Jimma University

Jimma Institute of Technology

Faculty of Computing and Informatics

M.Sc. in Computer Networking

Hybrid Approach to Improve Accuracy of Intrusion Detection System in IoT Environment

By

Aschalew Woldeyesus

Advisor: Mr. Kebebew Ababu (Ass.Prof.)

Co-advisor: Mr. Sofanit Alemu (M.Sc.)

January 2021 Jimma, Ethiopia

Jimma University

Jimma Institute of Technology

Faculty of Computing and Informatics

Hybrid Approach to Improve Accuracy of Intrusion Detection System in IoT Environment

By: Aschalew Woldeyesus

This is to certify that the thesis prepared by **Aschalew Woldeyesus**, entitled **Hybrid Approach to Improve Accuracy of Intrusion Detection System in IoT Environment,** Submitted in partial fulfillment of the requirements for the Degree of Master of Science in *Computer Networking* compiles with the regulations of the University and meets the accepted standards with respect to originality and quality.

Approved by board of Examining Committee:

	Name	Signature
Faculty Dean:	_____	_____
Advisor:	Mr. Kebebew Ababu (Ass.Prof.)	_____
External Examiner:	Dr. Melkamu Deressa (Phd)	_____
Internal Examiner:	_____	_____
Chair Person:	_____	_____

Jimma, Ethiopia

January,2020

DECLARATION

I, the undersigned, declare that this thesis entitled Hybrid Approach to Improve Accuracy of Intrusion Detection System in IoT Environment is my original work and has not been presented for a degree in this or any other universities, and all sources of references used for the thesis work have been appropriately acknowledged.

Name: Aschalew Woldeyesus

Signature:_____, Date:_____ This thesis has been submitted for examination with my approval as a University advisor.

Advisor Name: Mr. Kebebew Ababu

Signature: _____, Date:_____

This thesis has been submitted for examination with my approval as a University Co-advisor.

Co-Advisor Name: Mr. Sofanit Alemu

Signature: _____, Date :_____

Dedication

——I dedicate this thesis to my beloved family——

Acknowledgments

I would like to thank my advisor Mr. Kebebew Ababu and Mrs. Sofanit Alemu for the assistance for comments that greatly improved the manuscript. I thank all my colleagues from Jimma University who provided insight and expertise that greatly assisted the research. I would also like to show my gratitude to Mr. Boaz, Lecturer, Jimma University for sharing his pearls of wisdom with me during this research. I am also immensely grateful to all advisors and colleagues for their sincere support, encouragement, and meaningful guidance on doing this work.

Abstract

Internet of Things (IoT) is an interconnected system of devices that encourage continuous exchange of data between devices. Experts predict that by now, wireless network traffic size accounts for two third of total Internet traffics to be generated by 50 billion Wi-Fi and cellular connected devices [1]. These devices could be agricultural, medical, healthcare, smart city, driverless vehicles, industrial robots or wearables that can be remotely controlled and configured [1]. IoT devices are expected to become more popular than mobile devices and will have access to the most vital data [3]. This will cause in rise attack surface area and probabilities of attacks. Internet of Things (IoT) is recent technology and system that has the capacity to change our way of life.

As IoT becomes more pervasive every day, probabilities of an attack against it increase. Having such a boom in the number of these devices by 2020, marks the question of what sort of sensitive data these devices will be able to communicate. Such data can be environmental, financial, medical, or any other crucial data. Security attacks against IoT devices are increasing every year. September 14, 2019, about 2.9 billion security attacks were measured [4]. Hence, some form of intrusion detection can be used to alert security attack.

This work focuses on developing intrusion detection and prevention system for iot network.IT involves combining anomaly based and signature based modules. Signature based module is implemented using iptables firewall and anomaly based module is implemented using deep learning algorithm. Raw data collected from iot LAN containing 2 Raspberry Pi, 1 smart phone, 1 CCTV camera and one laptop. The collected data is preprocessed and trained on Google collaborator. The trained model is tested and evaluated using different performance measurement techniques and got 99.96% accuracy. Finally, the deep neural model is integrated with anomaly based module to maximize the accuracy of intrusion detection. This works focus on detection and preventing of doss attacks in IOT network.

Keywords: DNN, IDS, IOT, Anomaly based, Signature based

Contents

List of figures

List of tables

List of Acronyms

SIEM ----------Security information and event management.

IDS----------Intrusion Detection System

SIDS----------Signature intrusion detection systems

RBMs----------Restricted Boltzmann Machines

DBN ----------deep brief networks

DNN ----------deep neural networks

CNN----------convolutional neural networks

RNN ----------recurrent neural networks

LAN----------Local Area Network

IOT----------Internet of things

DOS----------—denial of service

DDOS----------distributed denial of service

NIDS----------network intrusion detection system

IPS----------intrusion prevention system

MITM----------man in the middle of attack.

VPN----------virtual private network

HIDS----------host based intrusion detection system

Gbps----------Giga bit per second

RNN---------- ----------Recurrent Neural Network

NAT----------Network Address Translation

TCP---------- transmission control protocol

UDP----------user datagram protocol

CSV---------- comma separated values

Chapter 1: INTRODUCTION

1.1 Background of the Study

Since 1960s the Internet has been playing a major role in making peoples and organization and business close together. It has broken down geographical barriers between people and has given them a robust, efficient, and cost-effective means of communication. Now the situation is about to change due to the emergence of smart objects that have the ability of generating and transmission of data over the Internet similarly to humans.

Internet of things is recent technology and systems that can change our way of life. IoT can be taken as a technology that combines two components; "Things" and "Internet." The "Things" refers to any device that collects data about the surrounding environment as well as itself. Depending on what type, smartness, and capabilities of this object, the object may be able to analyze and act smartly with other objects using the "Internet" as a network for communications [2].

Internet of things is an interconnected system of device that enhances continuous exchange of information between physical devices. Experts predict that by 2020, wireless network traffic is anticipated to account for two-thirds of total Internet traffics to be generated by 50 billion Wi-Fi and cellular-connected devices [1]. These devices could be medical, driverless vehicles, smart TVs, sensors, smart cities or healthcare devices that could be remotely controlled and configured [1]. IoT devices are expected to become more popular than mobile devices and will have access to personal and sensitive information [3]. This will result in increasing the chance of attack and surface area.

The insufficient security measures and lack of dedicated anomaly detection systems for these heterogeneous networks make them vulnerable to a range of attacks such as data leakage, spoofing, denial of service (DoS/DDoS), energy bleeding, insecure gateways, etc. These could result in destructive effect to hardware and network systems causing system blackouts, denying the availability of system or harming individuals physically [5], [6]. Therefore, it is clear These can lead to disastrous effects; causing damage to hardware, disrupting the system availability,

causing system blackouts, and even physically harms individuals [5], [6]. Therefore, it is clear that the scale of the impact of the attacks performed on IoT networks can vary significantly. For example, a relatively simple and seemingly harmless de-authentication attack can cause no significant damage, but if it is performed on a device with critical significance, such as a driverless vehicles can cause serious damage to human life. As a result, there is a major gap between security requirements and security capabilities of currently available IoT devices. Two of the major reasons that make IoT devices less secure are limitation in computational power and heterogeneity in terms of hardware, software, and protocols [7]. It is generally not resource effective for these devices with limited battery source, computational power, radio bandwidth and memory to execute computation intensive tasks and communication load [8]. So, it is difficult to employ complex and robust security techniques. The diversity of these devices is challenging to design and implement a security technique that can withstand the scale and range of devices [9]. Therefore, lightweight intrusion detection systems are being developed to tackle these anomalous behaviors.

