Czech University of Life Sciences Prague
Faculty of Economics and Management
Department of Information Technology (FEM)

Diploma Thesis

Network Intrusion Detection System Using Deep Learning Approach

Senait Molla Meressa

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

DIPLOMA THESIS ASSIGNMENT

B.Sc. Senait Molla Meressa, BSc

Systems Engineering and Informatics
Informatics

Thesis title

Network intrusion detection system using Deep learning Approach

---

Objectives of thesis

The objective of the thesis is to find a model IDS which has the highest Detection Rate (DR) and the lowest false alarm rate within the range of suitable models and for the specific network type.

Methodology

After the literature review consisting the analytics part of the currently used techniques, a deep learning approach to the model the intrusion detection system will be used.

For model creation, the KDD Cup'99 dataset will be used, which is the most popular dataset for the evaluation of anomaly intrusion detection methods. The dataset is a collection of simulated raw TCP dump data over a period of time on a local area network. The known attack types are those present in the training dataset while the novel attacks are the new attacks that are not present in the training dataset. The dataset provides four distinct attack types.

The proposed extent of the thesis

60

Keywords

DOS, U2R, R2L

Recommended information sources

1. Kim, Kwangjo, Muhamad Erza Aminanto, and Harry Chandra Tanuwidjaja. Network Intrusion Detection Using Deep Learning: A Feature Learning Approach. Springer, 2018.
2. Sangkatsanee, Phurivit, Naruemon Wattanapongsakorn, and Chalermpol Charnsripinyo. ”Practical real-time intrusion detection using machine learning approaches.” Computer Communications 34.18 (2011): 2227-2235.
3. Jeeshitha, Mrs J. ”NETWORK INTRUSION DETECTION USING DEEP LEARNING.” Tathapi with ISSN 2320-0693 is an UGC CARE Journal 19.21 (2020): 43-48.
4. Liu, Hongyu, and Bo Lang. ”Machine learning and deep learning methods for intrusion detection systems: A survey.” Applied Sciences 9.20 (2019): 4396.
5. Zhang, Lin, et al. ”An Improved Network Intrusion Detection Based on Deep Neural Network.” IOP Conference Series: Materials Science and Engineering. Vol. 563. No. 5. IOP Publishing, 2019.
6. Labib, Khaled, and Rao Vemuri. ”NSOM: A real-time network-based intrusion detection system using self-organizing maps.” Networks and Security 21.1 (2002).

Expected date of thesis defence

2020/21 SS – FEM

The Diploma Thesis Supervisor

Ing. Tomáš Vokoun

Supervising department

Department of Information Technologies

Electronic approval: 29. 7. 2020

Ing. Jiří Vaněk, Ph.D.

Head of department

Electronic approval: 21. 10. 2020

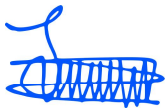
Ing. Martin Pelikán, Ph.D.

Dean

Prague on 21. 03. 2021

Declaration

I declare that I have worked on my diploma thesis titled “Network Intrusion Detection System Using Deep learning Approach” by myself and I have used only the sources mentioned at the end of the thesis.

In Prague on date 31/03/2021 signature 

Senait Molla Meressa

Acknowledgements

I would like to express my sincere gratitude to my supervisor Ing. Tomáš Vokoun for his invaluable support and guidance throughout this thesis work.

I would like to thank my Husband, Ing.Awet haileslassie Gebrehiwot , for being by my side, motivating me, and providing me invaluable comfort both to my professional and personal life: My parents and family members for their immense support.

Finally, I would like to thank the Almighty God to whom I owe my very existence for providing me this opportunity and granted me the capability to proceed successfully and, indeed, throughout my life.

Network Intrusion Detection System Using Deep learning Approach

Systém detekce narušení sítě využívající přístup DeepLearning

Abstraktní

Zvyšující se rychlost síťových útoků je znepokojující problém, který ovlivňuje dostupnost, důvěrnost a integritu dat. K detekci škodlivých kybernetických útoků je třeba mít skutečný bezpečnostní systém, který automatizuje proces monitorování událostí, ke kterým v síti dochází. V této práci byl studován systém pro detekci anomálií narušení a modelován na základě přístupu hlubokého učení. Při experimentální analýze této diplomové práce je použit datový soubor KDD Cup '99. Všechny typy útoků ve vybrané datové sadě spadají do jedné ze tří kategorií vrstev TCP / IP. Proto konkrétně navrhujeme vybudovat tři systémy IDS, tj. IDS aplikační vrstvy, IDS transportní vrstvy a IDS síťové vrstvy, na základě typů útoků malwaru, které se objevují v příslušných vrstvách TCP / IP. Za tímto účelem byl použit datový soubor KDD cup '99 a byla provedena příprava dat k rozdělení datového souboru do tří kategorií navrhovaných systémů IDS. Navrhované systémy detekce narušení byly jednotlivě hodnoceny pomocí různých metrik hodnocení výkonu. Rozsáhlé experimentální výsledky v každé vrstvě ukazují, že navrhovaná metoda dosáhla vyšší rychlosti detekce (DR) a nízké rychlosti falešných poplachů (FAR).

Keywords— : Systém detekce narušení, KDD cup '99, rychlost falešného poplachu, rychlost detekce, hluboké učení, detekce anomálií, LSTM

Abstract

The increasing rate of network attacks is a concerning issue that affects the availability, confidentiality, and integrity of data. To detect malicious cyberattacks, one should require a genuine security system that automates the process of monitoring the events that occur in a network. In this thesis, an anomaly intrusion detection system has been studied and modeled based on a deep learning approach. The KDD Cup '99 dataset is used during the experimental analysis of this thesis work. All attack types within the selected dataset fall under one of the three TCP/IP layer categories. Thus we specifically propose to build three IDS systems, i.e., application layer IDS, transport layer IDS, and Network layer IDS, based on the malware attack types that appear at the corresponding TCP/IP layers. To this end, the KDD cup '99 dataset has been utilized, and data preparation was performed to split the dataset into three categories of the proposed IDS systems. The proposed intrusion detection systems were individually evaluated using different performance evaluation metrics. Extensive experimental results at each layer show that the proposed method has achieved a higher detection rate (DR) and a low false alarm rate (FAR).

Keywords— : Intrusion Detection System, KDD cup '99, False Alarm Rate, Detection Rate, Deep Learning, Anomaly Detection, LSTM

Contents

List of Tables

List of Figures

Chapter 1

Introduction

With the wide-spreading utilization of the Internet and access to online content, the severity of attacks occurring in the network has increased drastically, and attackers continuously develop new exploits attack techniques designed to circumvent the network defense. The tremendous development of web utilization in everyday life raises the concern of how to secure network from different kinds of attacks. The developing prevalence of network attacks is a well-known issue that can affect the availability, confidentiality, and integrity of primary data for both people and organizations. Malicious cyberattacks pose genuine security issues requiring a novel, adaptable, and more solid interruption discovery framework such as an Intrusion detection system (IDS). The intrusion detection system has a significant role in ensuring information and network security.

An intrusion Detection System (IDS) is software or hardware systems that automate the process of monitoring and analyzing the events that occur in a computer network to identify various attacks in the network accurately [6]. It collects information from a network or computer system and analyzes the information for system breaches. An Intrusion Detection System (IDS) is one of every network security infrastructure's most common components.

Based on intrusive behaviors, intrusion detection system is classified into network-based intrusion detection system (NIDS) and host-based intrusion detection system (HIDS)[6]. Network Intrusion detection system (NIDS) monitors and analyses the data packets that

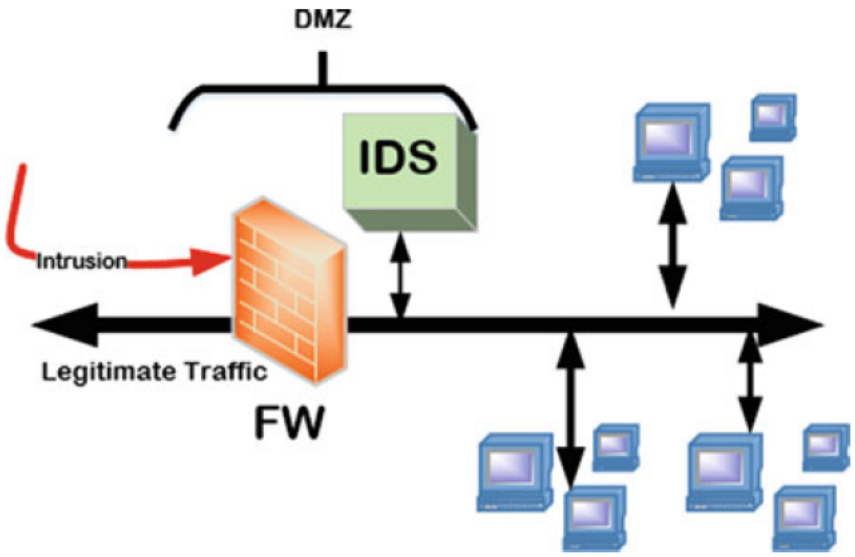


Figure 1.1: Network-based IDS [1].

travel over a network to identify attacks and possible threats concealed within network traffic. The network behaviors are collected using network equipment via mirroring by networking devices, such as routers and switches. A Host-based intrusion detection system (HIDS) uses system activities in the form of various log files running on the local host computer to detect attacks. HIDS is only installed on certain intersection points such as routers and servers. HIDS relies on the information of log files, which includes sensors logs, system logs, software logs, file systems, disk resources, users account information, and others of the system.

Based on the detection method, the Intrusion detection system can be divided into two different types: misuse/signature base and anomaly Intrusion detection system [6]. A misuse-based Intrusion detection system, also known as a signature-based IDS, uses signatures of some well-known attacks and looks for their network data occurrence. Comparing network traffic against a signature base of known attacks clearly indicates system breach whenever it finds any match. This type of IDS is suitable for detecting known attacks and usually achieves higher detection performance for known attacks. Because of the above reason, the misuse detection technique has higher accuracy and reduced false alarm rates than anomaly detection. This detection method's disadvantage is that it fails to detect new or unknown attacks because new attack types may not appear in the signature base. Thus, there is a

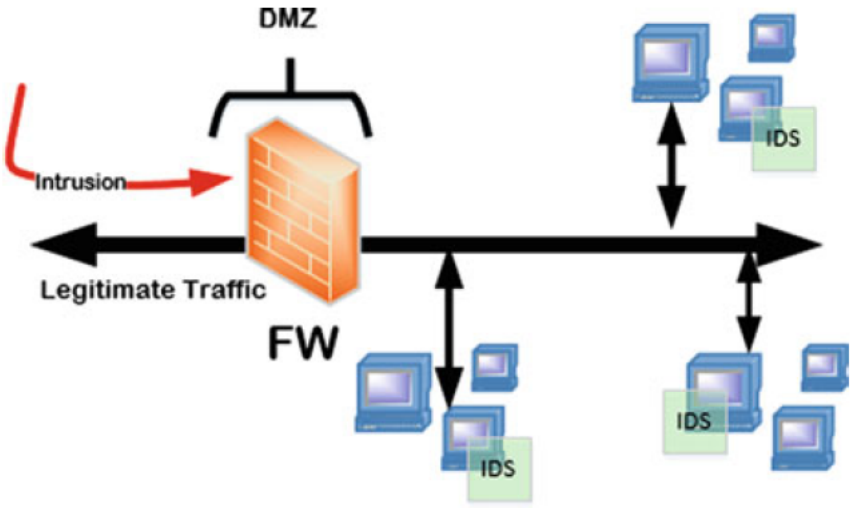


Figure 1.2: Host-based IDS [1].

need for manual interference to update the signature database constantly. Signature-based systems are reactive in that they combat against known attacks that have already damaged several systems before they have been identified.

An anomaly-based intrusion detection system detects an attack by monitoring network traffic and comparing it against an established normal traffic profile baseline. It then triggers an alarm if there is any deviation from it. The classification is based on rules rather than signatures. The big strength of an Anomaly-based Intrusion detection system is that it detects more types of unknown attacks. Its proactiveness is that it can detect security violations before they cause damage is also taken as an advantage. It is an autonomous detection method that can ensure security without any manual interference. The disadvantage of anomaly detection is that it generates a high False alarm rate (false positive rate).

The issue of False alarm (False positive rate) is critical factor in determining the performance of an intrusion detection system. An Intrusion detection system should not only be accurate in detecting the novel attacks,but it must also be able to suppress or eliminate the incorrect alertes (false alarms) raised.

Anomaly detection techniques differ from Misuse detection techniques so that it uses the normal data model to detect anomalous activities. In contrast, the misuse detection model uses signatures of some well-known attacks and looks for their network data occurrence. It

IDS Type	Advantage	Disadvantage	Data Sources
HIDS	-Can check end-to-end encrypted communications. -Detects intrusions by checking hosts file system	- Delays in reporting attacks -Needs to be installed on each host. -can monitor attacks only on the installed machines.	-Audits records, -log files, -Application Program Interface (API),
NIDS	- Detects attacks by checking network packets. - No need to install on each host. -Can check various hosts at the same period. -Can detect the broadest ranges of network protocols	-Difficult identify attacks from encrypted traffic. -It supports only identification of network attacks. -Difficult to analysis high-speed network.	-(SNMP) -Network packets (TCP/UDP/ICMP), - Management Information Base (MIB)

Table 1.1: Comparison of IDS technology types based on their placement within the computer system [1].

relies on human interference to constantly update the signature database.

Artificial intelligence theories such as Machine Learning and Deep Learning have already proven their significance when dealing with data of various sizes.

