer-to-peer

communication topology, is more scalable compared to the neural network used in the system with parameter server communication topology. This occurs due to the characteristics of the aggregation method used in the Distributed KNN, in which voting is little impacted by biased models, which tends to grow as the number of participants increases. Outside the scope of machine learning, the client-server network architecture presents several problems related to the communication flow bottleneck, and problems related to a single point of failure. Such problems are due to its centralized architecture, in which the server, or servers, need to handle all communication with the much more numerous clients. Such problems are mitigated in the architecture of point-to-point networks.

6. CONCLUSIONS AND FUTURE WORK

Due to the meteoric expansion in network-based services and information that is stored and accessible over networks, the significance of network security and the process of securing networks is expanding at a rate that is unprecedented. This is because the meteoric rise in network-based services and information that is stored and accessed over networks. Abnormalities and attacks on a network can be recognized with the assistance of an intrusion detection system, which is commonly referred to as an IDS. In order to extract useful information from vast amounts of network data, NIDS has extensively adopted data mining techniques. This was done with the intention of achieving the goal. This is due to the fact that the proliferation of new forms of attacks or intrusions comes hand in hand with an increase in how difficult it is to detect them. Clustering and classification algorithms are put to wide use in the field of intrusion detection systems (IDS). As a result, making improvements to the capabilities of these algorithms has the potential to significantly raise the overall efficiency of the sector. This research makes use of a hybrid approach by combining feature clustering with KNN classification in order to improve upon an existing technique of intrusion detection that has already been devised. This is accomplished through the utilization of a hybrid methodology. Using the NSLKDD dataset, we studied a range of methods with the goal of enhancing the intrusion detection classification performance of feature weighting within the scope of this research. We used the nearest neighbor (KNN), the farthest neighbor (k-FN), the second nearest neighbor (NN), and the second nearest neighbor (NN) to classify the data. Both sets of neighbors were required to share the same class label. The primary goal of the work that was presented was to carry out a comparison of these forward-thinking methodologies to the direct KNN classification that was applied in feature weighting. According to the findings, the one-of-a-kind strategies either resulted in enhanced performance or resulted in performance that was comparable to that of the direct KNN classification in terms of accuracy, detection rate, and reducing the number of false positive alerts. The extraction of useful information from large amounts of data is an important subject that, in subsequent research, will demand additional investigation. The creation of specialized solutions that can be used in intrusion detection may be able to assist ease some of the problems that are now hurting data mining in general. These problems have been brought about by recent events.

REFERENCES

[1] L. A. Maglaras, K. H. Kim, H. Janicke, M. A. Ferrag, S. Rallis, and P. Fragkou.

[2] A. Ahmim, M. Derdour, and M. A. Ferrag, "An intrusion detection system based on combining probability predictions of a tree of classifiers," *Int. J. Commun. Syst*, vol. 31, no. 9, pp. 3547–3547, 2018.

[3] A. Ahmim, L. Maglaras, M. A. Ferrag, M. Derdour, and H. Janicke, "A novel hierarchical intrusion detection system based on decision tree and rules-based mod- els," *15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 228–261, 2019.

[4] Z. Dewa and L. A. Maglaras, "Data mining and intrusion detection systems," *Int. J. Adv. Comput. Sci. Appl*, vol. 7, no. 1, pp. 62–71, 2016.

[5] B. Stewart, L. Rosa, L. A. Maglaras, T. J. Cruz, M. A. Ferrag, and P. Simões, "A novel intrusion detection mechanism for scada systems which automatically adapts to network topology changes," *EAI Endorsed Trans. Ind. Netw. Intell. Syst*, no. 10, pp. 4–4, 2017.

[6] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, *Toward generating a new intru- sion detection dataset and intrusion traffic characterization*. 2018.

[7] R. Ismael, F. Abeer, T. M. Nidaa, and F, "Optimized Deep Learning with Binary PSO for Intrusion Detection on CSE-CIC-IDS2018 Dataset," *J. Al Qadisiyah Comput. Sci. Math*, pp. 16–16, 2020.

[8] Akashshukla and A. Himanshosharma, "Future of Internet of Things: Trends, Challenges & Insight To Artificial Intelligence," *International Journal of Advanced Research in Computer*, vol. 9, no. 2, 2018. [9] G. A. Fink, D. V. Z. Thomas, and E. Caroll, 2015.