Recently, deep learning based intrusion detection system has been implemented to monitor network traffic. But the accuracy of the detection depends on the amount of data used, and the algorithms chosen for detection and classification. As result, different researchers use different algorithms and datasets to implement IDS in IoT networks giving good accuracies. So researches conducted in intrusion detection system in IoT are continuously booming to improve the accuracy of IDS in Internet of Things.

1.2 Statement of the problem

IoT devices are becoming more available and by now about 50 billion devices are being connected [1].As IoT devices become more and more abundant every day, assault against these devices increases. Having such a massive increase in the number of devices by 2020 underlines the question of what sort of sensitive data these devices will be able to communicate. Such data could be medical, agricultural, financial, and environmental or any other types of data that has variety of significance. With this in consideration it is vital to think how to make these devices secure. Security attacks on IoT devices are increasing every day. On September 14, 2019, about

2.9 billion security attacks were recorded [4]. Hence, some way of intrusion detection system mechanisms can be implemented to alarm security attack. One of the old ways of detecting intrusion is by the use of misuse-based detection system. In misuse-based detection, a signature is defined for each type of attack that is to be detected. If the network packet that is monitored matches the signature of an attack an alarm is rung [5]. However, it is unable to detect new previously unknown attack (unable to detect zero-day attack). The better way of detecting zero-day attack is by using Anomaly-based detection system. In anomaly detection based IDS the normal behavior of the system is defined and actions that do not fit in this defined normal behavior are labeled as anomalous. The advantage of Anomaly-based IDS is that they can detect zero-day attacks, which rule-based IDS would not be able to detect because it has no signature for these attacks. But anomaly-based IDS has high rate of false positivity because of not anything outside normal behavior is an attack [11].

Intrusion detection system technique has been developed and been used for traditional network the current techniques IDSs for IoT are insufficient to detect different types of attacks for the following reasons [6]. IoT specific characteristics create a challenge for designing IDS. IoT devices are large in quantity and have to host IDS agents; in addition, small storage and computational power of IoT creates a challenge on how to implement IDS on IoT. The other important issue is the characteristic associated with the IoT network design. In the traditional networks, the computer system is completely connected to specific computer nodes that are responsible for sending packets to the endpoints. In contrast, the IoT ecosystem communicates with numerous sensors and actuators to accomplish several monitoring and control tasks. IoT devices have significantly more varieties and types of networks than traditional networks. Therefore, applying traditional IDS detection system to the IoT ecosystem is hard because of its specific features, such as limited resources, particular protocol stacks, and network requirements [6].

Therefore, this research will attempts to answers the following research questions.

- How to improve the performance of IDS in IoT networks by using hybrid machine learning algorithms?

- How to minimize the false positive rate of IDS in the IoT network?

1.3 Objectives

1.3.1 General Objective

The objective of the study is to develop an intelligent hybrid-based deep learning intrusion detection system for IoT to improve the performance of the network security and proposing new IDS architecture.

1.3.2 Specific Objective

To achieve the above general objective the following specific objectives will be followed:

- To review previous works in IoT network security based on intrusion detection systems and to understand deep learning algorithms.
- To collect an up-to-date IoT network dataset
- To prepare and preprocess the collected data
- To train dataset using deep learning algorithms
- Propose a hybrid based IDS model
- Test and evaluate the model

1.4 Methodology of the Study

Different methodologies will be used in this research to achieve the general and specific objectives of this study. The methodologies that will be used in conducting this research are described as follows:

1.4.1 Literature Review

To achieve the objectives of the study relevant literature has been used in different books, journal articles, and conference papers on the areas related to IDS in IoT environment to have a detailed understanding of the area. Revising the literature is a key component of the research method. The literature is reviewed throughout our research with consideration of newly published works where necessary.

1.4.2 Data sources and Data Collection

While collecting network traffic data it will be checked whether it is wired network data or IoT network traffic data. The data set must include both the normal as well as malicious data. Up to date data is collected using the t-shark network monitoring tool inside my LAN network.

1.4.3 Data Preparation

In the data preparation phase, the data which is usually in .csv file format opened and checked for cleanliness. For example, empty rows will be removed hence they have a negative impact in decreasing the accuracy of the model. Textual data will be converted into numbers. Unnecessary columns will be removed because the number of columns increases the duration of training time increases.

1.4.4 Training dataset

GoogleCollab Tensor Flow/Keras is used to train the dataset using the preferred algorithms. The output of the training is a knowledge base model which will be able to identify malicious network packet from entire IoT network traffic.

1.4.5 Testing and evaluating

The accuracy of the developed system is evaluated by using different techniques such as Precision, Recall, and F-measure.

1.5 Scope and limitation of the study

The main intent of this study is to improve the performance of the intrusion detection system in the IoT network environment by examining deep learning algorithms to identify the best algorithms that best suits for IoT network intrusion detection. This work also focuses on intrusion prevention system that can take an action for the incoming intrusion. Moreover, the anomaly-based system uses only a single detection algorithm even if combining many detection architectures may have an impact on increasing detection rate. This work only focuses on detecting and preventing doss attacks only and other types of attacks are not included in this paper.

1.6 Application of Results

The frequency and application of IoT devices are growing globally. Those devices may contain medical, financial, environmental or any other vital and sensitive data. Therefore; securing the data contained in IoT devices is important. IDS systems play a vital role in notifying system administrators about unauthorized and illegitimate network access. So improving the effectiveness of IDS in IoT network has positive impact on the internet of things security.

The research is applicable where persistent IoT network security monitoring is important. For example, it may be applicable to monitor network traffic of smart connected cars, smart agriculture and body area network sensors, and so on. This research will improve the effectiveness of anomaly classifiers. At the end of this work, it provides input for network admins about the presence of malicious traffic in the IoT network environment.

1.7 Organization of the thesis

The rest of the thesis is organized as follows:

Chapter Two: focuses on the related work in the field of intrusion detection using different detection techniques. It also discusses how intrusion detection systems are classified, what are the key functions of an IDS, what are key the components of this system, the deployment scenario of an IDS in different architecture and finally it discusses signature-based open-source IDS.

Chapter Three: discusses related works that have significant relation with this thesis. Even if there are several works done on this area, this Chapter selects the most related works to our thesis and presents them based on the attack detection approach.

Chapter Four: concentrates on providing a design for the proposed work. Issues raised here are a signature detection module, an anomaly detection module, and a signature generation module. It rises and answers the question "What are the requirements of a hybrid intrusion detection system?”

Chapter Five: the proposed system is implemented in this Chapter by applying the proposed algorithms. Evaluation of the work using experimental analysis and comparing to other works is presented here.

Chapter Six: summarizes the contributions made in the thesis, and concludes based on the results obtained from the thesis work. Furthermore, new issues that have been surfacing while working on the thesis are suggested as future work.