Chapter 2

Objective and Methodology

2.1 Objective

This thesis focuses on building an intrusion detection system using a deep-learning approach and applying it to a real-world application. The specific goals of this thesis work are listed respectively below as following:

- To model IDS which has highest Detection Rate (DR) and lowe False Alarm Rate (FAR).
- To understand the available techniques to support the vision envisaged for "Anomaly Detection using Machine Learning Technique".
- To understand various anomaly detection and machine learning techniques.
- Identify requirements for building platform for anomaly detection system.

2.2 Methodology

Machine learning techniques can be used to increase the performance of detecting an attack. In this thesis work the concept of deep recurrent neural network algorithms combined with the concept of Transmission Control Protocol/Internet Protocol (TCP/IP) is applied

to design a light-weight and multi-layered Intrusion detection system that will be able to classify each sample of as “normal” or “attack”. The KDD Cup ’99 Dataset which is the publicly available and mostly widely used dataset for intrusion detection system is used in this thesis work. The entire intrusion detection system was implemented in python by using the libraries such as numpy, pandas, and tensorflow. Python is used because it has packages which are useful that can do very difficult computational tasks easily. To evaluate the performance of the proposed algorithms at the 3 layers: Application layer, transport layer and Network layer the evaluation metrics such accuracy, false alarm rate, f1 score, precision and recall are Used. The confusion matrix is also used to maintain the information about actual and predicted classes of the classification system.

2.2.1 Overview of The Proposed System

The data set used in this thesis work is KDD Cup ’99. As first step of this thesis work Data preparation take place: this includes the split of the data set into different layers of TCP/IP based on the attack types present at the respective layers. Below From Figure 2.1 to Figure 2.5 present an illustration on how the detests were prepared to be used as input for the proposed three IDS of TCP/IP layers i.e., Application Layer, Transport Layer, and Network Layer IDS.

```
#Load the Entire Data
#Importing all the required Libraries

import pandas as pd
import numpy as np
mydata = pd.read_csv("kddcup.data_10_percent_corrected_large.csv",header = None,engine = 'python',sep=",")
```

Figure 2.1: Importing all the required Libraries, and then Load the Entire Data.

```
mydata.columns = ["duration","protocol_type","service","flag","src_bytes","dst_bytes","land","wrong_fragement",
                  "urgent",
                 "hot","num_failed_logins","logged_in","num_compressed","root_shell","su_attempted","num_root",
                  "num_file_creations",
                 "num_shells","num_access_files","num_outbound_cmds","is_hot_login","is_guest_login","count","srv_count",
                 "srv_serror_rate","rerror_rate","srv_rerror_rate","same_srv_rate","diff_srv_rate",
                  "srv_diff_host_rate","dst_host_count",
                 "dst_host_srv_count","dst_host_same_srv_rate","dst_host_diff_srv_rate","dst_host_same_src_port_rate",
                  "dst_host_srv_diff_host_rate","dst_host_serror_rate","dst_host_srv_serror_rate","dst_host_rerror_rate",
                 "dst_host_srv_rerror_rate","labels"]
```

Figure 2.2: Define the input features form the dataset.

```
ApplicationLayer = mydata[(mydata['labels'].isin(['normal.','smurf.','back.','satan.','pod.','guess_passwd.',
                                                  'buffer_overflow.',
                                                 'warezmaster.','imap.','loadmodule.','ftp_write.','multihop.'
                                                  ,'perl.']))]
print (ApplicationLayer['labels'].value_counts())


ApplicationLayer.to_csv('DataPrep/Results/FinalApp.txt',header = None,index = False)
```

```
smurf.              280790
normal.              97278
back.                 2203
satan.                1589
pod.                   264
guess_passwd.           53
buffer_overflow.        30
warezmaster.            20
imap.                   12
loadmodule.              9
ftp_write.               8
multihop.                7
perl.                    3
Name: labels, dtype: int64
```

Figure 2.3: Filter and Split the datasets that contain the Attack behaviours which fall to Application Layer.

```
TransportLayer = mydata[(mydata['labels'].isin(['normal.','neptune.','portsweep.','teardrop.','buffer_overflow.',
                                                'land.','nmap.']))]
print (TransportLayer['labels'].value_counts())
len(TransportLayer)
TransportLayer.to_csv('DataPrep/Results/TransApp.txt',header = None,index = False)
```

```
neptune.            107201
normal.              97278
portsweep.            1040
teardrop.              979
nmap.                  231
buffer_overflow.        30
land.                   21
Name: labels, dtype: int64
```

Figure 2.4: Filter and Split the datasets that contain the Attack behaviours which fall to Transport Layer.

```
NetworkLayer = mydata[(mydata['labels'].isin(['normal.','smurf.','ipsweep.','pod.','buffer_overflow.']))]
print (NetworkLayer['labels'].value_counts())
print (len(NetworkLayer))
NetworkLayer.to_csv('DataPrep/Results/NetwApp.txt',header = None,index = False)
```

```
smurf.              280790
normal.              97278
ipsweep.              1247
pod.                   264
buffer_overflow.        30
Name: labels, dtype: int64
379609
```

Figure 2.5: Filter and Split the datasets that contain the Attack behaviours which fall to Network Layer.

Next, data pre-processing is needed in-order to transform the raw unprocessed data into a comprehensible format. In the next step the unique features of the data set are extracted using the Random forest classifier techniques and those extracted features will be used as an input data to the deep learning model. The data set is then split into two portions, training portion and the testing portion. Then The proposed model learns through the training data

set and then later evaluated and checked for accuracy using the testing data set in order to evaluate if the model is able to classify the data as normal and attack. chapter 4 presents the details regarding the proposed method and its working principle.

Chapter 3

Literature Review

3.1 What is firewall

Even though the ease of connectivity to the Internet is advantageous, the malicious intrusions and risks are also increasing day today. As a result of the widespread Internet, computer networks' exploitation is becoming common, and protecting once data from threats becomes a challenge. Personal computers and computer networks are increasingly vulnerable to various kinds of attacks that can violate privacy and can cause important data to be lost [7]. For this reason, information needs to be protected from attacks. The attacks are caused by a failure to implement security policies and the failure to use available security tools. The various security tools that are available in the market include Firewall, Intrusion Detection System, and Honeypot.

Firewalls protect a legitimate network from an illegitimate network by filtering traffic according to a specified security policy [8]. It is a combination of hardware and software systems that separates an organization's internal network from other outside networks, allowing some packets to pass and blocking others. It avoids unauthorized or illegal sessions which are established to the devices inside the network areas it protects. Firewalls are configured to protect against unauthenticated logins from the outside [7].

The firewall has a pair of mechanisms: blocking the traffic and allowing traffic. Many fire-

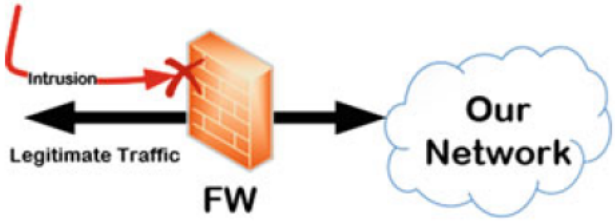


Figure 3.1: Firewall [1]

walls can be deployed in the managed network's proper positions for cooperative, integrated, and in-depth network security protection. Administrators that manage the firewalls have to be careful while setting the firewall rules.

A distributed Denial-of-Service attack (DDoS) is a significant danger for cloud frameworks. Conventional protection approaches cannot be effectively applied in cloud security due to their generally expansive capacity and less effectiveness. According to such types of challenges, a Confidence-Based Filtering method (CBF) is studied for cloud computing [9]. The method is deployed in two periods, i.e., non-attack period and attack period. Mainly, normal packets are collected at a non-attack period for extracting attribute pairs to generate a nominal profile. With the nominal profile, the Confidence-Based Filtering method is carried by calculating the score of a particular packet at the attack period to determine whether to block it or not. The result shows that the Confidence-Based Filtering method (CBF) has a high scoring speed, a small storage requirement, and an acceptable filtering accuracy, making it suitable for real-time filtering in a cloud framework.

3.2 Types of firewall

3.2.1 Packet Filtering Router

A packet-filtering router applies a set of rules to the individual incoming and outgoing IP packet and then forwards or discards the packet. The router is configured to filter the incoming and outgoing IP packet. The Filtering rules are based on information that is contained in a network packet, such as the destination IP address, source IP address, source transport-level

address and destination transport level address, IP protocol interface, field. The packet filter is typically set up as a list of rules based on matches to fields in the IP/TCP header. During a match to one of the rules, that rule is invoked to decide whether to forward or discard the packet. On the other hand, if there is no match to any rule, then a default action is taken. This default action can either be to discard or forward the packet.

3.2.2 Application level gateways

An application-level gateway is also known as a proxy server. The user uses a TCP/IP application, such as Telnet or FTP, to contact the gateway, and in return, the gateway asks the client for the name of the remote host to be accessed. When the client gives and provides substantial Client-ID and authentication information, the gateway contacts the remote host application and relays the TCP segments containing the application data between the two endpoints. If the gateway does not implement the proxy code for a particular application, the service will not be supported and cannot be forwarded across the firewall. Application-level gateways are more secure than packet filters. It is easy to log and audit all incoming traffic at the application level. The main drawback of this type of gateway is the additional processing overhead on each connection.

3.2.3 Circuit level gateways

A circuit-level gateway does not allow an end-to-end TCP connection; and instead, the gateway sets up two TCP connections. One connection is set up between itself and a TCP client on an inner host and one between itself and a TCP client on an outside host. Once the two connections are set up, the gateway typically relays TCP segments from one connection to the other without looking at the contents. The security function comprises of deciding which connections will be permitted.

3.2.4 Advantages of Firewalls

- Firewalls can prevent the traffic which is non-legitimate [7].
- Firewalls can filter those protocols and services that can be easily exploited [7].
- A firewall helps to protect the internal network by hiding names of internal systems from the outside hosts [7].
- Firewalls can concentrate extended logging of network traffic on one system [7].

3.2.5 Disadvantages of Firewalls

- Firewalls use a manually configured set of rules that differentiate legitimate traffic from non-legitimate traffic [7].
- The firewall can ”t react to a network attack nor can initiate effective counter-measures [7].
- Most firewalls do not analyze the contents of the data packets that make up network traffic [7].
- Firewalls cannot prevent attacks coming from Intranet [7].
- Filtering rules of the firewall cannot prevent attack coming from application layer [7]

A different set of firewalls is being used this time. A taxonomy is required to understand firewall vulnerabilities in the context of firewall operations since it is infeasible to examine or test each firewall for all possible potential problems. Seny Kamara et al. [10] described a novel methodology for analyzing vulnerabilities in Internet firewalls. A firewall vulnerability is an error that is made during firewall design, implementation, or configuration that can be exploited/utilized to attack the trusted network that the firewall is supposed to protect. They have examined firewall internals and cross-reference each firewall operation with causes and effects of the weaknesses in that operation, analyzing twenty reported problems with

available firewalls. Their analysis is a set of matrices that illustrate the distribution of firewall vulnerability causes and effects over firewall operations. These matrices help avoid and detect unforeseen problems during both firewall implementation and firewall testing.

3.3 Intrusion Detection System

An intrusion detection system protects a computer network from unauthorized users, which are also known as attackers. An IDS becomes a standard security measure in network security. Unlike Firewall (FW) in Figure 3.1, IDS is usually located inside the network to monitor all internal traffics. Intrusion detection is detecting actions that attempt to compromise the confidentiality, integrity, or availability of a resource. Incidents have many causes, such as malware (i.e., worms, spyware), authorized users who misuse their access or attempt to gain additional access for which they do not have the authorization, attackers gaining unauthorized access from the Internet [8]. Using both Firewall and IDS can help to solid protection of the network.IDS framework is categorized into two groups: An anomaly-based Intrusion Detection System and a Signature-based Intrusion Detection System.

3.3.1 Anomaly Detection

Anomaly detection intends to establish a model for the regular operation of the network. This technique is capable of detecting novel attacks. However, this can lead to false positives (new traffic that is legitimate) or false negatives (attacks newly disguised as legitimate traffic) [3].

A graphical representation of an anomaly detection algorithm's typical operation is given in Figure Figure 3.2.

3.3.2 Misuse/Signature Detection

Misuse detection intends to detect intrusions by matching traffic to specific strings of known attack patterns, unlike anomaly detection, which tries to identify network traffic that

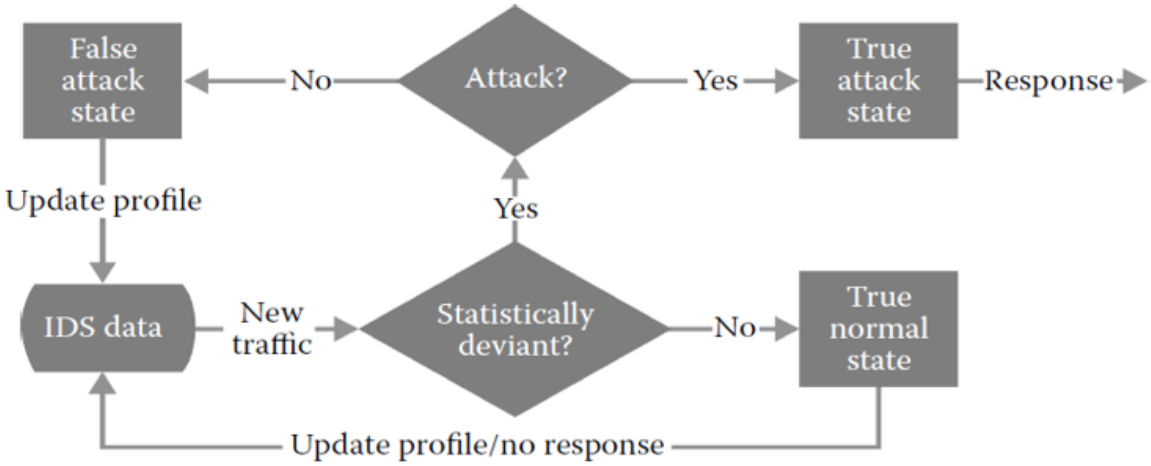


Figure 3.2: Anomaly detection process flow [3]

	Misuse-base	Anomaly-base
Method	Identify known attack patterns	Identify unusual activity patterns
Detection rate	High	Low
False alarm rate	Low	High
Detection of Unknown attack	Incapable	Capable
Drawback	Updating signatures is burdensome	Computing any machine learning is heavy

Table 3.1: Comparison of IDS types based on the methodology [1]

does not fall within its bounds. The technique is effective at identifying the known attacks Because signatures are developed from known attacks. However, this method is not able to identify novel intrusions.