[10] S. Q. Tan, B. H. M. Sundaram, S. Wang, T. Ng, Y. Chang, V. Aung, and K. M. M, "Secure searching on cloud storage enhanced by homomorphic indexing," *Future Gener. Comput. Syst*, vol. 65, pp. 102–110, 2016.

[11] M. H. Prasantagogoi and Bhuyan, 2012.

[12] A. A. Ghorbani, W. Lu, and M. Tavallaee, 2010.

[13] S. Ho, “Intrusion Detection - Systems for today and tomorrow,” *SANS Institute 2019*, pp. 2–2.

[14] A. R. Javed, M. O. Beg, M. Asim, T. Baker, A. H. Al-Bayatti, and Alphalogger, “Detecting motion-based side-channel attack using smartphone keystrokes,” *J. Ambient Intell. Humaniz. Comput*, 2020.

[15] H. Liao, C. R. Lin, Y. Lin, and K. Tung, “Journal of Network andComputer Applications Intrusion detection system,” *J. Netw. Comput*, vol. 36, pp. 16–24, 2013.

[16] A. Nisioti, A. Mylonas, and V. Katos, 2018.

[17] E. Kabir, J. Hu, H. Wang, and G. Zhuo, “A novel statistical technique for intrusion detection systems,” *Future Gener. Comput. Syst*, vol. 79, pp. 303–318, 2017.

[18] M. Petkovic, I. Basicevic, D. Kukolj, and M. Popovic, “Evaluation of takagi-sugeno-kang fuzzy method in entropy-based detection of DDoS attacks,” *Comput. Sci. Inf. Syst*, vol. 15, pp. 139–162, 2018.

[19] A. Feizollah, S. S. Anuar, N. B. Salleh, and R, “A study of machine learning classifiers for anomaly-based mobile botnet detection,” *Malaysian J Comput Sci*, vol. 26, no. 4, pp. 251–265, 2014.

[20] A. H. Hamamoto, L. F. Carvalho, L. H. Sampaio, T. Abrão, and M. L. Proença, “Network anomaly detection system using genetic algorithm and fuzzy logic,” *Expert Syst. Appl*, vol. 92, pp. 390–402, 2018.

[21] G. Dupont, J. Hartog, S. Etalle, and A. Lekidis.

[22] Shawqm, Mehibs, H. Soukaena, and Hashim, “Proposed Network Intrusion Detection System In Cloud Environment Based on Back Propagation Neural Network,” *Journal of Babylon university/Pure and Applied Sciences*, vol. 26, no. 1, 2018.

[23] M. W. Berry, M. A. Yap, and B. W, *Supervised and Unsupervised Learning for Data Science*. New York, NY: Springer, 2019.

[24] M. A. Al-Garadi, 2020.

[25] T. M. Richardzuech and Khoshgoftaar, “A survey on feature selection for intrusion detection,” in *21st ISSAT International Conference on Reliability and Quality in Design*, 2015.

[26] A. Tawfeeqshawly and Elghariani, 2019.

[27] S. Mohammadi, H. Mirvaziri, and H. Mostafaghazizadeh-Ahsaee, “Cyber intrusion detection by combined feature selection algorithm,” *Journal of Information Security and Applications*, vol. 44, pp. 80–88, 2019.

[28] A. Fadisalo, A. Bounassif, and Essex, “Dimensionality Reduction with IG-PCA and Ensemble Classifier for Network Intrusion Detection,” *Computer Networks*, 2018.

[29] F. Gottwalt, E. Chang, T. Dillon, and Corrcorr, “A Feature Selection Method for Multivariate Correlation Network Anomaly Detection Techniques,” *Computers&security*.

[30] M. Kwangjokim and Harrychaandra, “Network Intrusion Detection using Deep Learning: A Feature Learning Approach,” *SpringerBriefs on Cyber Security Systems and Networks*, 2018.

[31] N. Quamar, W. Sun, Y. Ahmad, and M. Javaid, “A Deep Learning Approach for Network Intrusion Detection System,” *Conference Paper in Security and Safety*, 2015.

[32] T. A. Tang, L. Mhamdi, and Mclernon, “Deep Learning Approach For Network Intrusion Detection in Software Defined Networking,” *International Conference on Wireless Networks and Mobile Communications (WINCOM)*, 2016.