Chapter 2: Literature Review

2.1 Overview

Internet of things is an interconnected system of device that enhances continuous exchange of information between physical devices. Experts predict that by 2020, wireless network traffic is expected to account for two-thirds of total Internet traffic that will be generated by 50 billion Wi-Fi and cellular-connected devices [1].There are a variety of applications and services provided by the IoT ranging from critical infrastructure to agriculture, military, home appliances, and personal healthcare [40]. In addition, the domains covered by the IoT services include, but not limited to power, building management, medical, retail, transportation, Industries, manufacturing, and so on.

Privacy and security are the two major factors that play major role in in the commercial realization of the IoT services and applications. The Internet acts as a media for f security attacks ranging from simple hacks to high level coordinated security breaches that has impact on medical, business and any other industries. The limited resource of IoT devices and environment they operate adds a challenge for the security of devices [39]. Recently privacy and security issues are being highly researched from different point of view such as data security, communication, and privacy and other. The large scale of IoT networks presents a challenge to manage these devices, storage capacity, communication, and security and privacy.

Intrusion detection system is a program or hardware device that monitors the network and notifies an alarm when it finds malicious activity. When it finds malicious data it stores it in security information and event management (SIEM) system. A SIEM system merges results from multiple sources and uses alarm filtering tools to differentiate malicious activity from false alarms [12].

2.2 Internet of Things Security

Computer Security is the prevention of computer systems and information from damage, theft, and unauthorized use. Computer hardware is mainly protected by the same measures used to protect other important or sensitive equipment, namely, serial numbers, doors and locks, and

alarms. The protection of data and system access, on the other hand, is accomplished through other methods, some of them quite complicated [13].

IoT security is the technology area related with protecting connected devices and networks in the internet of things. IoT involves increasing internet connectivity to a system of connected computing devices, hardware machines and digital machines, people, animals, and/or objects. Each "thing" is provided a special identifier and the capability to automatically transfer data over a network. Allowing devices to connect to the internet paved the way for creating serious vulnerability if they are not properly protected [14].

IoT security has become the subject of focus after many mainly known incidents where a common IoT device was used to gain access and attack the larger network. Implementing security technique is critical to ensuring the safety of networks with IoT devices connected to them.

2.3 Security attacks

In computer and computer networks an attack is a trial to expose, modify, stop, damage, steal, or gain non legitimate access to or make unauthorized use of an asset. A security attack can be defined as any type of illegal procedure that focuses on computer information systems, infrastructures, computer networks, or personal computer devices. An attacker is a person or process that tries to access data, functions, or other restricted places of the system without authorization, potentially with malicious purpose. Depending on the situation, a security assault can be part of cyber-warfare or cyber-terrorism. A security attack can be employed by sovereign states, individuals, groups, society, or organizations, and it may emanate from an anonymous source [12].

A security attacker may steal, alter, or damage a given target by hacking into a susceptible system. [3] Security attacks can range from installing spyware on a personal computer to trying to destroy the infrastructure of entire countries. Legal experts are asking to limit the use of the term to incidents causing physical damage, recognize it from the more routine data breaches and wider hacking activities.

2.3.1 Types of Attacks

2.3.1.1 Passive Attack

Passive attacks are these, where the attacker focus to get the information. They do not want to modify the data contained inside the original message. It is very hard to detect because it does not alter the data. Releases of a message, scanning, traffic analysis, sniffing and key loggers are some methods of passive attacks. These are some types of passive attacks.

- **Interception**

  Interception is a kind of attack that is done without the need or knowledge of the users. It disobeys the rules of confidentiality in the principle of security. In simple term, we can say interception results loss of message confidentiality. It is a type of passive attack. It is additionaly categorized into two subtypes i.e. Traffic analysis and Release of message contents. It is of four types:-

  - **Release of message**

    After you send a message to your companion, you need that because it was that person can think about the message .Using certain security instruments, we are able avoid the discharge of message substance. For illustration, we can encode the message using the algorithm.
  - **Traffic analysis**

    On the off chance that numerous messages are passed through a single channel at that point the client gets confounded can provide a few data to the assailant because it considers that message comes from his party.
  - **Sniffing**

    Sniffing may be a strategy to sniff the exchanged information that was sent by the sender. It fair tries to discover out what sort of message or information is exchanged by the sender without the authorization of the sender.
  - **IP spoofing**

    IP spoofing could be a procedure of creating IP packets with a source address that has a place to somebody else. Spoofing makes a peril when has on the LAN allow

to get their assets and administrations to trusted has by checking the source IP of the parcels. Utilizing spoofing, an interloper can fake the source address of his packets and make them see like they started on the trusted hosts.

- **Key loggers**

  It could be a program that runs within the foundation, recording all the keystrokes. Once keystrokes are logged, they are covered up within the machine for later recovery, or transported crude to the assailant. The aggressor at that point examines them carefully within the trusts of either finding passwords, or conceivably other valuable data that may be utilized to compromise the framework or be utilized in a social building assault. For case, a key logger will uncover the substance of all mail composed by the client. Key logger is commonly included within the rootkits.

2.3.1.2 Active Attack

Active assaults are assaults that make a few alterations within the unique message or creation of a few untrue messages. These assaults are exceptionally complex and cannot avoid effectively. It can assist categorize into 3 sorts: Intrusion, Manufacture, and Adjustment. Beneath these categorize Denial of service (DoS), DDoS, DRDoS, SQL Infusion, Replay assault, Disguising, Man in Middle Attacks are a few common assaults.

- **Interruption**

Interruption attacks are dynamic attacks. In this assault, an authorized substance imagines to be another substance. For illustration, there are three clients A, B & C. Client A can be postured as client C and send a message to client B. Client B accept that message came from client C. Interruption puts the accessibility of the asset in threat. It is classified into four sorts

- **Denial of Service (DoS)**

  When a framework getting the demands gets to be active attempting to set up a return communication way with the initiator (which may or may not be employing

a valid IP address) and stays in a wait condition due to which legitimate users are denied access.

- **Distributed Denial of Services (DDoS)**

  On the web, a distributed denial-of-service (DDoS) assault is one in which a huge number of compromised frameworks assault a single target, in this manner causing a refusal of benefit for clients of the focused on framework. The surge of approaching messages to the target framework basically powers it to closed down, subsequently denying service to the system to legitimate clients.

- **SQL Injection Attack**

  SQL injection may be a security defenselessness that happens within the database layers of an application. It is the act of passing SQL code into intuitively web applications that employ in database services.

- **Fabrication**

In this attack, client use some accessing service, which they are not eligible for.

- **Replay Attack**

  A replay attack may be a frame of dynamic assault in which substantial information transmission is perniciously rehashed or postponed. An assailant captures the authorized information and resend them to his individual utilize. For case, Client A needs to exchange a few sum to Client C's Bank account. Both Client a & c have an account with Bank B. Client A sends an electronic message to Bank B, asking a support exchange. Client C may capture this message, and send a moment duplicate to Bank B but Bank B could not have the thought that usually an unauthorized message. Hence Client C would get the advantage of support exchange twice. A replay assault can be avoided utilizing solid computerized marks that incorporate time stamps and the incorporation of special data from the past exchange such as the esteem of a continually increased sequence number.

- **Masquerading**

The masquerading attack may be a sort of an assault in which one framework accepts the personality of another. It's a method utilized by the attacker to imagine himself as an authorized individual to get private data illegally.