Furthermore, developing these patterns is a complex and time-consuming process, will always face the incapability to perfectly replicate real life, which leads to false positives as in anomaly detection [3]. As Patterns tend to be developed from historical attack data, they are used less regularly, which causes the rules to become outdated. The detailed comparison of the two detection methods is shown in Table 3.1 A graphical representation of a misuse/signature detection algorithm's typical operation is given in Figure Figure 3.3.

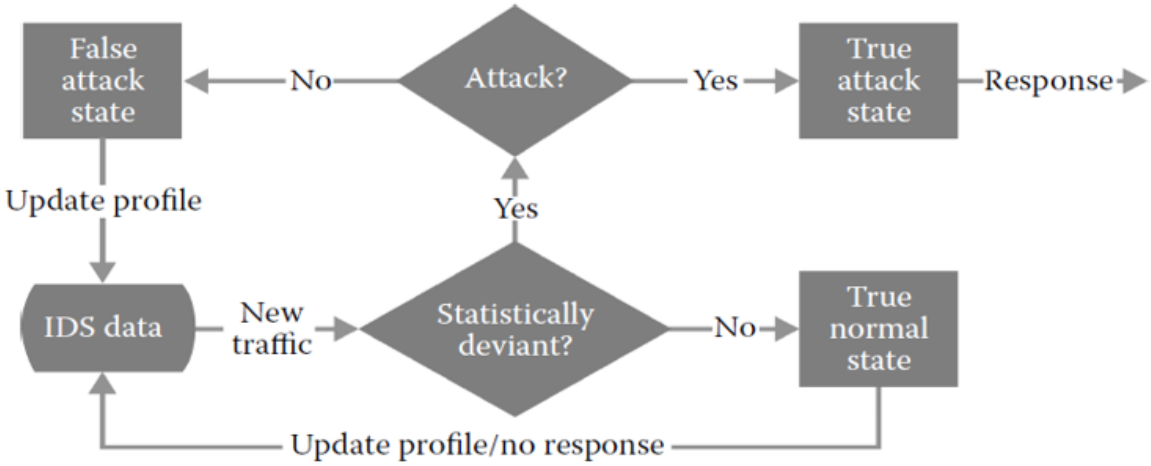


Figure 3.3: misuse detection process flow [3]

3.3.3 Advantage and Disadvantage of Intrusion Detection System (IDS)

3.3.4 Advantages of IDS

- IDS are easier to deploy as it does not affect existing systems or infrastructure.[7]
- Network-based IDS sensors can detect many attacks by checking the packet headers for any malicious attack like TCP SYN attack, fragmented packet attack etc. [7]
- IDS monitors traffic in a real-time. So, network-based IDS can detect malicious activity as they occur.[7]
- IDS sensor deployed outside the firewall can detect malicious attacks on resources behind the firewall [7].

3.3.5 Disadvantages of IDS

- IDS is not an alternative to strong user identification and authentication mechanism.[7]
- IDS is not a solution to all security concerns.[7]
- Human intervention is required to investigate the attack once it is detected and reported.[7]
- False positives occur when IDS incorrectly identifies normal activity as being malicious.[7]

- False negatives occur when IDS fails to detect the malicious activity [7].

3.3.6 State of The Art (SOTA) IDS

Research on security issues relating to Intrusion Detection Systems exists since the birth of computer architectures. In recent days, applying machine learning-based solutions to Intrusion Detection systems has gained enormous interest among security researchers and specialists. This section discusses some of the studies that explore the field of machine learning techniques used in different ways for detecting intrusions on the publicly available datasets such as AGGARWAL2015842'99 [11], DARPA 1998 [1]. These datasets have 41 features.

A hybrid Intrusion Detection System (IDS) architecture that utilizes anomaly and misuse detection approaches using AGGARWAL2015842 Cup 99 Dataset was proposed [12]. The architecture consists of an anomaly detection module that uses a Self-Organizing Map(SOM) structure to model expected behavior. Any deviation from normal behavior is considerd as an attack. The proposed misuse detection module's architecture uses the J.48 decision tree algorithm to classify different types of attacks. A rule based Decision Support System (DSS) is also developed for interpreting the results of both misuse and anomaly detection modules. They have shown that the proposed hybrid approach gives better performance over the individual approach.

Meryem et.al[13] tried to build an intrusion detection tool using machine learning and deep Learning approaches and compared both the tools for accuracy. The deep learning algorithm was then used to create a tool, and finally, the test data was inserted into the tools and checked the output with the expected output.

Bouzida et al.[14] performed the Intrusion Detection System using Back Propagation Neural Network (BPL NN) classifier and Decision tree separately for misuse detection. The experiment was carried out using the AGGARWAL2015842'99 dataset [2] both for training and

[1] https://archive.ll.mit.edu/ideval/files/kkendall_thesis.pdf
[2] http://AGGARWAL2015842.ics.uci.edu/databases/AGGARWAL2015842cup99/AGGARWAL2015842cup99.html

testing. The number of neurons in the input layer is defined as the number of input variables, and the number of neurons in the output layer is equal to the total number of classes. The Neural network architecture used considers only one hidden layer. As a result, the Neural Network was able to perform well for detecting DoS and Probe attacks. However, it fails to detect the low-frequency attacks because the number of records for these attacks is very less than other attacks (DoS and Probe).

Amoli et al.[15] proposed an unsupervised clustering method which is based on an anomaly detection approach to detect and classify the DoS, DDoS, Probe attacks. The model is composed of two detection engines that monitor and inspect the network's behavior in normal or encrypted communications. The first engine calculates a self-adaptive threshold value to detect the network traffic changes that are caused by attacks such as DoS, DDoS, scanning, and worm. The clustering is done in two steps: The network traffic does not pass the threshold, the engine clusters the attack-free traffic according to the DBSCAN algorithm. The clustering algorithm calculates the acceptable distance of the network instances and puts the points into the cluster. Once the traffic passes the threshold value, again clusters are created for outliers. The points that cross the acceptable distances are treated as outliers. The second engine aims to detect the bot-master. The first engine sends the IP addresses with attack details to the second engine, which then correlates the packets to find the main system controlling DoS.

Both of the above techniques consider a Single classifier using all 41 features of the dataset. Thus they are simple and easy to understand. However, they are sensitive to input parameters, choice of the kernel function, number of training variables, and overfitting. This reduces the total performance of these approaches; also, since they consider all features of the dataset, for a dataset with too many attributes, the classifier fails to provide timely results.

Gharaee et al.[16] proposed the new feature selection-based intrusion detection model (GF-SVM) to detect intrusions in the network. A feature selection approach is proposed where a Genetic algorithm (GA) and SVM are integrated to provide an optimal set of fea-

tures. The authors have done slightly modified the fitness function of the GA. Instead of using the accuracy and number of features (NumF) as fitness function parameters, they have used three parameters: True positive rate (TPR), False Positive Rate FPR, and NumF. Each chromosome is evaluated in each iteration of GA, and chromosomes with the highest classification accuracy (using SVM) are selected. The optimal features are used to filter the dataset, and Least Squared Support Vector Machine (LSSVM) is used to learn from the training dataset with selected features. They have considered seven features for normal attacks and 6-14 features for different types of attacks. Applying feature selection improves the performance of classification. However, one needs to consider which combination of feature selection and machine learning algorithm will provide the best results. It also makes the classifier faster over a selected set of attributes of features. However, there is less or moderate improvement in the classification results. Mukkamala et al.[17] proposed the ensemble approach, which integrates Artificial Neural Network (ANN), Multivariate Adaptive Regression Splines (MARS), and Support Vector Machine (SVM). The Multi-Layer Feed Forward algorithm is very robust in detecting anomalies. However, ANN suffers from the drawback of detecting low-frequency attacks, which are in limited numbers in the training data set, but ANN requires a huge dataset in training. SVM can perform well even on the small data set, but it is significantly slower than other classifiers. MARS is a mathematical process in finding the optimal variable transformations and interactions. It can find the complex data structures too that often hide in high dimensional space. The gathering of these classifiers reduces the mean squared error and increases the accuracy of the predictive model. In the initial phase, the data preprocessor obtains data from the DARPA 1998. Each classifier is trained over the data set, and individual learned classifiers are formed in the next phase. In the final phase, a majority voting scheme is applied to make the final decision over the test instance. The detected class is one in which the majority of the classifiers agreed. The approach is providing good accuracy for each category of attacks. The classifiers are trained to learn all features of the training data set. Although there is an improvement in the system's accuracy, the computational cost and complexity of the system are high. The detection rate

is improved by multiple classifier algorithms, especially for low-frequency attacks such as U2R and R2L.

Chandrasekhar et al.[18] proposed a hybrid Model which uses the power of Clustering (K-means), Fuzzy Neural Network (neuro-fuzzy), and Support Vector Machine (SVM) to identify the intrusions. Processing a huge chunk of data introduces errors and affects the efficiency of the classifier. Hence, in the initial phase, the proposed framework divides the training data set into small subsets based on the similarity of the data items by using the K-means clustering algorithm. It reduces the sparsity in data and makes it more suitable for the classifier. In the next phase, five neuro-fuzzy classifiers are trained over the five training subsets. It is difficult to determine the number of neurons and hidden layers in the Neural Network. The problem is overcome by introducing fuzzy logic with the neural network. It can manage imprecise, partial, and vague information. It uses the backpropagation algorithm to find out the input membership function. Each Neuro-Fuzzy Network outputs the set of features with the membership value. In the next phase, SVM is trained using the selected features for each of the training samples, and support vectors are generated. In the testing phase, SVM classifies the test instances based on the generated hyperplanes. The algorithm is performing well for detecting all types of attacks. Applying feature selection definitely improves the speed of classification, and in some cases, it is improving the detection rate also. However, the time complexity is not much reduced. The overall complexity is still high as it consists of multiple classifiers and feature selection algorithms.

Wang et al. [19] proposed an algorithm based on Convolutional Neural Network(CNN) that classifies malicious software traffic. By mapping the traffic characteristics to pixels, the network traffic image is generated, and the image is used as the input of the CNN to realize traffic classification. Torres et al. [20] at first converted network traffic characteristics into a series of characters and then used Recurrent Neural Network(RNN) to learn their temporal characteristics, which were further used to detect malicious network traffic.

Staudemeyer and Shams [21] proposed an intrusion detection algorithm based on Long Short-Term Memory(LSTM) that detects probe attacks and Denial of service (DoS) attacks

with unique time series using the AGGARWAL2015842 Cup99 dataset.

Tama et al. [22]A two-stage classifier ensemble for an intelligent anomaly-based intrusion detection system,"] proposed an anomaly-based IDS based on a two-stage meta-classifier, which uses a hybrid feature selection method to obtain accurate feature representations. The model had an improved detection rate, and the proposed method was conducted on the NSL-AGGARWAL2015842 and UNSW-NB15 intrusion datasets.

3.4 What is Machine learning

Machine Learning is one of the applications of artificial intelligence (AI) that provides systems the ability to automatically learn and improve the experience from the data's knowledge. Among the wide range of machine learning applications are regression, classification, prediction and clustering, recognition, etc. The commonly used machine learning algorithms include linear regression, Navie-Bayes classifier, logistic regression, support vector machines, artificial neural networks. With the quick progressions in cyber-attacks and accessibility of a tremendous amount of malicious data on the Internet, machine learning, data mining, and other related areas are most frequently used to address cybersecurity challenges.

Using machine learning techniques in intrusion detection systems is to create an IDS with improved accuracy and less requirement for human knowledge. Machine learning can be applied in signature detection, anomaly detection techniques of an intrusion detection system.

3.4.1 Machine Learning applications on IDS

Machine Learning algorithms are classified into different categories as reinforcement learning, supervised learning, and unsupervised learning[23]. Supervised learning is where a label that needs to be predicted has already had values; thus, we know what the output will be like for different instances. Unsupervised learning will be used where we need to find the implicit link in an unlabeled data set. Reinforcement learning will be in neither of the above two, and we will be able to find out some information for any action, but without any message.

Researches in the area of network intrusion detection using machine learning approach, scholars mainly distinguish normal network traffic from abnormal network traffic by clustering, dimensionality reduction, and classification to realize the identification of malicious attacks [24].

One of the major challenges in building an effective intrusion detection algorithm is the difficulty to develop appropriate rules or profiles as these must be both specific enough to identify attacks among normal traffic and general enough to be used in many different scenarios, locations, and network environments [3]. Relying on the abilities of individual human beings to design detection characteristics is highly dependent on the particular knowledge, experience, and beliefs of the individuals that may vary considerably.

Machine learning is an area of research in the field of artificial intelligence that aims to alleviate this problem by providing the algorithm with a composition of training data. The data must be real traffic but can also have specific types of attacks added in order to bias detection ability.

For an Intrusion detection system, which is built Within a machine learning approach, the machine can be taught to detect attacks within the data in any one of three ways:

- Supervised
- Unsupervised
- Semi-supervised

3.4.2 Supervised Machine learning

Supervised learning-based IDS techniques detect intrusions by using labeled training data. A supervised learning approach usually consists of two stages, namely training and testing. In the training stage, relevant features and classes are identified, and then the algorithm learns from these data samples. In supervised learning IDS, each record is a pair containing a network or host data source and an associated output value (i.e., label), namely intrusion or normal.