[33] M. F. Elrawy, A. I. Awad, F. A. Hesham, and Hame, “Intrusion detection systems for IoT-based smart environments: a survey,” *Journal of Cloud Computing :Advances, Systems and Applications*, 2018.

[34] K. Kim and Muhamaderzaaminanto, 2017.

[35] D. Kwon, Hyunjookim, S. C. Jinohkim, I. Suh, K. J. Kim, and Kim, 2017.

[36] K. Kim and Muhamaderzaaminanto, 2017.

[37] R. C. Staudemeyer, 2015.

[38] R.-H. Hwang, M.-C. Peng, V.-L. Nguyen, and Yu-Lunchang, "An LSTM-Based Deep Learning Approach for Classifying Malicious Traffic at the Packet Level," *AppliedScience*, pp. 9163414–9163414, 2019.

[39] G. Kim, H. Yi, J. Lee, S. Yunheungpaek, and Yoon, 2016.

[40] C. Li1, J. Wang1, and X. Ye1, "Using a Recurrent Neural Network and Restricted Boltzmann Machines for Malicious Traffic Detection," *NeuroQuantology*, vol. 16, no. 5, pp. 823–831, 2018.

[41] L. Zhang, M. Li, X. Wang, and Y. Huang, "An Improved Network Intrusion Detection Based on Deep Neural," *Network IOP Conference Series: Materials Science and Engineering*, 2019.

[42] M. Kasunamarasinghe and Manic, 2018.

[43] W. Wang, Y. Sheng, and J. Wang, "HAST-IDS: Learning Hierarchical Spatial-Temporal Features using Deep Neural Networks to Improve Intrusion Detection," *IEEE*, 2017.

[44] B. . Lee, Amaresh, . Sandhya, C. . Green, and D. Engels, "Comparative Study of Deep Learning Models for Network Intrusion Detection," *SMU Data Science Review*, vol. 1, no. 1, 2018.

[45] D. H. B. Roy and Cheung, "A Deep Learning Approach for Intrusion Detection in Internet of Things using Bi-Directional Long Short-Term Memory Recurrent Neural Network," *28th international Telecommunication Network Communication Conference (ITANC)*, 2018.

[46] P. Wei, Y. Li, Z. Zhang, and T. Hu, 2019.

[47] V. Kanimozhi and P. Jacob, "UNSW-NB15 Dataset Feature Selection and Network Intrusion Detection using Deep Learning," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 7, no. 5S2, 2019.

[48] R. Vinayakumar, S. P. Mamounalazab, and Prabaharanp.

[49] D. G. Palmieri and F, "Network traffic classification using deep convolutional recurrent autoencoder neural networks for spatial-temporal features extraction," *Journal of Network and Computer Applications*, pp. 102890–102890, 2021.

[50] M. Aqeel, A. D. H. Alhussainy, and Jasim, "Half Gaussian-based wavelet transform for pooling layer for convolution neural network," *TELKOMNIKA Telecommunication, Computing, Electronics and Control*, vol. 19, pp. 163–172, 2021.

[51] A. Ss, S. A. Chao, and C. Cs, "Passive browser identification with multi-scale Convolutional Neural Networks," *Neurocomputing*, vol. 378, pp. 238–247, 2020.

[52] Y. Shi, D. Feng, and Y. Cheng, "A natural language-inspired multilabel video streaming source identification method based on deep neural networks," *Signal Image and Video Processing*, pp. 1–8, 2021.

[53] Y. He and W. Li, "Image-based Encrypted Traffic Classification with Convolution Neural Networks," *IEEE Fifth International Conference on Data Science in Cyberspace (DSC)*, 2020.

[54] S. Mishra, A. Y. Rafidsagban, and Gandhi, "Swarm Intelligence in anomaly detection systems: an overview," *International Journal of Computers and Applications*, 2018.

[55] Selvakumarb and K. Muneeswaran, "Firefly algorithm based Feature Selection for Network Intrusion Detection," *Computers & Security*, 2018.

[56] Y. Zhang, P. Li, and Xinhengwang.

[57] K. Yanqingyang, Chunhuawu, and Y. Xinxinniu, "Building an Effective Intrusion Detection SystemUsing the Modified Density Peak Clustering Algorithm and Deep Belief Networks," *Applied Science*, vol. 9, 2019.

[58] Mazinim, Shirazib, and I. Mahdavi, “Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoostalgorithms,” *Journal of King Saud University - Computer and Information Sciences*, 2018.