- **Modification**

  Modification causes misfortunes of judgment guideline. For case, an individual did a web exchange of 100$. But the attacker hacks this and modify it to 1000$. This can be a case of astuteness. Beneath this assault procedure is the man in the middle of attack.

  - **Man of the middle attack**

    It is abbreviated as MITM. It is a dynamic web assault that endeavors to caught, studied, and change the data drifting between the clients of an open arrange and any asked site. The attacker uses the illegally picked up data for personality theft and other sorts of fraud [15].

2.4 Security Countermeasures

Security countermeasure points to avoid assaults and frauds [6]. By conveying preventive countermeasures it points to avoid security dangers by dispensing with as numerous vulnerabilities as conceivable [2]. In anticipating malevolent assaults and unauthorized utilization of computers numerous advances and strategies have existed and numerous Organizations are endeavoring to preserve privacy, judgment, and accessibility of their organized assets and numerous strategies have been utilized to protect against organize interruption. In any case, indeed in spite of the fact that these measures give a level of security, they are lacking in numerous ways. The foremost common countermeasures are listed underneath:

I. Use of Intrusion Prevention system

An intrusion prevention system (IPS) could be a shape of organize security that works to identify and anticipate recognized dangers. Intrusion prevention systems persistently monitor your network, trying to find conceivable malicious occurrences, and capturing data almost them. The IPS reports these occasions to system administrators and takes preventative activity, such as closing get to focuses and designing firewalls to anticipate future assaults. IPS arrangements can

moreover be utilized to distinguish issues with corporate security policies, deterring employees and network guests from violating the rules these policies contain [16].

II. Use of Firewall

A firewall is a network security device that monitors incoming and active network activity and chooses whether to permit or block particular activity based on a defined set of security rules. Firewalls have been the primary line of defense in an organization security for over 25 years. They set up an obstruction between secured and controlled inner systems that can be trusted and untrusted exterior systems, such as the Web. A firewall can be hardware, software, or both [17].

III. Cryptographic techniques

Cryptography stows away data from unauthorized clients; in any case, this strategy makes it difficult to know whether an assault has taken put. By and large, key management isn't a straightforward task. Cryptosystems may require uncommon key management systems such as the use of a Terminal Access Controller Access System (TACACS) or Remote Authentication Dial-In Client Benefit (RADIUS) server. This might mean specialized equipment or configuration. Otherwise, hackers might pick up get to these keys and break into the system.

IV. Authentication

A procedure utilized to confirm clients of a network asset. The viability of this is often debilitated by the reality that numerous still simple passwords whereas a few clients are either deceitful or are fair careless with their passwords such that numerous times can effortlessly be got by unauthorized users.

V. Virtual private network

A virtual private network, or VPN, is an encrypted association over the Web from a device to a network. The scrambled connection makes a difference guarantee that delicate information is securely transmitted. It avoids unauthorized individuals from listening in on the activity and

permits the client to conduct work remotely. VPN innovation is broadly utilized in corporate situations [17]. A VPN expands a corporate arrange through scrambled associations made over the Web. Since the activity is scrambled between the device and the network, activity remains private because it voyages. A worker can work outside the office and still safely connect to the corporate organize. Indeed smartphones and tablets can connect through a VPN.

VI. Physical security

The primary level of security in any computer network is physical security. Physical security is vital for workstations but imperative for servers. Any programmer worth his or her salt can quickly defeat all but the foremost jumpy security measures on the off chance that he or she can pick up physical get to a server. To ensure the server, take after these guidelines:

- Lock the computer room.
- Give the keys as it were to individuals you trust.
- Keep track of who has the keys.
- Mount the servers on cases or racks that have locks.
- Disable the floppy drive on the server. (A common hacking method is to boot the server from a floppy, in this way bypassing the carefully made security highlights of the network working system.)
- Keep a prepared watch pooch within the computer room and nourish it as it was sufficient to keep it hungry and frantic.

2.5 Intrusion Detection Systems

Intrusion detection is the method of checking the occasions happening in a computer framework or arrange and analyzing them for signs of conceivable episodes, which are infringement or inescapable dangers of infringement of computer security approaches, satisfactory utilize arrangements, or standard security practices .IDS is either a computer program or hardware that mechanizes intrusion detection, monitors network activity for suspicious activities, and sends notices to an administrator. An IDS can be a piece of installed software or a physical appliance

that screens network activity in order to identify undesirable action and events such as illicit and pernicious activity, activity that abuses security policy, and traffic that abuses acceptable utilize policies[25]. Karen [25] moreover said that, an intrusion detection system (IDS) is program that mechanizes the intrusion detection process.

These days an IDS (Intrusion Detection System) could be a vital portion of the network security because it gives full assurance of the network, IDS distinguishes both fruitful and unsuccessful attempts of interruption. The reason of the IDS is to report all the anomalous behavior of the framework and adversely distinguish all the non-attacks. A profound ponder on the behaviors and the signature of the intrusion detection, IDS can allow a real time reaction of all such interruption occasions.

2.5.1 Signature Based Intrusion Detection System

Signature intrusion detection systems (SIDS) are based on design coordinating methods to discover a known assault; these are moreover known as Knowledge-based Discovery or Misuse Detection [20] .In SIDS, coordinating strategies are utilized to discover a past intrusion. In other words, when an intrusion signature matches with the signature of a past intrusion that as of now exists within the signature database, an alert flag is triggered. For SIDS, host's logs are reviewed to discover arrangements of commands or activities which have already been recognized as malware.

The most thought is to construct a database of intrusion signatures and to compare the current set of activities against the existing marks and raise an alarm in the event that a match is found. For example, a run the show within the shape of "if: predecessor -at that point: consequent" may lead to "if (source IP address=destination IP address) at that point name as an assault" [20].

SIDS as a rule gives an amazing detection exactness for already known intrusions [21]. Be that as it may, SIDS has trouble in identifying zero day assaults for the reason that no matching signature exists within the database until the signature of the new assault is extricated and put away. SIDS is utilized in various common apparatuses, for occurrence, Grunt and NetSTAT.In my case I will execute signature based module utilizing iptables.

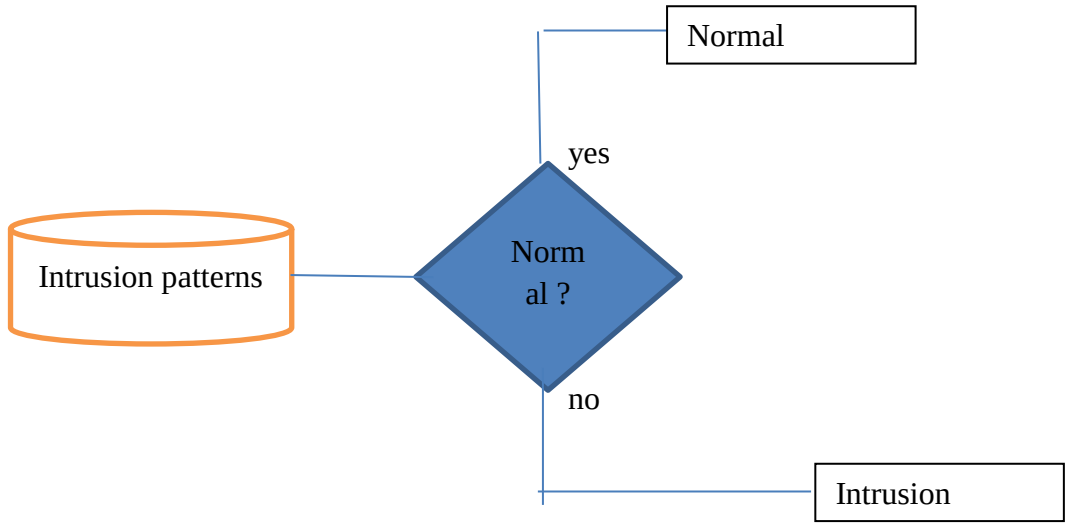

Figure 1signature based IDS [20]