It depends on a human expert to train the machine's learning process to learn well and determine which set of data indicate an attack and which indicates normal traffic. In supervised learning, the whole data is labeled as either normal or attack by a human expert. This data will be useful to the machine to form thresholds, clusters, states, or relationships to generate a set of rules or profiles [3]. This learning technique's benefit is that it allows the machine to make connections that could be sophisticated for a human to identify. This technique also allows for constant, automatic updating of the detection parameters as more traffic travels through the network [25]. one of the drawbacks of supervised learning is that the need for a human expert to identify, categorize and label the Normal and abnormal traffic, which is a not an easy task. The labeling of the data by Each expert might be done differently or suboptimally.

3.4.3 Unsupervised Machine learning

Unsupervised learning is a form of machine learning technique used to obtain interesting information from input datasets without class labels. The input data points are normally treated as a set of random variables. A joint density model is then created for the data set. In supervised learning, the output labels are given and used to train the machine to get the required results for an unseen data point, while in unsupervised learning, no labels are given, and instead, the data is grouped automatically into various classes through the learning process. In developing an IDS, unsupervised learning means using a mechanism to identify intrusions by using unlabelled data to train the model.

This approach relies on the assumption that normal network traffic is appreciatively distinct from abnormal traffic, and the machine can use these patterns to differentiate between the two without the help of human expert [3]. One advantage of this system is that no need of human intervention is needed, and it is able to detect novel attacks. This system's drawbacks include that it is challenging to provide root causes for detection alerts as the rules and profiles generated may be complicated for a human to interpret quickly.

3.4.4 Semi-Supervised Machine learning

Semi-supervised learning falls between supervised learning (with totally labeled training data) and unsupervised learning (without any categorized training data). Researchers have shown that semi-supervised learning could be used in conjunction with a small amount of labeled data classifier's performance for the IDSs with less time and costs needed.

Semi-supervised learning is a midway point between supervised and unsupervised learning. Only the conclusively known or a subset of conclusively known traffic is labeled by a human in this approach. This reduces the labeling burden on the human and does not require labeling of complex or distributed attacks, which can be time-consuming. This allows the system to create parameters for suspicious or attack activities and can potentially differentiate between different attacks rather than normal and abnormal. Nonetheless, the identification of conclusively good traffic is still a difficult task for a human to carry out correctly. Identifying "anomalous" rather than attack traffic has been discussed .

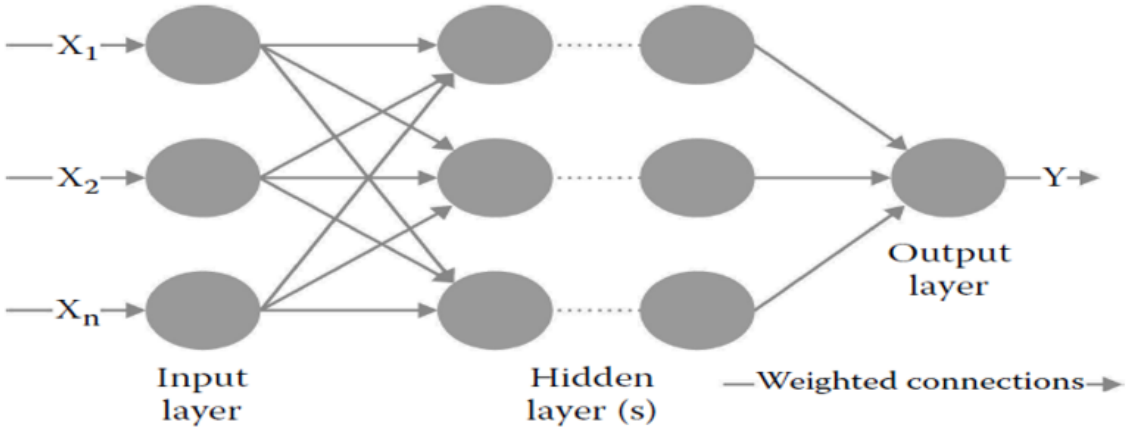


Figure 3.4: Multilayer perception [3]

3.4.5 What is deep learning

Deep Learning is a complex version of machine learning with multiple levels of abstraction of data at multiple processing layers [26]. Deep Learning can learn the complicated structures within the data set through backpropagation and demonstrates how machine

changes the inside parameters at each layer. Artificial Neural Networks (ANN) containing one or more hidden layers will make the structure deep, and the data is processed at each layer, thus, making the learning task deeper.

The commonly used deep learning architectures include deep belief networks (DBN), deep neural networks 12(DNN), and recurrent neural networks (RNN), which are applied to research fields such as natural language processing, speech recognition, computer vision, audio recognition, machine translation, and social network filtering. Recurrent neural networks (RNN) in deep learning have the tendency to learn from the previous time-steps and can be used with less human intervention [27]. In RNN, each node's output in the hidden layer is used as input to the same node at each time-step. The valuable information is stored in the memory and can be used for learning purposes in future time steps.

Deep learning replicates the functions of the human brain for learning and training itself to produce predictive models, i.e., in this thesis work, a classifier that is capable of distinguishing between attacks and normal connections. It is composed of an artificial neural network with an input layer followed by many hidden layers and a final output layer. It works like neurons in our brain. The neurons get signals from dendrites; it processes it and sends signals to the axon. Similarly, the deep learning model also has nodes or artificial neurons that take input and weight processes them using a function and send out outputs [27]. It is an approach to build accurate learning methods for a better prediction. Deep learning explains how we make some important decisions, which the systems extract from the data feed, and it is through neural methods similar to machine learning. These methods are logical ones posting some sets of binary questions, which can be right or wrong, extracting every piece of data fed into the system and putting them across the answers. Thus, such advanced technology can make it more intelligent when used for intrusion detection and help build better security tools.

Deep learning uses artificial neural networks which provide automatic extraction in feature engineering. They are straightforward to handle, as you just need to feed the raw images in artificial neural networks, and they provide incredible outputs.

Deep learning has made remarkable achievements in the fields of Computer Vision(CV), Autonomous driving(AD), Natural Language Processing(NLP) with the powerful ability of feature extraction [20]. Many scholars apply deep learning to intrusion detection to mine the potential characteristics of high-dimensional data through a training model and transform network traffic anomaly detection into a classification problem [deep learning approach for network intrusion detection system]. Through a large number of sample data training, adaptive learning between normal network traffic and abnormal network traffic effectively enhances real-time intrusion processing.

3.4.5.1 Recurrent Neural Networks (RNN)

It is an extension of neural networks with cyclic links to process sequential information. These cyclic links are placed between higher and lower-layer neurons, enabling RNN to propagate data from previous to current events. This property makes RNN having a memory of time series events.

One advantage of RNN is the ability to connect previous information to the present task; however, it cannot reach ”far” previous memory. This problem is commonly known as long-term dependencies. Long Short-Term Memory (LSTM) networks are introduced [1] to overcome this problem.

3.4.5.2 Long-Short-Term Memory RNN (LSTM)

Researchers came up with the Long-Short-Term-Memory (LSTM) network to overcome this vanishing gradient problem, which bridges the minimal time gaps. LSTM makes use of a gating mechanism to handle long-term dependencies [1] .

LSTMs are an extension of RNN with four neural networks in a single layer, where RNN have only one The main advantage of LSTM is the existence of a state cell in which the line is passing through at the top of every layer. The cell accounts for propagating information from the previous layer to the next one. Then, ”gates” in LSTM would manage which information will be passed or dropped. [28] Recurrent Neural Networks (RNN), when trained in

real-time, learn from previous time-steps by backpropagation through time (BPTT). A deep neural network is unfolded in time and constructs an FNN for every time-step. Then, the gradient rule updates each hidden layer's weights and biases, thus minimizing the loss between the expected and actual outputs. However, standard RNNs cannot perform better when the time-steps are more than 5-10. The prolonged backpropagation leads to vanishing or blow-up of error signals, leading to oscillating weights, which makes the network performance poor.

LSTM has a cell state which is passed to every-time step. A gating mechanism is used to optimize the information that is passing through. It contains a sigmoid function layer that outputs between one and zero. A value of one means "pass all the information through," whereas the value of zero means "do not pass any information through." The "forget gate" decides the information that needs to be let through between the current input and previous cell state output using the sigmoid function. The "input gate" decides what information is required to store in the cell state. This gate contains two functions - "sigmoid" to decide what values need to be updated and the "tanh" function to create a new vector of values that are to be added to the cell state. The "output gate" decides on what information from the cell state is required to output with the help of a sigmoid function. The output information is passed through the "tanh" function before passing through the "sigmoid" to make sure that the values are between -1 and +1.

LSTM is an artificial recurrent neural network (RNN) architecture used in the field of deep learning [29]. It has feedback connections, Unlike standard feed-forward neural networks. It can also process entire sequences of data (such as speech or video), not only single data points (such as images). LSTM is applicable to tasks such as speech recognition, connected handwriting recognition, and anomaly detection in intrusion detection systems (IDS) A standard LSTM unit comprises a cell, an input gate, an output gate, and a forget gate. The cell remembers values over arbitrary time intervals, and the three gates regulate the flow of information into and out of the cell.

LSTM networks are well suited to processing, classifying, and making predictions based

on time series data since there can be lags of unknown duration between essential events in a time series [29]. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional RNNs. Relative insensitivity to gap length is advantageous for LSTM over RNNs, hidden Markov models, and other sequence learning methods in numerous applications.

Hochreiter and Jurgen et al. proposed the Long Short-Term Memory(LSTM) network [30].which is a Recurrent Neural Network(RNN) structure. The LSTM network is universal, Like most RNN, because as long as there is a suitable weight matrix, the LSTM network is able to calculate any network element that any conventional computer can calculate.

The LSTM network is very suitable for learning from experience, which is different from the traditional RNN. When there is a time lag of unknown size and boundary between essential events, the time series can be classified, processed, and predicted. LSTM is not sensitive to gap length and has advantages over other RNN and hidden Markov models and other sequence learning methods in many applications [31] The problem of gradient disappearance and gradient explosion is solved by introducing the gate structure and storage unit.

3.4.6 Random Forest Classifier

Random Forest Classifier is an ensemble machine learning technique for supervised learning tasks. It is one of the types of classification algorithms that are usually used for a large amount of data and unbalanced data. Random Forest Classifier is used to select features with top ranks. It evaluates the importance of the features. Selecting the features with ”higher importance” is relevant for the model performance. This algorithm provides higher accuracy and is used widely. When there are a lot of missing data in the data set, this algorithm provides one of the best results as compared to others. It can also handle a large number of input variables for prediction purposes. It can also be used to find out which variables are the most important. The advantages of Random Forest a[32]

- Ability to handle many input variables without a necessity for variable deletion.

- Provides estimates of important variables for the classification.
- Lightweight when compared to other boosting methods.
- when compared to single classifiers it is Robust to noise and outliers

Leo Breiman et al. proposed that random [32] is an excellent supervised learning algorithm that can train a model to predict which classification results in a certain sample type belong to based on a given dataset's characteristic attributes and classification results. Random Forest is based on a decision tree and adopts the Bagging(Bootstrap aggregating) method to create different training sample sets. The random subspace division strategy selects the best attribute from some randomly selected attributes to split internal nodes. The various decision trees formed are used as weak classifiers, and multiple weak classifiers form a robust classifier, and the voting mechanism is used to classify the input samples. After a random forest has established a large number of decision trees according to a certain random rule when a new set of samples is input, each decision tree in the forest makes a prediction on this set of samples separately, and integrates the prediction results of each tree, get a final result.

3.5 Dataset

Most of the datasets which are used for network packet analysis are not easily accessible due to the issue of privacy and security. However, some freely available datasets such as DARPA, AGGARWAL2015842, NSL-AGGARWAL2015842, and ADFA-LD are used as benchmarks. In this thesis work, the DARPA AGGARWAL2015842 Cup '99 data set is used Since it is the most widely used Data set on the application of machine learning algorithms on the intrusion detection system.

The DARPA AGGARWAL2015842 Cup '99 datasets were generated by the Defense Advanced Research Projects Agency (DARPA ITO) on a simulated air force model. The collected network packets were around four gigabytes containing about 4,900,000 records. The

test data of 2 weeks had around 2 million connection records, each of which had 41 features [11].

The complete network traffic is either classified as one of the attack types or ”normal”. The datasets can be found on the UCI website, where the repository links to the three different data set versions that exist. The three versions of the AGGARWAL2015842 99’Cup IDS datasets are – full AGGARWAL2015842 data set, corrected AGGARWAL2015842, 10 % AGGARWAL2015842. In this thesis work, 10 % AGGARWAL2015842 data set is used as in most literature.

The 10 % AGGARWAL2015842 dataset contains 24 attack types, which are mainly categorized into four classes – Probe, Denial of Service(DoS), User to Root (U2R), and Remote to Local (R2L). Those categories of attack are described as following:

Denial of Service Attack (DoS): A Denial-of-Service attack is one that aims at shutting down the network, rendering it inaccessible to its intended users[4]. DOS attacks do this by flooding the destination with traffic or sending information, triggering a crash. The attacker refuses the valid users’ access, or the attacker makes the memory full, so legitimate users cannot preprocess their requests [1IDS]. e.g., syn flood;

User to Root Attack (U2R): In this type of attack, the attacker has local access to the victim’s machine and tries to gain superuser benefits. With a normal user account, the attacker initiates the access to the system and can then use the system vulnerabilities to achieve root access [4]. Unauthorized access to local superuser (root) privileges. The attacker acquires access to the regular user’s identity through various methods, thus exploiting this access to get the system [1 application of deep learning and machine learning]. e.g., various ”buffer overflow” attacks.