[59] S. Rathore and J. H. Park, “Semi-supervised learning based distributed attack detection framework for IoT,” *Appl Soft Comput*, vol. 72, pp. 79–89, 2018.

[60] N. Moustafa, B. Turnbull, and K. R. Choo, “An Ensemble Intrusion Detection Technique Based on Proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things,” *IEEE Internet of Things Journal*, vol. 6, pp. 4815–4830, 2019.

[61] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, and A. Alazab, “A Novel Ensemble of Hybrid Intrusion Detection System for Detecting Internet of Things Attacks,” *Electronics*, vol. 8, no. 11, pp. 1210–1210, 2019.

[62] J. Arshad, M. A. Azad, and R. Amad, “Khaled Salah, Mamoun Alazab and Razi Iqbal "A Review of Performance, Energy and Privacy of Intrusion Detection Systems for IoT,” *Electronics*, vol. 2020, pp. 9040629–9040629.

[63] C. D. Mcdermott, F. Majdani, and A. V. Petrovski, “Botnet Detection in the Internet of Things using Deep Learning Approaches,” *International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2018.

[64] V. Sharma, I. You, K. Yim, I. Chen, and J. Cho, “BRIoT: Behavior Rule Specification-Based Misbehavior Detection for IoT-Embedded CyberPhysical Systems,” *IEEE*, vol. 7, pp. 118556–118580, 2019.

[65] D. Li, L. Deng, M. Lee, and H. Wang, “IoT data feature extraction and intrusion detection system for smart cities based on deep migration learning,” *Int. J. Inf. Manag*, vol. 49, pp. 533–545, 2019.

[66] J. Bassey, D. Adesina, X. Li, L. Qian, A. Aved, and T. Kroecker, “Intrusion Detection for IoT Devices based on RF Fingerprinting using Deep Learning,” *Fourth International Conference on Fog and Mobile Edge Computing (FMEC)*, pp. 98–104, 2019.

[67] M. A. Junaidarshad, K. Azad, W. Salah, Jie, and M. Raziiqbal, 2018.

[68] G. Shevenchawla, “Security as a Service: Real-time Intrusion Detection in Internet of Things,” *ACM ISBN*, 2018.

[69] J. Tournier, F. Lesueur, F. L. Mouël, L. Guyon, and H. Ben-Hassine, 2020.

[70] S. Zhao, W. Li, T. Zia, and A. Y, “A Dimension Reduction Model and Classifier for Anomaly-Based Intrusion Detection in Internet of Things,” *IEEE. 15th Intl Conf on Dependable, Autonomic and Secure Computing*, 2017.

[71] J. Nourmoustafa& and Slay, “The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set,” *Information Security Journal: A Global Perspective*, 2016.

[72] Y. Yiranhao and J. Sheng, 2019.

[73] A. A. Diro and N. Chilamkurti, “Distributed attack detection scheme using deep learningapproach for Internet of Things,” *Future Generation Computer Systems*, 2017.

[74] N. Rakesh, “Performance analysis of anomaly detection of different IoT datasets using cloud micro services,” *International Conference on Inventive Computation Technologies (ICICT)*, pp. 1–5, 2016.

[75] S. Geethapriyathamilarasu, 1977.

[76] V. Kanimozhi and T. P. Jacob, 2019.

[77] M. K. Putchala, “Deep Learning Approach For Intrusion Detection System (IDS,” *The Internet Of Things (IOT) Network Using Gated Recurrent Neural Networks (GRU)*, 2017.

[78] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, “Deep learning for cyber security intrusion detection: approaches, datasets, and comparative study,” *Journal of Information Security and Applications*, vol. 50, pp. 102419–102419, 2020.

[79] C.-C.-I. Dataset, 2019.

[80] Cicflowmeter, 2019.

[81] C. Dataset, 2019.

[82] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, “Towards the development of realistic botnet dataset in the internet of things for network forensic ana- lytics: Bot-IoT dataset,” *Fut. Gener. Comput. Syst*, vol. 100, pp. 779–96, 2019.

[83] Bot-Iot and Dataset, 2019.

[84] M. A. Khan, 2021.

[85] P. Bedi, N. Gupta, and V. Jindal, “I-SiamIDS: An improved Siam-IDS for handling class imbalance in network-based intrusion detection systems,” *Appl. Intell*, vol. 51, pp. 1133–1151, 2021.