IPtables

A firewall could be a set of rules. When an information bundle moves in or out of a secured network space, its contents (in specific, data around its root, target, and the convention it plans to utilize) are tried against the firewall rules to see if it should be permitted through. Iptables could be a command-line firewall utility that uses approach chains to permit or block activity. When a connection tries to set up itself on your framework, Iptables looks for a run the show in its list to match it to. In the event that it doesn't discover one, it resorts to the default activity. The taking after figure shows how iptables work.

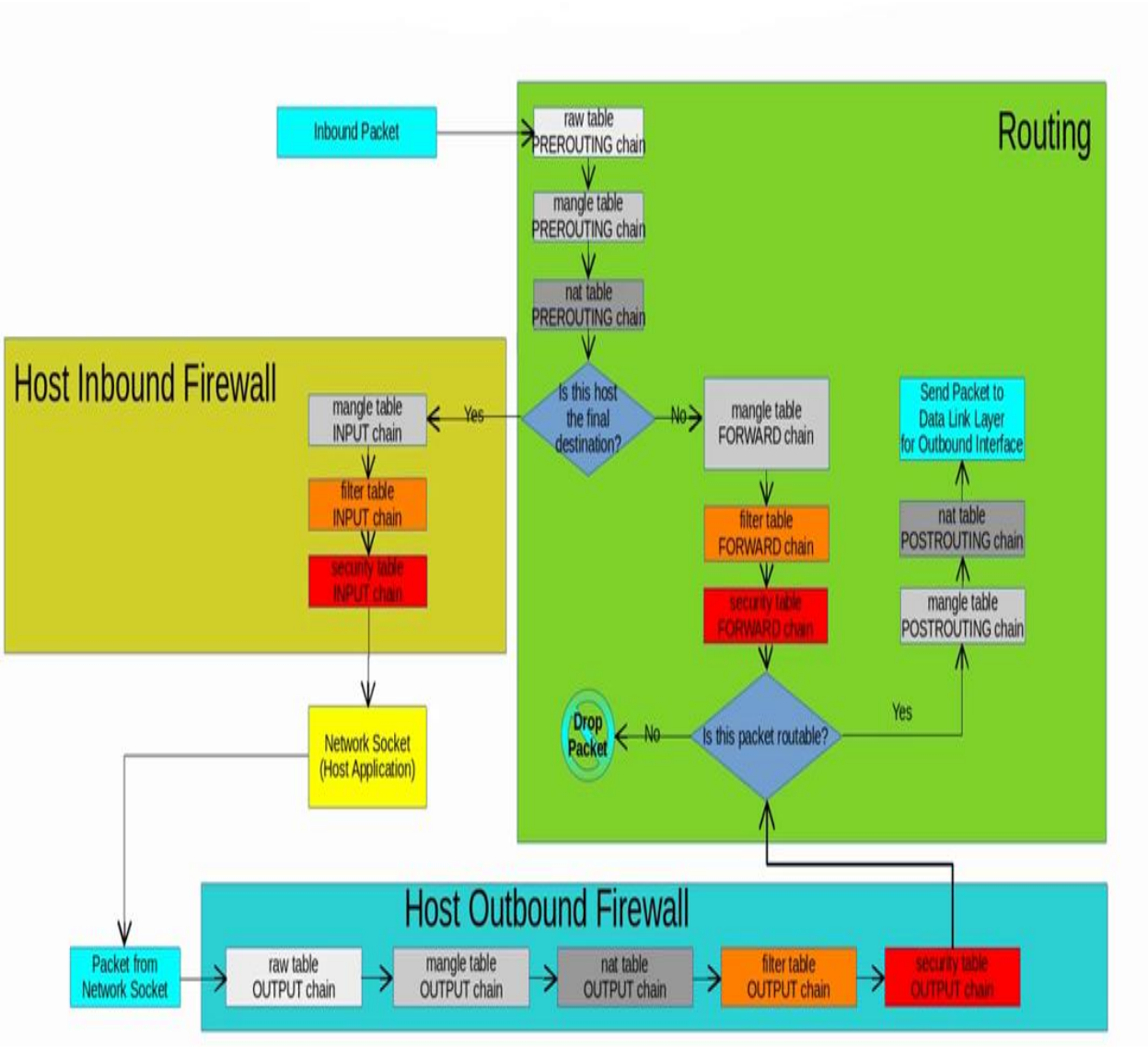

Figure 2 iptables [34]

Component of Iptables

I. Tables

A table is an IPtables construct that depicts wide categories of usefulness, such as packet filtering or Network Address Translation (NAT). There are four tables: filter, Nat, mutilate, and raw. Filtering rules are applied to the filter table, NAT rules are connected to the Nat table, and specialized rules that alter packet information are connected to the mangle table.

II. Chains

Each table has its own set of built-in chains, but user-defined chains can also be made so that the client can construct a set of rules that's related by a common tag such as INPUT_ESTABLISHED or DMZ_NETWORK. The foremost vital built-in chains for our purposes are the INPUT, Output, and FORWARD chains within the filter table:

- • The INPUT chain is navigated by bundles that are predetermined for the nearby Linux framework after a steering calculation is made inside the bit (i.e., parcels ordained for a neighborhood socket).
- The Output chain is saved for parcels that are produced by the Linux framework itself.
- The FORWARD chain administers parcels that are directed through the Linux framework (i.e., when the Iptables firewall is utilized to put through one arrange to another and bundles between the two systems must stream through the firewall).Two extra chains that are imperative for any genuine iptables deployment are the PREROUTING and POSTROUTING chains within the nat table, which are utilized to alter bundle headers some time recently and after an IP steering calculation is made inside the kernel.

III. Matches

Each Iptables rule contains a set of matches beside a target that tells Iptables what to do with a parcel that adjusts to the run the show. An Iptables coordinate may be a condition that must be met by a bundle in arrange for Iptables to process the packet according to the activity indicated by the rule target. For case, to apply a rule as it were to TCP packets, you'll utilize the --protocol match.

Each match is specified on the Iptables command line. The most important Iptables are listed below.

--source	**(-s)**	Match on a source IP address or network
--destination	**(-d)**	Match on a destination IP address or network
--protocol	**(-p)**	Match on an IP value
--in-interface	**(-i)**	Input interface (e.g., eth0)
--out-interface	**(-o)**	Output interface
--state		Match on a set of connection states
--string		Match on a sequence of application layer data bytes
--comment		Associate up to 256 bytes of comment data with a rule within kernel memory

TCP Protocol Match options

These match options are available for the TCP protocol (`-p tcp`).