Remote to Local Attack (R2L): The attacker has no account on the victim machine and sends packets over a network to that machine and uses system vulnerabilities as a user of that machine to gain local access [4]. Unauthorized access from a remote machine in this type of attack, the attacker has the right to send the package through the network. However, it does not have to access the machine. They utilize this ability to send packages over the network

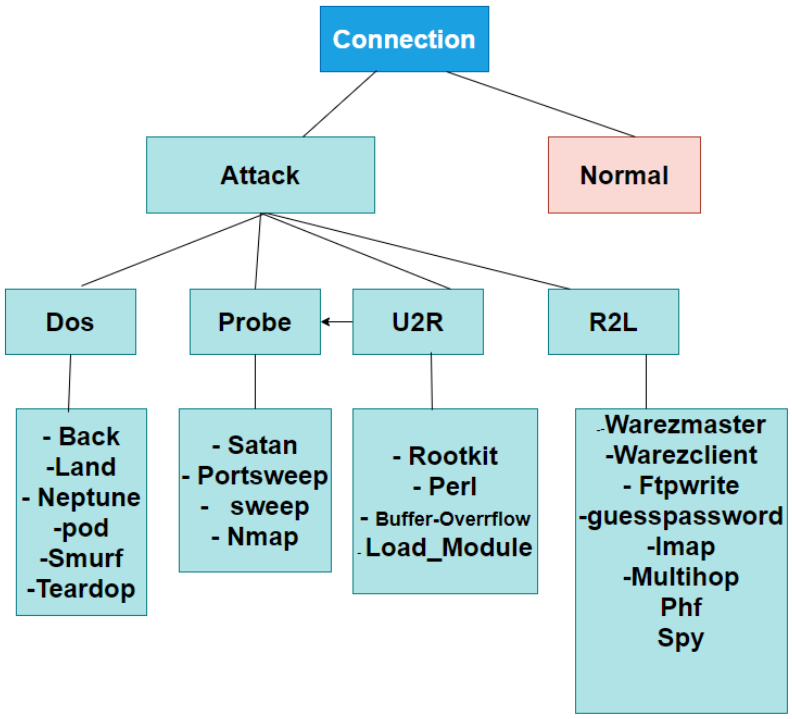


Figure 3.5: Types of attack present in AGGARWAL2015842 cup '99 dataset [4]

to get access to the system as a user illegally and thus exploit the system. e.g., guessing password;

Probing Attack: The probe is an attack category in which attackers browses a network together with information about the target system before initiating an attack [4]. The attacker collects information regarding the network of computers to find a way around the security measures taken. e.g., port scanning. Figure 3.5 shows the classification of the categories of attacks in the AGGARWAL2015842 99 dataset.

The training and testing samples are represented with 41 features and a label with either "normal" or "attack type." The intrusion detection data set randomly selected a specific amount of records separated them for training and testing the models. The test data is not from the same probability distribution as the training data, and it includes specific attack types not seen in the training data. This makes the task more realistic. [11] Some intrusion experts believe that most novel attacks are variants of known attacks, and the "signature" of known attacks can be sufficient to catch novel variants. The data sets contain a total of 24 training attack types, with an additional 14 types in the test data only.

The 1998 DARPA Intrusion Detection Evaluation Program was prepared and managed by MIT Lincoln Labs. The main objective was to survey and evaluate research on intrusion detection. A standard set of data which includes a diverse intrusion simulated in a military network environment, was provided. The 1999 AGGARWAL2015842 intrusion detection contest uses a version of this dataset [11].

Lincoln Labs has set up an environment to acquire raw TCP dump data for a local-area network (LAN) simulating a typical U.S. Air Force LAN. They operated the LAN as if it were a true Air Force environment, although they peppered it with multiple attacks.

A single connection is a sequence of TCP packets starting and ending at some well-defined times, between which data flows to and from a source IP address to a target IP address under some well-defined protocol. Each connection is also labeled as either normal or an attack, with exactly one specific attack type. Each connection record is consists of about 100 bytes.

The dataset features discription is persented in Figure 3.6.

Attribute Number	Features	Description
1	duration	Length of the time duration of the connection
2	protocol_type	Protocol used
3	service	Service used by destination network
4	flag	Status of the connection (Error or Normal)
5	src_bytes	Number of data bytes transferred from source to destination
6	dst_bytes	Number of data bytes transferred from destination to source
7	land	If source and destination port no. and IP addresses are same then it will set as 1 otherwise 0
8	wrong_fragment	Total number of wrong fragments in a connection
9	urgent	Number of urgent packets (these packets with urgent bit activated)
10	hot	Number of 'hot' indicators means entering in a system directory
11	num_failed_logins	Number of failed login attempts
12	logged_in	Shows login status (1- successful login, 0- otherwise)
13	num_compromised	Number of compromised conditions
14	root_shell	Shows root shell status (1-if root shell obtained otherwise 0)
15	su_attempted	Set as 1 if 'su_root' command used otherwise set as 0
16	num_root	Number of operations performed as root
17	num_file_creations	Number of file creation operations
18	num_shells	Number of shell prompts in a connection
19	num_access_files	Number of operations on access control files
20	num_outbound_cmds	Number of outbound commands in a ftp session
21	is_host_login	If login as root or admin then this set as 1 otherwise 0
22	is_guest_login	Set as 1 if login as guest otherwise 0
23	count	Number of connections to the same destination host
24	srv_count	Number of connection to the same service (port number)
25	serror_rate	Percentage of connections that have activated flag (#4) s0,s1,s2 or s3, among the connections aggregated in count (#23)
26	srv_serror_rate	Percentage of connection that have activated flag (#4) s0,s1,s2 or s3, among the connections aggregated in srv_count (#24)
27	rerror_rate	Percentage of connections that have activated flag (#4) REJ, among the connections aggregated in count (#23)
28	srv_rerror_rate	Percentage of connections that have activated flag (#4) REJ, among the connections aggregated in srv_count (#24)
29	same_srv_rate	Percentage of connections that were to the same services, among the connections aggregated in count (#23)
30	diff_srv_rate	Percentage of connections that were to the different services, among the connections aggregated in count (#23)
31	srv_diff_host_rate	Percentage of connections that were to different destination machines among the connections aggregated in srv_count (#24)
32	dst_host_count	Number of connections having the same destination host IP address
33	dst_host_srv_count	Number of connections having same port number
34	dst_host_same_srv_rate	Percentage of connections that were to the same service among the connections aggregated in dst_host_count (#32)
35	dst_host_diff_srv_rate	Percentage of connections that were to different service among the connections aggregated in dst_host_count (#32)
36	dst_host_same_src_port_rate	Percentage of connections that were to the same source port among the connections aggregated in dst_host_srv_count (#33)
37	dst_host_srv_diff_host_rate	Percentage of connections that were to the different destination machines among the connections aggregated in

Attribute Number	Feature	Description
38	dst_host_serror_rate	Percentage of connections that have activated flag (#4) s0,s1,s2 or s3, among the connections aggregated in dst_host_count (#32)
39	dst_host_srv_serror_rate	Percentage of connections that have activated flag (#4) s0,s1,s2 or s3, among the connections aggregated in dst_host_srv_count (#33)
40	dst_host_rerror_rate	Percentage of connections that have activated flag (#4) REJ, among the connections aggregated in dst_host_count (#32)
41	dst_host_srv_rerror_rate	Percentage of connections that have activated flag (#4) REJ, among the connections aggregated in dst_host_srv_count (#32)
42	label	Attack class label

Figure 3.6: KDD Cup'99 Data set Features List with Description [5].

Chapter 4

Analytical Work

4.1 System Design

The data set used in this thesis work is KDD Cup '99. As the first step of this thesis work, Data preparation takes place. This includes the split of the data set into different layers of TCP/IP based on the attack types present at the respective layers. Next, data prepossessing is needed in order to transform the unprocessed raw data into a comprehensible format. In the next step, the data set's unique features are extracted using the Random forest classifier techniques, and those extracted features will be used as input data to the deep learning model. The data set is then split into two portions, the training portion, and the testing portion. Then, the proposed model learns through the training data set and then later evaluated and checked for accuracy using the testing data set to evaluate if the model can classify the data as normal and attack. The pipeline of the overall system is shown in Figure 4.1.

4.1.1 Data Preparation:

The 10% KDD dataset is selected to be used to train and test the machine learning algorithm. Then, the data set is split into different layers based on the attack types at the TCP/IP layers. The Data Link Layer is not considered part of this thesis work because there is no attack type in the data set that falls into the data Link Layer category. All attack types in the

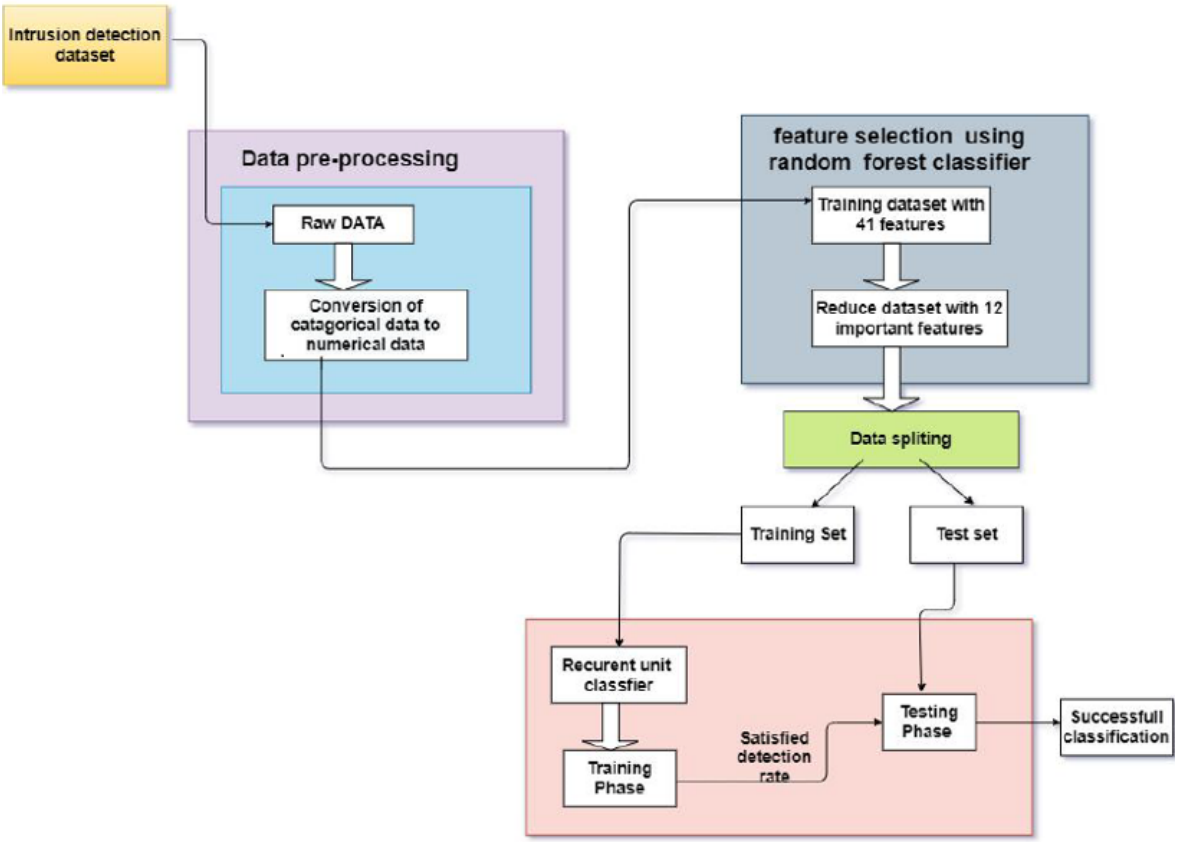


Figure 4.1: End to End data flow of the proposed IDS Model. Own Work

data set to fall under one of the three TCP/IP layer categories as shown in Table Table 4.1.

For the Application Layer, the original number of records in the training dataset before removing duplicates is 382266, and the number of records in the training dataset after removing the duplicates is 90695. For the transport layer, the Original number of records in the training dataset before removing duplicates is 206780, and a number of records in the training dataset after removing the duplicates resulted in 141193. For the Network layer, the Original number of records in the training dataset before removing duplicates is 379609, and a number of records in the training dataset after removing the duplicates resulted in 89360.

Layer	Attack Type
Application Layer	Smurf, Buffer-Overrflow, Load_module, Warezmaster, Perl, Nmap, pod, back Guess password, satan, impa, ftp_write, multiple and normal.
Transport layer	Neptune, land, teardrope, port sweep, buffer overflow, Nmap and normal
Network Layer	overflow, smurf, pod, IP sweep and normal

Table 4.1: Attack Types for each TCP/IP Layer. Own work

4.1.2 Data Pre-processing

Data pre-processing is a technique of Transforming data into a useful format in order to build a predictive model. It is extracting only the required data and cleaning the data set after gathering the data. Next, the data pre-processing techniques step is followed as the original data could be very noisy, contain duplicate values, and miss values resulting from extraction errors or input errors. To make the data consistent, those types of data should be removed.

The data set contains three categorical features which need to be encoded into numerical form before they are provided as input to the neural network architecture. The features ”protocol-type”, ”service,” and ”flag” are encoded to numerical values. For example, “TCP”, “UDP” and “ICMP” are functions of protocol types. After One Hot encoding, they become binary vectors (1, 0, 0), (0, 1, 0), (0, 0, 1) So that this will be suitable input to the algorithm model.

4.1.3 Feature selection

Is the process of automatic selection of attributes in the data. Reducing the dataset when developing a predictive model Some of the advantages of Feature selection include Reduction of Data Redundancy to avoid unwanted calculations on the useless features by selecting only useful attributes. Feature selection also helps to Remove the probability of Over-Fitting by avoiding the correlated attributes. The model’s overall accuracy can also be enhanced, and the computational cost of the model can be minimized by avoiding the useless feature and Reducing the number of input variables. In this thesis work, the 41 features of the KDD CUP dataset are reduced to 12 important features. The random forest classifier is used as the feature selection technique, which is proven to reduce the dimensionality for the KDD’99 Cup data set [33].

4.2 Algorithm implementation (predictive model)

The data set used is divided into training and test data. For every data set, 70% of the data is considered the training data, and 30% of the data is considered the testing data. Each dataset is later divided into a feature set and the corresponding label set. The label ”normal” is encoded as [0 1] and ”all other attack types” as [1 0]. The training dataset consists of known output, and the developed model acquires knowledge on the data so that it will be included in other data in the upcoming stage [34]. The test dataset is then used to test the model ”s prediction capacity.

4.2.1 Intrusion Detection Model

The implemented model is based on a deep learning approach containing more than one hidden layer. The training dataset is used to train the system, and the testing dataset is used to test the performance of the system if it can classify the data as normal and attack. The output of the implemented system will be either normal or an attack. If the output is attack, the data will fall under any one of the following attacks i.e., Denial of Service, probe, R2L, and U2R.

The output layer uses the softmax activation function [35]. This function calculates and evaluates the probability distribution of the different events over ”n” various events. The softmax function will work out each target class’s probabilities on top of all possible target classes. This function’s output is similar to a well-known categorical probability distribution; it provides the probability that any of the other classes are true. Mathematical representation of a softmax function is shown in Equation 4.1 [35].

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{4.1}$$

Where:

- σ = softmax

- $\vec{Z}$ = input vector
- e^{zi} =standard exponential function for input vector
- K=number of class in the multi-class classifier
- e^{z}=standard exponential function for output vector

The calculated probabilities will be helpful in the later phase for evaluating the target class for the given inputs. The training cycle(epoch) value was set to 1000 and the optimization function used is Adam optimizer. Instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data.

4.3 Evaluation Metrics

In the field of machine learning, a confusion matrix is a specific table layout that allows visualization of the performance of an algorithm in a tabular form. This thesis work, it is used to evaluate the performance of the classification model.

Table 4.2 depicts the confusion matrix for a two-class classifier which can be used for evaluating the performance of an IDS. In the matrix, each column represents the instances in a predicted class, and each row represents the instances in an actual class. IDS are typically evaluated based on the following standard performance measures:

	Predicted As Normal	Predicted As Attack
Actually As Normal	TN	FP
Actually As Attack	FN	TP

Table 4.2: Representation of Confusion Matrix[2]

.

Where,

- **True Positive (TP)**:

  is the total number of samples predicted by the model as ”attack” while they were ”attack”.

- **False Negative (FN)**: is the total number of samples predicted by the model as ”normal” while they were ”attack”.

- **False Positive (FP)**: is the total number of samples predicted by the model as ”attack” while they were ”normal”.

- **True Negative (TN)**: It is the total number of samples predicted by the model as ”normal” while they were ”normal”.

- **Accuracy**:

  It measures how accurate the IDS is in detecting normal or anomalous traffic behavior. It estimates the ratio of the correctly recognized connection records to the entire test dataset.A better predictive machine learning model should have higher accuracy (Accuracy $\in$ [0; 1]). Accuracy is defined as follows Equation 4.2 [2].

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.2}$$

- **Precision** :

  It estimates the correctly identified attack connection records' ratio to the number of all identified attack connection records. Abetter, machine learning model, should have higher Precision (Precision $\in$ [0; 1]). Precision is defined as follows in Equation 4.3 [2].

  Number of predicted intrusions that were intrusions.

$$Precision = \frac{TP}{TP + FP} \tag{4.3}$$

- **Recall**:

  It is also called True Positive Rate (TPR): It estimates the ratio of the correctly classified Attack connection records to the total number of Attack connection records. If the

TPR is higher, the machine learning model is better (TPR $\in$ [0; 1]). TPR is defined ass follows in Equation 4.4 [2].

$$Recall = \frac{TP}{TP + FN} \tag{4.4}$$

- **Specificity**

$$Specificity = \frac{TN}{TN + FP} \tag{4.5}$$

- **F1 score**

F1-Score is also called as F1-Measure. It is the harmonic mean of Precision and Recall. If the F1-Score is higher, the machine learning model is better (F1-core $\in$[0; 1]). F1-Score is defined as follows in Equation 4.6 [2].

$$F1\ Score = \frac{2 * (Precision * Recall)}{Precision + Recall} \tag{4.6}$$

- **False Alarm Rate**

False Alarm Rate (FAR): Also known as False Positive Rate, it estimates the Normal connection records' ratio flagged as Attacks to the total number of Normal connection records. If the FPR is lower, the machine learning model is better (FPR $\in$ [0; 1]). FPR is defined as follows Equation 4.7 [2].

$$False\ alarm\ rate(FAR) = \frac{FP}{FP + TN} \tag{4.7}$$

Chapter 5

RESULTS AND DISCUSSION

5.1 RESULT

5.1.1 Analysis of Application layer IDS

Figure 5.1 depicts the important of each features of the Application Layer dataset . The first 12 features starting from left side are the highly important features which were reduced from 41 to 12 .

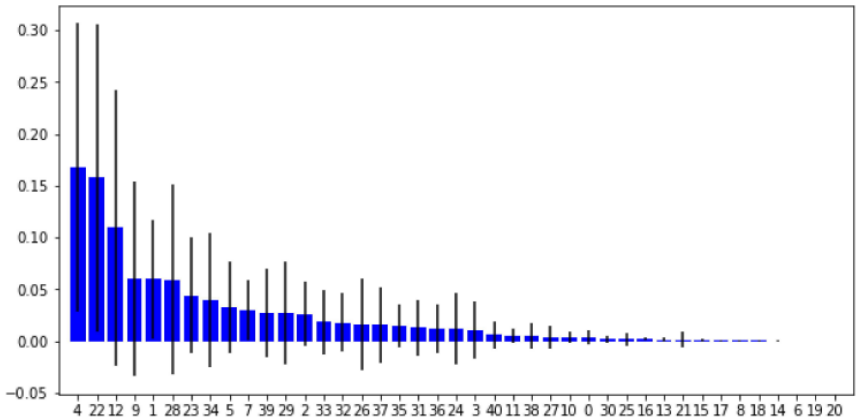


Figure 5.1: Application layer Feature importances (Y-axis) Vs Features IDS (X-axis). Own Visualization.

As shown in Figure 5.2 the training loss at each training cycle (epoch) is decreasing towards zero. in the other hand Figure 5.3 depicts the Validation accuracy at each training cycle (epoch) is increasing towards one. This shows that the application layer IDS training

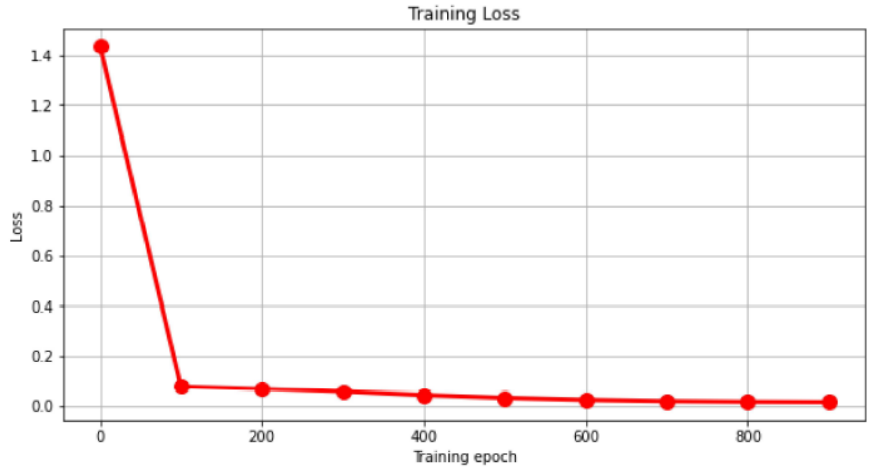


Figure 5.2: The Training Loss of Application Layer IDS at each training cycle(epoch). Own Visualization.

process performed well, i.e decreasing the training loss while increasing the validation accuracy without model over-fitting which results better generalization capability of the network model.

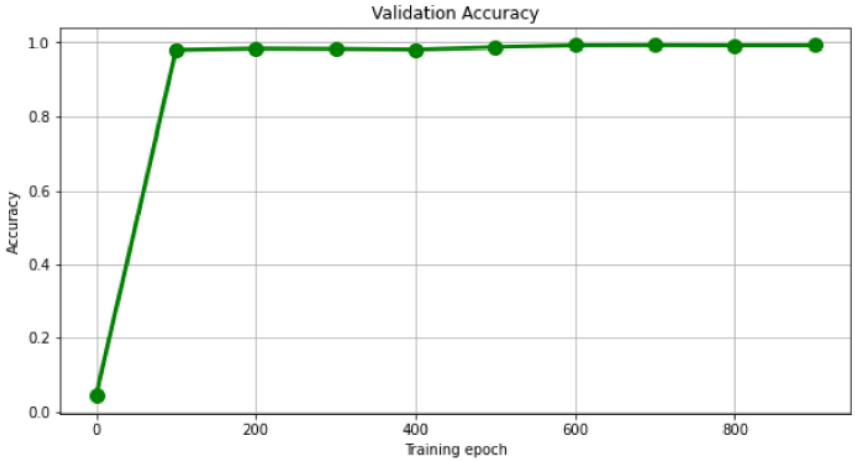


Figure 5.3: Accuracy of the Application layer dataset with respect to the number of training cycle. Own Visualization.

Table 5.1 depicts the percentage values of the The calculated Metrics.

	Predicted As Normal	Predicted As Attack
Actually As Normal	**TN=19774**	**FP=66**
Actually as Attack	**FN=48**	**TP=801**

Table 5.1: Numerical Representation of Confusion Matrix for Application Layer. Own Work

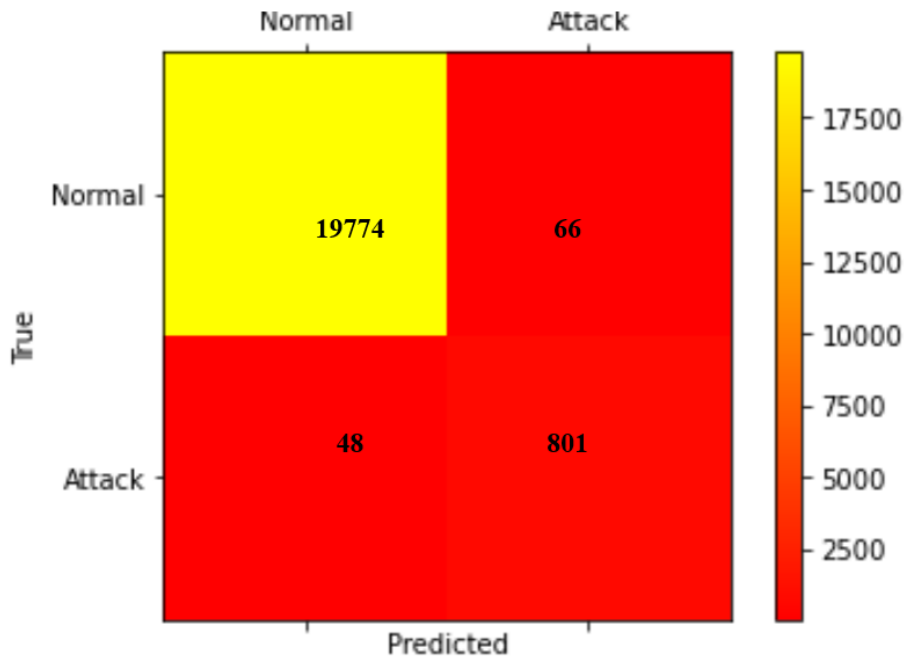


Figure 5.4: Heat Map of Application Layer Confusion Matrix. Own Visualization.

	Evaluation Metric	Performance out of 1	Performance in %
1	Accuracy ⇑	0.9944	99.44%
2	Precision ⇑	0.9239	92.39%
3	Recall ⇑	0.9435	94.35%
4	Specificity ⇑	0.9967	99.67%
5	F1 score ⇑	0.9336	93.36%
6	False alarm rate ⇓	0.0033	0.33 %

Table 5.2: Qualitative Result of Application layer IDS using the evaluation metrics. Own Work

As shown in Table 5.2 The application layer IDS has Recall value (Detection Rate) of 94.35 % and False alarm rate(False positive rate) of 0.33 % .This shows the modeled IDS has achieved a better performance while keeping the false alarm rate low. In the table the upper arrows shows the higher the values the better the result is and the down arrow shows the lower the value the better the result .

5.1.2 Analysis of Transport Layer

Figure 5.5 depicts the important of each features of the Transport Layer dataset . The first 12 features starting from left side are the highly important features which were reduced from

41 to 12 .