- dport — Sets the destination port for the bundle. Utilize either a organize benefit title (such as www or smtp), port number, or run of port numbers to design this option.
- --sport — Sets the source port of the packet utilizing the same alternatives as --dport. The --source-port coordinate choice is synonymous with --sport.
- --syn — Applies to all TCP packets planned to start communication, commonly called SYN bundles. Any packets that carry an information payload are not touched. Placing an exclamation point character (!) as a hail after the --syn alternative causes all non-SYN bundles to be matched.
- --tcp-flags — Permits TCP packets with particular set bits, or banners, to coordinate a run the show. The --tcp-flags match alternative acknowledges two parameters. The primary parameter is the mask, which sets the flags to be inspected within the packet. The second parameter refers to the flag that must be set to match.

The possible flags are:

- ACK
- FIN
- PSH

- RST
- SYN
- URG
- ALL
- NON

• For case, an iptables rule which contains -p tcp --tcp-flags ACK,FIN,SYN SYN as it were matches TCP packets that have the SYN flag set and the ACK and Fin flags unset. Utilizing the exclamation point character (!) after --tcp-flags switches the impact of the match choice.

UDP Protocol match options

These match choices are accessible for the UDP protocol (-p udp):

- --dport — Indicates the destination port of the UDP packet, utilizing the service name, port number, or extend of port numbers. The --destination-port match alternative is synonymous with --dport.
- --sport — Indicates the source port of the UDP packet, utilizing the service name, port number, or run of port numbers. The --source-port coordinate alternative is synonymous with --sport.

ICMP Protocol

The following match options are available for the Internet Control Message Protocol (ICMP) (-p icmp):

•--icmp-type — Sets the name or number of the ICMP type to coordinate with the rule. A list of substantial ICMP names can be recovered by writing the iptables -p icmp -h command.

Additional Match Option Modules

- State module — Empowers state matching. The state module empowers the taking after alternatives:--state — match a packet with the following connection states:
- ESTABLISHED — the matching packet is related with other packets in an set up connection.
- INVALID — the matching packet cannot be tied to a known association.
- NEW -the matching packet is either making a new association or is portion of a two-way association not already seen.
- RELATED — the matching packet is starting a new connection related in some way to an existing connection.

These connection states can be used in combination with one another by separating them with commas, such as -m state --state INVALID,NEW.

- mac module — Empowers equipment MAC address matching. The mac module empowers the following alternative:
- --mac-source — matches a MAC address of the network interface card that sent the packet. To avoid a MAC address from a rule, put an exclamation point character (!) after the --mac-source match alternative.

Targets

At last, iptables supports a set of targets that trigger an activity when a packet matches a rule. The foremost vital targets utilized in this paper are as follows:

- ACCEPT:-Allows a parcel to proceed on its way.

• DROP:-Drops a packet. No encourage processing is performed, and as distant as the accepting stack is concerned, it is as in spite of the fact that the packet was never sent.

LOG: - Logs a packet to syslog.

- REJECT D: - drops a packet and at the same time sends an appropriate response packet (e.g., a TCP Reset packet for a TCP connection or an ICMP Port Inaccessible message for a UDP bundle).
- RETURN Continues processing a packet within the calling chain.

Iptables policy

Iptables for input chain

The INPUT chain is the iptables construct that oversees whether packets that are ordained for the nearby framework (that's, after the result of a steering calculation made by the part assigns that the packet is predetermined for a nearby IP address) may talk to a local socket. On the off chance that the primary rule within the INPUT chain instructs iptables to drop all packets (or on the off chance that the approach setting of the INPUT chain is set to DROP), at that point all endeavors to communicate straightforwardly with the system over any IP communications (such as TCP, UDP, or ICMP) will fail.

The iptables command is followed by the following key words:-

- `-A` — Adds the iptables rule to the end of the required chain. Usually the command utilized to include a rule when rule arrange within the chain does not matter.
- `-C` — Checks a specific rule before including it to the user-specified chain. This command can assist you build complicated iptables rules by provoking you for extra parameters and alternatives.
- `-D` — erases a rule in a specific chain by number (such as 5 for the fifth run the show in a chain). You'll moreover sort the complete rule, and iptables erases the rule within the chain that matches it.

- `-E` — renames a user-defined chain. This does not influence the structure of the table.
- -F — flushes the chosen chain, which viably erases each run the show within the chain. On the off chance that no chain is indicated, this command flushes each run the show from each chain.
- -h — gives a list of command structures, as well as a fast rundown of command parameters and options.
- -I — embed a rule in a chain at a point indicated by a user-defined numbers value. In the event that no number is indicated, iptables places the command at the beat of the chain.
- `-L` — records all of the rules within the chain indicated after the command.

The following command rejects all incoming packet with invalid state.

- *$ IPTABLES -A INPUT -m state --state INVALID -j DROP.*

The following command accepts all incoming connection whose state is valid:-

- *$IPTABLES -A INPUT -m state --state ESTABLISHED, RELATED -j ACCEPT*

The following command prevent from spoofing attack

- *$ iptables -A INPUT -i eth0 -s 192.168.0.0/16 -j REJECT/ DROP*

Iptables for output chain

The OUTPUT chain allows Iptables to apply kernel-level controls to network packets generated by the local system. For example, if an SSH session is initiated to an external system by a local user, the OUTPUT chain could be used to either permit or deny the outbound SYN packet.

- The following command is for allowing output connection with new state:
  - *$IPTABLES -A OUTPUT -p tcp --dport 21 --syn -m state --state NEW -j ACCEPT*
- D The following command block out going connection with invalid state
  - *$IPTABLES -A OUTPUT -m state --state INVALID -j DROP*

Iptables for FORWARD Chain

The iptables rules that pertain to packets that do not have a source or destination address associated with the firewall, but which nevertheless attempt to route through the firewall system.

The iptables FORWARD chain within the channel table gives the capacity to wrap access controls around packets that are sent over the firewall interfaces.

- The following command rejects all sent packet with invalid state.
    - *$IPTABLES -A FORWARD -m state --state INVALID -j DROP*.
- The following command accepts all sent connection whose state is valid:-
    - *$IPTABLES -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT*
- The following command block sent spoofed packets
    - *$IPTABLES -A FORWARD -i eth1 -s ! $INT_NET -j DROP*

2.5.2 Anomaly Based Intrusion Detection System

AIDS has drawn interest from a lot of researchers due to its capacity to overcome the restriction of SIDS. In AIDS, an ordinary model of the behavior of a computer framework is made utilizing machine learning, statistical-based or knowledge-based strategies. Any critical deviation between the watched behavior and the show is respected as a peculiarity, which can be translated as an interruption [21]. The presumption for this bunch of procedures is that pernicious behavior varies from ordinary client behavior. The behaviors of anomalous clients which are different to standard behaviors are classified as interruptions. Advancement of Helps comprises two stages: the preparing stage and the testing stage. Within the preparing stage, the ordinary traffic profile is utilized to memorize a model of typical behavior, and after that within the testing stage, a modern information set is utilized to set up the system's capacity to generalize to already inconspicuous interruptions.

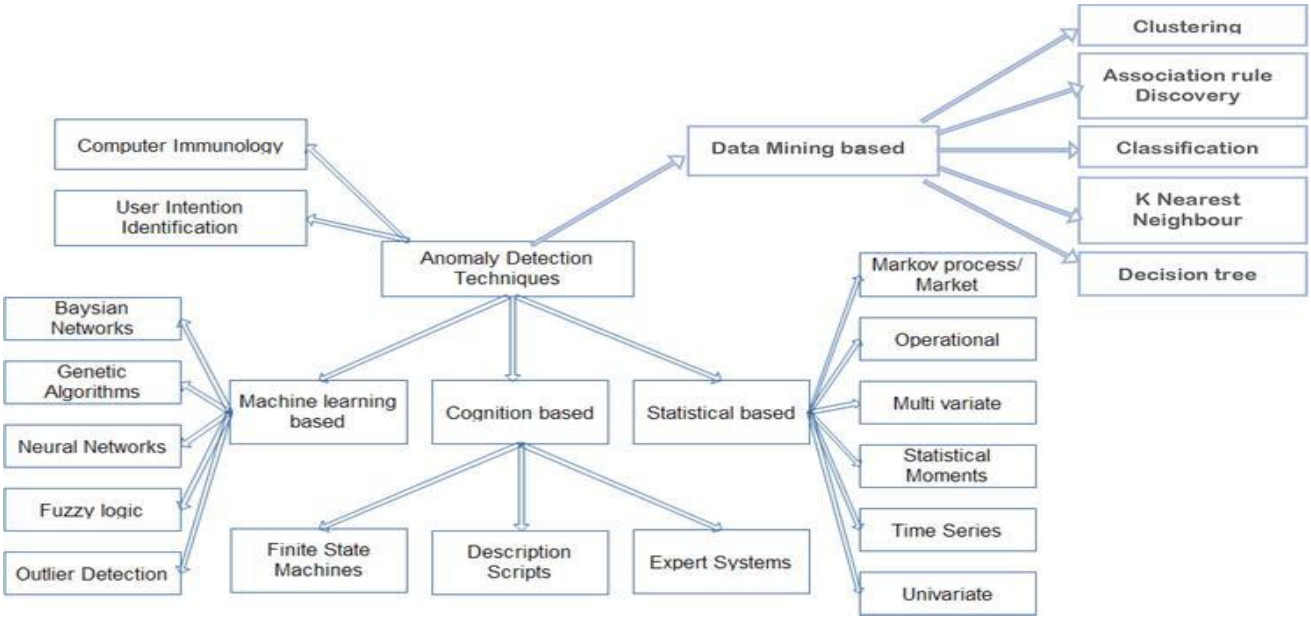

Figure 3anomaly detection techniques [20]

The most advantage of AIDS is the capacity to recognize zero-day assaults due to the reality that recognizing the irregular client movement does not depend on a signature database concurring to Alazab as cited in [20]. AIDS triggers a peril flag when the inspected behavior contrasts from the regular behavior. Moreover, Helps has different benefits. To begin with, they have the capability to find inside noxious exercises. On the off chance that an attacker begins making exchanges in a stolen account that are unidentified within the ordinary client action, it makes a caution. Moment, it is exceptionally troublesome for behavior without creating a caution as the framework is built from customized profiles.

According to Garcia some of AIDS used nowadays are the following: AirDefense Guard, Barbedwire IDS Softblade, FireProof, Prelude IDS, SecureNet IDS/IPS, Siren, Snort IDS, Snort_inline, SPADE and StealthWatch [20].

Air Defense IDS is an 802.11 remote LAN Intrusion Detection and security solution that recognizes security dangers and assaults, conducts real-time network reviews and screens the wellbeing of remote networks. An enterprise-class arrangement based on a secure apparatus and disseminated savvy sensors, Air Defense solutions scale to back single workplaces, corporate

campuses or hundreds of locations. As a key component of wireless LAN security and operational support, AirDefense is merchant freethinker and complements remote VPNs, encryption and authentication. From Times 1000 companies to government organizations, associations depend upon AirDefense solutions to secure and oversee more than 3,500 remote LANs worldwide [23].

Snort is the preeminent Open Source Intrusion Prevention System (IPS) within the world. Snort IPS employs a series of rules that offer assistance characterize pernicious network action and uses those rules to discover packets that match against them and creates cautions for users [25].Snort can be sent inline to halt these packets, as well. Snort has three essential uses: As a packet sniffer like tcpdump, as a packet lumberjack which is valuable for network activity monitoring or it can be utilized as a full-blown network intrusion prevention framework. Snort can be downloaded and configured for individual and business use alike.

Snort inline is fundamentally an adjusted adaptation of Snort that acknowledges packets from iptables and IPFW through libipq(linux) or occupy sockets(FreeBSD), rather than libpcap. It at that point employments modern run the show sorts (drop, sdrop, dismiss) to tell iptables/IPFW whether the bundle ought to be dropped, rejected, altered, or permitted to pass based on a grunt rule set. Think of this as an Interruption Prevention System (IPS) that employments existing Interruption Detection System (IDS) signatures to create choices on packets that navigate snort inline [25].

In spite of the promising nature of anomaly-based IDS, as well as its generally long presence, there still exist a few open issues with respect to these frameworks. A few of the foremost noteworthy challenges within the range are [24]:

- Low detection efficiency, particularly due to the high false positive rate more often than not gotten. This perspective is for the most part clarified as emerging from the need of great ponders on the nature of the intrusion occasions. The issue calls for the investigation and advancement of unused, exact preparing plans, as well as superior organized approaches to modeling organize frameworks.

- Low throughput and high cost, basically due to the high information rates (Gbps) that characterize current wideband transmission advances concurring to Kruegel as cited in[24] .A few proposition planning to optimize intrusion detection are concerned with framework procedures and distributed detection standards
- The absence of fitting measurements and evaluation techniques, as well as a common system for assessing and comparing elective IDS techniques.
- Another significant issue is the examination of ciphered information (e.g. in remote and portable situations), in spite of the fact that this is often too a common issue confronted by all intrusion discovery stages. In addition, this issue might be managed with by essentially finding the location specialists at those utilitarian focuses within the framework where information are accessible in ‘‘plaintext’’ arrange and, for which the comparing location examination can be carried out without extraordinary restrictions

2.5.3 Hybrid Intrusion Detection System

Both of the above mention intrusion detection methods are perfect in detecting network attacks; each of them has their own defects. To overcome their defects and to enhance the accuracy intrusion detection, combining both methods are proposed by many researchers.

2.5.4 Key Functions of IDS

The following are key functionalities given by Intrusion Detection System.

- Keep an eye on the system and user activities.
- Verification of the system errors.
- Evaluating the integrity of systems and data files.
- Note down any abnormal behavior make statically records.
- Recognition activity model mapping known attacks and alerts.
- Monitor and analyze both user and system activities
- Analyze system configurations and vulnerabilities
- Access system and file integrity
- Ability to recognize patterns typical of attacks
- Analysis of abnormal activity pattern

- Track user policy violation

2.5.5 Components of Intrusion Detection/Prevention System

According to IATR as cited in [18] , Intrusion Detection Systems are generally made up of the following main types of components:-

2.5.5.1 Sensors

Sensors— these are sent in a network or on a gadget to gather information. They take input from different sources, counting network packets, log records, and system call traces. Input is collected, organized, and after that sent to one or more analyzers.