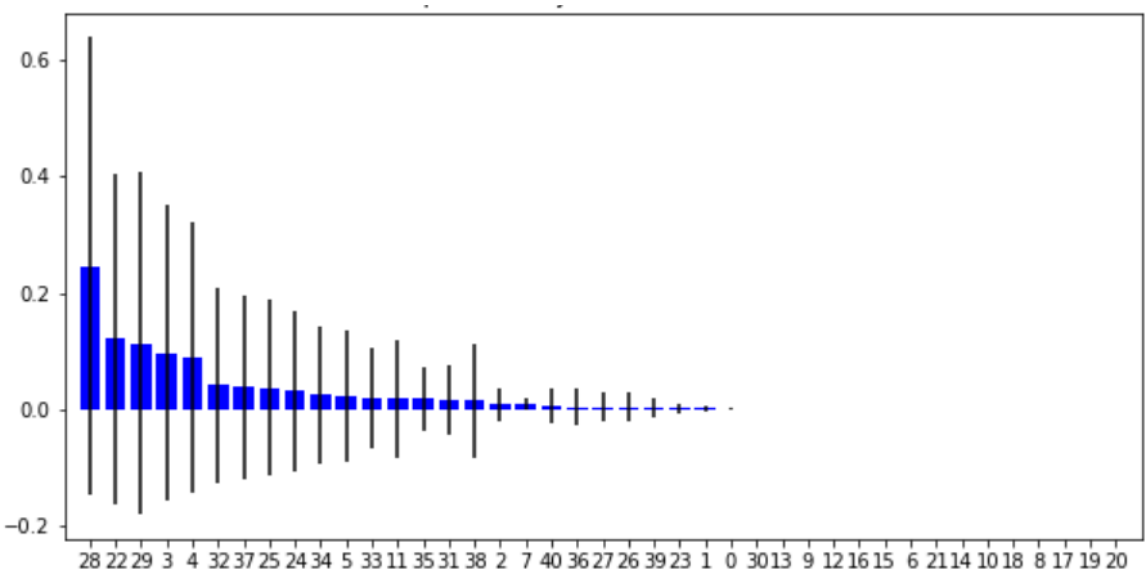


Figure 5.5: Trasport Layer Feature importance (Y-axis) VS feature IDS (X-axis). Own Visualization.

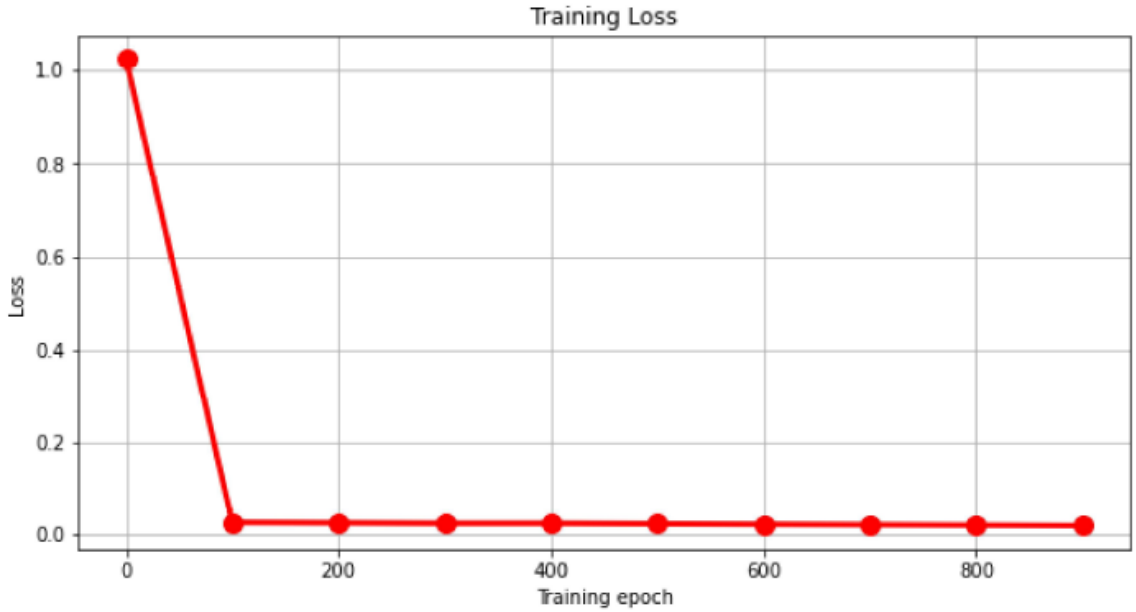


Figure 5.6: The Training Loss of Transport Layer IDS at each training cycle (epoch). Own Visualization.

As shown in Figure 5.6 the training loss at each training cycle (epoch) is decreasing towards zero. in the other hand Figure 5.7 depicts the Validation accuracy at each training cycle (epoch) is increasing towards one. This shows that the Transport layer IDS train-

ing process performed well, i.e decreasing the training loss while increasing the validation accuracy without model over-fitting which results in better generalization capability of the network model.

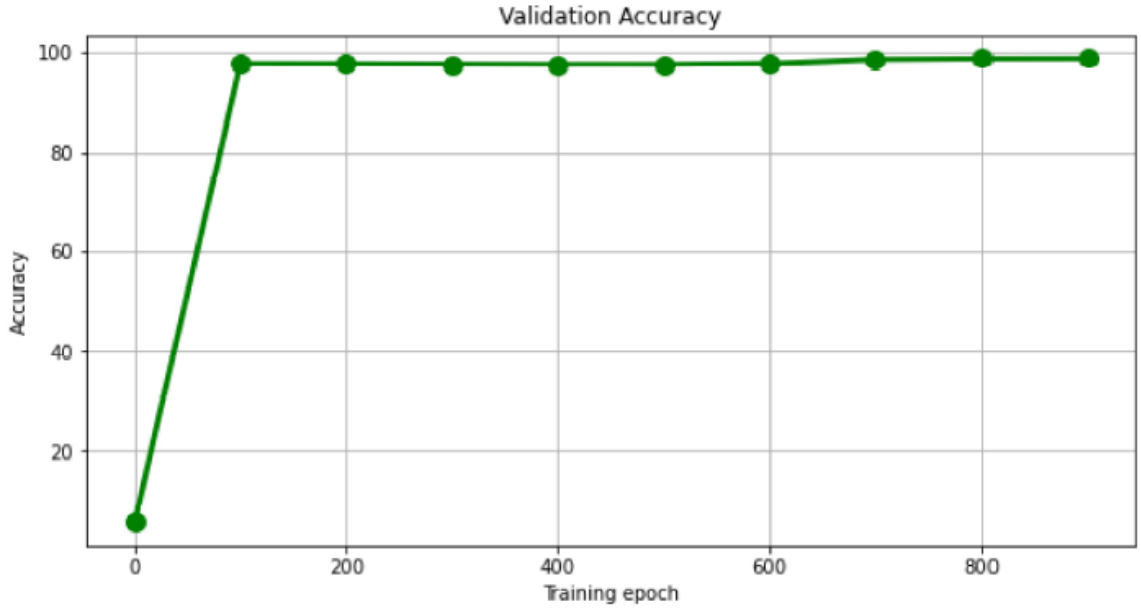


Figure 5.7: Accuracy of the Transport layer dataset with respect to the number of training cycle. Own Visualization.

Table 5.4 depicts the percentage values of the The calculated Metrics.

	Predicted As Normal	Predicted As Attack
Actually As Normal	**TN=15450**	**FP=18**
Actually as Attack	**FN=125**	**TP=5096**

Table 5.3: Numerical Representation of Confusion Matrix for Transprt Layer. Own Work.

	Evaluation Metric	Performance out of 1	Performance in %
1	Accuracy ⇑	0.9930	99.30%
2	Precision ⇑	0.9965	99.65%
3	Recall ⇑	0.9761	97.61%
4	Specificity ⇑	0.9988	99.88%
5	F1 score ⇑	0.98.62	98.62%
6	False alarm rate ⇓	0.0012	0.12 %

Table 5.4: Qualitative result of Transport Layer IDS using the evaluation metrics. Own Work

As shown in Table 5.4 The Transport layer IDS has Recall value (Detection Rate) of 97.61 % and False alarm rate(False positive rate) of 0.12 % .This shows the modeled IDS has

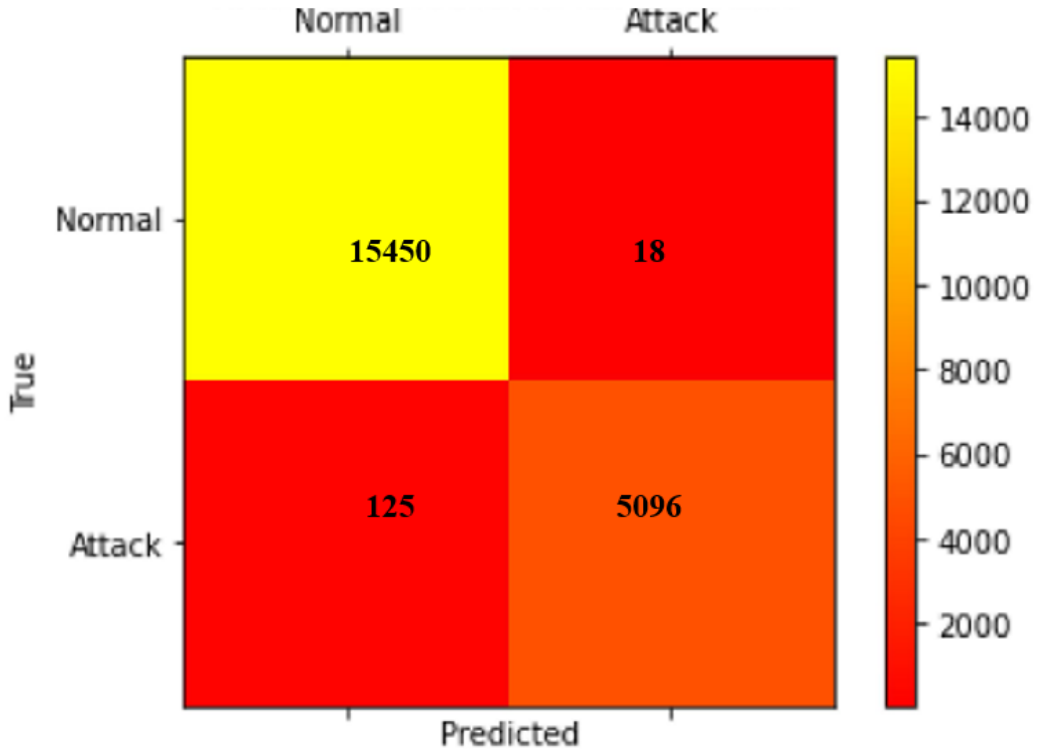


Figure 5.8: Heat map of Transport Layer Confuion Matrix. Own Visualization.

achieved a better performance while keeping the false alarm rate low. In the table the upper arrows shows the higher the values the better the result is and the down arrow shows the lower the value the better the result.

As it is shown in Figure 5.8 The Number of true negative classifications on the transport layer is higher than both the Application layer and Network layer.

5.1.3 Analysis of Network layer IDS

In this section, Network layer IDS training and evaluation results are discussed.

Figure 5.9 depicts the important of each features of the Network Layer dataset . The first 12 features starting from left side are the highly important features which were reduced from 41 to 12 .

Table 5.6 depicts the percentage values of the The calculated Metrics.

As shown in Figure 5.11 the training loss at each training cycle (epoch) is decreasing towards zero. in the other hand Figure 5.12 depicts the Validation accuracy at each training

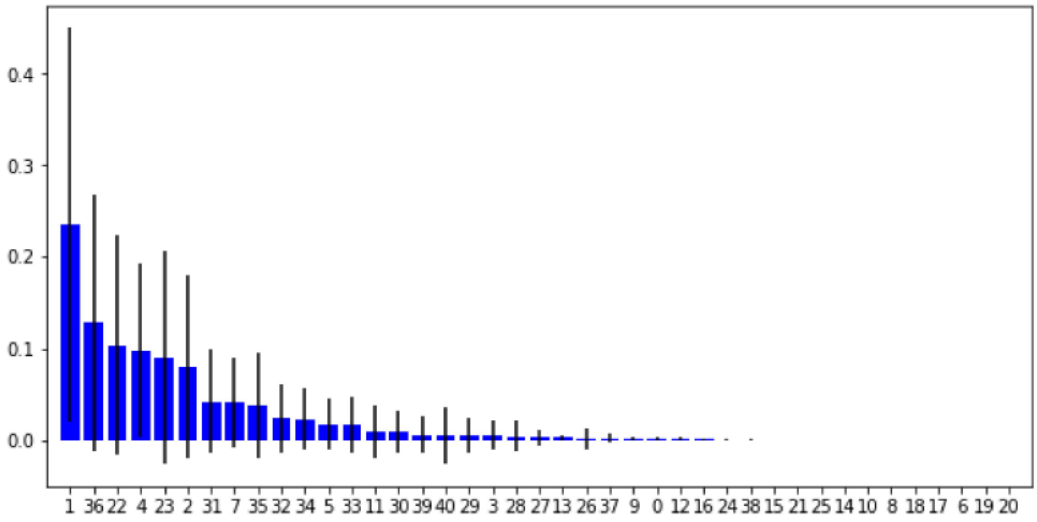


Figure 5.9: Network Layer Feature importance (Y-axis) VS feature IDS (X-axis). Own Visualization.

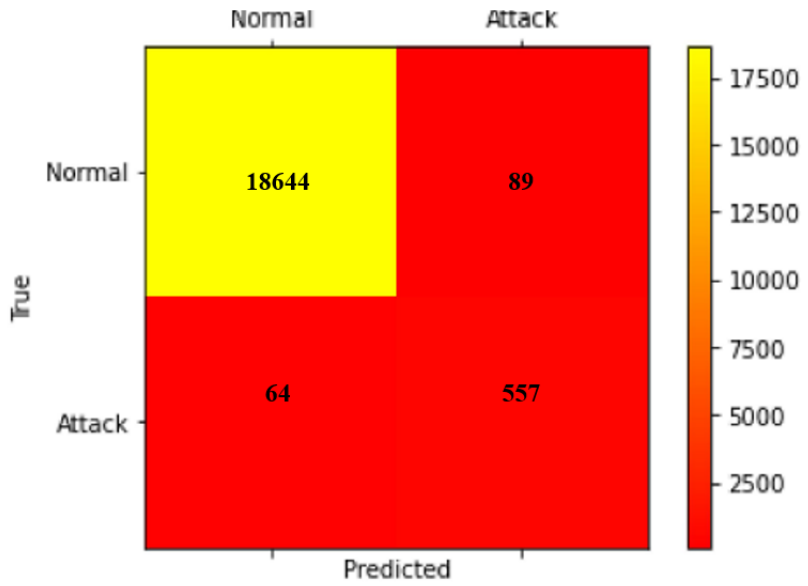


Figure 5.10: Heat Map of Confusion Matrix for Network Layer. Own visualization.

	Predicted As Normal	Predicted As Attack
Actually As Normal	**TN=18644**	**FP=89**
Actually as Attack	**FN=64**	**TP=557**

Table 5.5: Numerical Representation of Confusion Matrix For Network Layer. Own Work.

cycle (epoch) is increasing towards one. This shows that the network layer IDS training process performed well, i.e decreasing the training loss while increasing the validation accuracy without model over-fitting which results better generalization capability of the network

	Evaluation Metric	Performance out of 1	Performance in %
1	Accuracy ⇑	0.9921	99.21%
2	Precision ⇑	0.8623	86.23%
3	Recall ⇑	0.8969	89.69%
4	Specificity ⇑	0.9953	99.53%
5	F1 score ⇑	0.8793	87.93%
6	False alarm rate ⇓	0.0047	0.47 %

Table 5.6: Qualitative result of Network Layer IDS using the evaluation metrics. Own Work

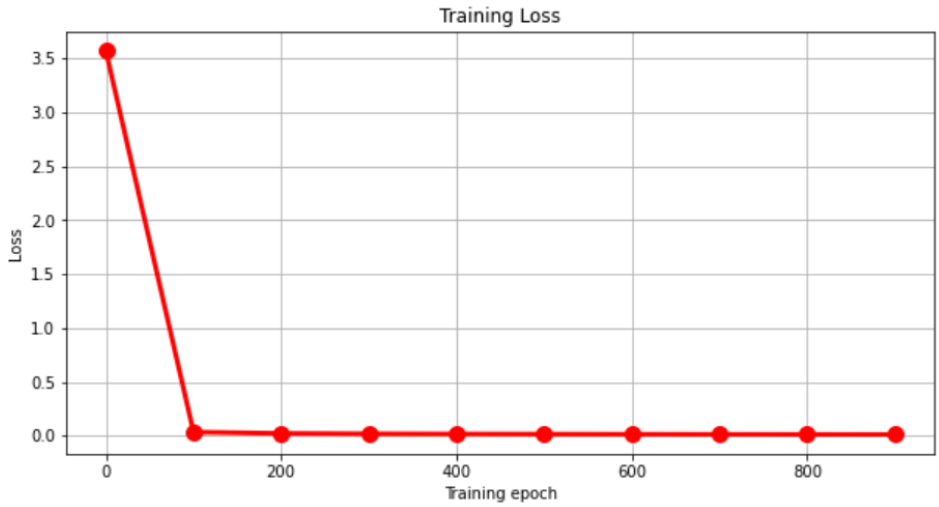


Figure 5.11: The Training Loss of Network Layer IDS at each layer (epoch). Own Visualization.

model.

As shown in Table 5.6 The Network layer IDS has Recall value (Detection Rate) of 89.69 % and False alarm rate(False positive rate) of 0.47 %.This shows the modeled IDS has achieved a better performance while keeping the false alarm rate low. In the table the upper arrows shows the higher the values the better the result is and the down arrow shows the lower the value the better the result.

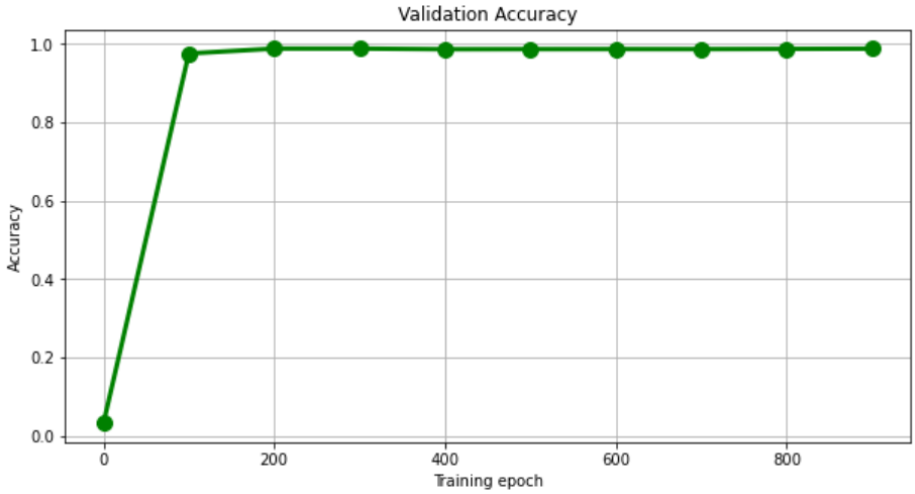


Figure 5.12: Accuracy of the Network layer dataset with respect to the number of training cycle. Own Visualization.

Chapter 6

Conclusion

Cybercriminals are affecting computer users by using advanced and sophisticated techniques. Therefore, it becomes progressively crucial for computer frameworks to be protected using advanced intrusion detection systems which are competent in recognizing advanced malware. It is fundamental to have a complete overview of current IDS research's qualities and impediments to construct such IDS systems. In this paper, a summary of intrusion detection systems, including their preferences and restrictions, has been studied and presented. Even though different machine learning techniques have been proposed to detect zero-day attacks, many of them may have the problem of generating and updating the information about new attacks, which results in less accuracy and a high false alarm rate (FAR). As a solution to this IDS issue, anomaly Intrusion detection systems have been modeled using a deep learning algorithm, and it was implemented using a TensorFlow deep learning framework. In addition, The KDD cup '99, the most popular public datasets that are utilized in the IDS research, is used during the experiment analysis of this thesis work. Based on the attack types present at the TCP/IP layers, the data set is split into different layers, i.e., application layer, transport layer, and Network layer, respectively. The result of the experiment is also explained in a graphical representation in a way that shows the detailed performance analysis of the predictive model. The application layer IDS has a detection rate of 94.35% and a False alarm rate(False positive rate) of 0.33%. The Transport layer IDS has a Detection Rate of

97.61 %. and a False alarm rate(False positive rate) of 0.12 %. The Network layer IDS has Detection Rate of 89.69 % and False alarm rate(False positive rate) of 0.47 % .

The carried out experiment results at each layer have demonstrated that the proposed model has a high detection rate along with a low False alarm rate (FAR).

6.1 Future work

Implementation of proactive IPS technology will provide a comprehensive and robust defense line to protect systems from any attack. An intrusion prevention system (IPS) is considered the next step in the evolution of intrusion detection systems (IDS). As future work For anyone who wants to pursue in this area, I recommend to develop Network Intrusion Prevention Systems based on deep learning approach that provide advanced protection beyond what is offered by firewalls and Intrusion Detection Systems.

Bibliography

[1] K. Kim, M. E. Aminanto, and H. C. Tanuwidjaja, *Network Intrusion Detection Using Deep Learning: A Feature Learning Approach*. Springer, 2018.

[2] I. A. Saroit, M. E. Elhamahmy, and H. N. Elmahdy, "A new approach for evaluating intrusion detection system," *CiiT International Journal of Artificial Intelligent Systems and Machine Learning*, vol. 2, 2010.

[3] A.-S. K. Pathan, *The state of the art in intrusion prevention and detection*. CRC press, 2014.

[4] M. Rai and H. Mandoria, "Network intrusion detection: A comparative study using state-of-the-art machine learning methods," in *2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, vol. 1, pp. 1–5, IEEE, 2019.

[5] S. Choudhary and N. Kesswani, "Analysis of kdd-cup'99, nsl-kdd and unsw-nb15 datasets using deep learning in iot," *Procedia Computer Science*, vol. 167, pp. 1561–1573, 2020. International Conference on Computational Intelligence and Data Science.

[6] P. Sangkatsanee, N. Wattanapongsakorn, and C. Charnsripinyo, "Practical real-time intrusion detection using machine learning approaches," *Comput. Commun.*, vol. 34, no. 18, pp. 2227–2235, 2011.

[7] T. Kaur, V. Malhotra, and D. Singh, "Comparison of network security tools-firewall, intrusion detection system and honeypot," *Int. J. Enhanced Res. Sci. Technol. Eng*, vol. 200204, 2014.

[8] S. Kamara, S. Fahmy, E. Schultz, F. Kerschbaum, and M. Frantzen, "Analysis of vulnerabilities in internet firewalls," *Computers & Security*, vol. 22, no. 3, pp. 214–232, 2003.

[9] Q. Chen, W. Lin, W. Dou, and S. Yu, "CBF: A packet filtering method for ddos attack defense in cloud environment," in *IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing, DASC 2011, 12-14 December 2011, Sydney, Australia*, pp. 427–434, IEEE Computer Society, 2011.

[10] S. Kamara, S. Fahmy, E. Schultz, F. Kerschbaum, and M. Frantzen, "Analysis of vulnerabilities in internet firewalls," *Computers Security*, vol. 22, no. 3, pp. 214–232, 2003.

[11] P. Aggarwal and S. K. Sharma, "Analysis of kdd dataset attributes - class wise for intrusion detection," *Procedia Computer Science*, vol. 57, pp. 842–851, 2015. 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015).

[12] O. Depren, M. Topallar, E. Anarim, and M. K. Ciliz, "An intelligent intrusion detection system (ids) for anomaly and misuse detection in computer networks," *Expert Systems with Applications*, vol. 29, no. 4, pp. 713–722, 2005.

[13] A. Meryem and B. E. Ouahidi, "Hybrid intrusion detection system using machine learning," *Network Security*, vol. 2020, no. 5, pp. 8–19, 2020.

[14] Y. Bouzida and F. Cuppens, "Neural networks vs. decision trees for intrusion detection," in *IEEE/IST workshop on monitoring, attack detection and mitigation (MonAM)*, vol. 28, p. 29, Citeseer, 2006.

[15] P. V. Amoli, T. Hamalainen, G. David, M. Zolotukhin, and M. Mirzamohammad, "Unsupervised network intrusion detection systems for zero-day fast-spreading attacks and botnets," *JDCTA (International Journal of Digital Content Technology and its Applications*, vol. 10, no. 2, pp. 1–13, 2016.

[16] H. Gharaee and H. Hosseinvand, "A new feature selection ids based on genetic algorithm and svm," in *2016 8th International Symposium on Telecommunications (IST)*, pp. 139–144, IEEE, 2016.

[17] S. Mukkamala, A. H. Sung, and A. Abraham, "Intrusion detection using an ensemble of intelligent paradigms," *Journal of network and computer applications*, vol. 28, no. 2, pp. 167–182, 2005.

[18] A. Chandrasekhar and K. Raghuveer, "Intrusion detection technique by using k-means, fuzzy neural network and svm classifiers," in *2013 International Conference on Computer Communication and Informatics*, pp. 1–7, IEEE, 2013.

[19] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *2017 International Conference on Information Networking (ICOIN)*, pp. 712–717, IEEE, 2017.

[20] P. Torres, C. Catania, S. Garcia, and C. G. Garino, "An analysis of recurrent neural networks for botnet detection behavior," in *2016 IEEE biennial congress of Argentina (ARGENCON)*, pp. 1–6, IEEE, 2016.

[21] H. Shapoorifard and P. Shamsinejad, "Intrusion detection using a novel hybrid method incorporating an improved knn," *Int. J. Comput. Appl*, vol. 173, no. 1, pp. 5–9, 2017.

[22] B. A. Tama, M. Comuzzi, and K.-H. Rhee, "Tse-ids: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system," *IEEE Access*, vol. 7, pp. 94497–94507, 2019.

[23] R. Sathya and A. Abraham, "Comparison of supervised and unsupervised learning algorithms for pattern classification," *International Journal of Advanced Research in Artificial Intelligence*, vol. 2, no. 2, pp. 34–38, 2013.

[24] D. A. Cieslak, N. V. Chawla, and A. Striegel, "Combating imbalance in network intrusion datasets.," in *GrC*, pp. 732–737, Citeseer, 2006.

[25] O. Depren, M. Topallar, E. Anarim, and M. K. Ciliz, "An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks," *Expert Syst. Appl.*, vol. 29, no. 4, pp. 713–722, 2005.

[26] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[27] E. Aminanto and K. Kim, "Deep learning in intrusion detection system: An overview," in *2016 International Research Conference on Engineering and Technology (2016 IRCET)*, Higher Education Forum, 2016.

[28] L. Bontemps, J. McDermott, N.-A. Le-Khac, *et al.*, "Collective anomaly detection based on long short-term memory recurrent neural networks," in *International Conference on Future Data and Security Engineering*, pp. 141–152, Springer, 2016.

[29] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

[30] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[31] A. Raghavan, F. D. Troia, and M. Stamp, "Hidden markov models with random restarts versus boosting for malware detection," *J. Comput. Virol. Hacking Tech.*, vol. 15, no. 2, pp. 97–107, 2019.

[32] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[33] S. Devaraju and S. Ramakrishnan, "Performance comparison for intrusion detection system using neural network with kdd dataset.," *ICTACT Journal on Soft Computing*, vol. 4, no. 3, 2014.

[34] N. El Kadhi, K. Hadjar, and N. El Zant, "A mobile agents and artificial neural networks for intrusion detection," *Journal of software*, vol. 7, no. 1, pp. 156–160, 2012.

[35] Z. Wang, "Deep learning-based intrusion detection with adversaries," *IEEE Access*, vol. 6, pp. 38367–38384, 2018.