2.5.5.2 Analyzers

Analyzers— Analyzers in IDS collect information sent by sensors and after that decide on the off chance that an interruption has really happened. Yield from the analyzers ought to incorporate prove supporting the interruption report. The analyzers may moreover give suggestions and direction on relief steps.

2.5.5.3 Honeypot

Honeypot— in completely conveyed IDS, a few directors may select to introduce a "honeypot," basically a framework component set up as snare or bait for gatecrashers. Honeypots can be utilized as early caution frameworks of an assault, distractions from basic frameworks, and information collection sources for assault examinations. Numerous IDS merchants keep up honeypots for inquire about purposes, and to create unused interruption marks. Note that a honeypot ought to as it were be conveyed when the organization has the assets to preserve it. A honeypot cleared out unmanaged may ended up a noteworthy obligation since aggressors may utilize a compromised honeypot to assault other frameworks. Review information processor, information base, choice motor, caution era and reactions.

2.5.5.4 User Interface/Event Generator

User interface— the user interface of the IDS gives the end user a view and way to interact with the framework. Through the interface the client can control and arrange the framework. Numerous clients interfacing can produce reports as well.

2.5.6 Classification of Intrusion Detection/Prevention System.

The classification of intrusion detection system is based on following factors: Location, Functionality, Deployment Approach and Detection Mechanism

2.5.6.1 Based on location

I. Host Based Network Intrusion Detection

HIDS is single computer particular intrusion detection framework which screens the security of that framework or computer from inner and outside assaults. The inner assaults allude to the circumstance where it recognizes which program gets to which asset and is there any security break. For illustration word-processor all of a sudden begins getting to framework watchword database and begins altering it. In moment portion that's outside assaults HIDS examinations parcels to and from that framework (computer) on its interfacing. HIDS reacts by logging the action and educating approximately it to assigned specialist. In HIDS against risk applications such as antivirus, spyware are introduced on a framework which screens security.

Pros:

- HIDS can protect off the LAN.
- HIDS is versatile.
- Requires lesser training than NIDS.
- HIDS does not require large bandwidth.
- HIDS provides local machine registry scan.

Cons:

- Passive system that have to wait for an event to be an indication of an attack and cannot proactively prevent it
- Data collection occurs on a per-host basis
- Writing to log or reporting activity will generate extra load for network
- Clever hackers can attack and disable HIDS while attacking HIDS does consume processing time, storage, memory and other system resources.

II. Network intrusion detection system:

Network intrusion detection system (NIDS) screens network activity and analyzes the passing activity for assaults. On recognizable proof of an assault or when an anomalous behavior is detected an alarm can be sent to the administrator. NIDS can distinguish 4 major sorts of assaults: denial of services, Test, user to root and remote to user. Cases of NIDS execution are Snort ISS, Cisco Secure IDS and Dragon Enterasys.

Pros:

- Adaptable to cross platform environment.
- NIDS is centrally managed.

Cons:

- Requires more training.
- Uses up LAN bandwidth.
- Failure rate is higher.

2.5.6.2 Based on Detection Mechanism

I. Signature based verses Anomaly based intrusion detection system

Signature based

In signature based detection mechanism the attack designs are spared within the database. Each parcel of the organize activity is compared with the assault designs to distinguish unusual behavior. Signature based interruption location framework recognizes as it were know attacks. Examples Suricata may be a signature based interruption location framework.

Pros:

- If attack signatures are clearly defined then it has low false positive.
- Easy to use.

Cons:

- Requires particular knowledge of intrusion behavior and collect information some time recently the interruption might be out of date.
- Difficult to distinguish obscure assaults.
- Raises cautions notwithstanding of the result. Illustration on the off chance that a windows worm tries to assault a Linux framework at that point the IDS sends numerous cautions of unsuccessful assault.
- The information of the assaults is subordinate on the particular environment.

Anomaly based intrusion detection system

Anomaly based intrusion detection framework is based on the network behavior. The organize behavior is characterized by the administrator or is learned by the dataset during the preparing stage of the advancement of IDS. Rules are characterized for ordinary behavior and anomalous behavior. Case, Grunt and Bro-IDS are irregularity based interruption location framework.

Pros:

- It has the ability to detect unknown attacks.

Cons:

- Defining the rule set for intrusion detection is troublesome.
- Efficiency of framework depends on the wellness of the rules and its testing on the testing datasets.

II. Rule Engine verses Artificial Intelligence

Rule Engine

Rule based intrusion detection framework recognizes atypical behavior by comparing the highlights of the parcels to a few predefined rules which are characterized by the director or which are made by a few calculations through learning. Run the show based frameworks utilize exceedingly distributable predefined marks to identify known assaults.

Pros

- It has a very high detection rate for known attacks.

Cons

- The choice of features to distinguish each assault is troublesome.
- For the intrusion detection framework to have high detection rate the rules must be carefully specified.

Artificial intelligence/ neural network/ Genetic Algorithm based intrusion detection

Artificial intelligence based intrusion detection framework is distinctive from all the calculations is that manufactured insights is utilized to characterize unused set of rules for assault location. Neural systems are the foremost common sort of counterfeit insights sort for intrusion detection. Neural arrange could be a set of cells that have a weighted association to other cells. Through preparing the weights of associations are modified and the yield is compared to wanted yield. Cycles are carried out until wanted precision isn't gotten for test information set. In hereditary calculation rules are characterized. Each run the show comprises of highlights which can unmistakably distinguish a course of assault. These rules are tried and after each cycle the rules with higher fitness factor are chosen and adjusted to make modern rules till craved location rate isn't accomplished.

Pros:

- Unknown assault detection rate is expanded by utilizing combination of hereditary calculation with signature based interruption discovery framework.

Cons:

- Careful choice of features to characterize rules needs information of the space.
- Detection rate depends on the preparing information and the rules provided.

2.5.6.3 Detection Time

Real-time

Real-time intrusion detection frameworks work online i.e. these intrusion detection frameworks capture packets from the network (live) for identifying unusual exercises. The execution of genuine time interruption discovery frameworks profoundly depends upon the number of

highlights chosen because it needs to compare those features with the features of the incoming packets at a very high rate. The number of features too influences the asset utilization of the genuine time framework [18].

Pros:

- Real-time frameworks distinguish anomalous behavior whereas it is happening which is wanted from an interruption location framework

Cons:

- Real-time frameworks require more assets.
- Real-time frameworks may get to be bottleneck.

Offline

Offline intrusion detection systems work offline i.e. these intrusion detection systems process saved attack data sets like KDD cup 99 data set. Offline intrusion detection systems provide information about the attack and help repair the damage caused by the attack.

2.5.7 Deployment Strategies for IDSs

There exist three vital areas where NIDS can be introduced within the network for successful monitoring of the network, as portrayed within the diagrams underneath.

2.5.7.1 Before the Gateway firewall.

In this point, the NIDS can keep track of all network occasions of interests, indeed those assaults which along these lines may come up short. Because it must handle huge activity, NIDS have to be introduced on a quicker machine so that examination is done in genuine time. Moreover it must be designed accurately so that number of untrue alerts can be decreased [34].

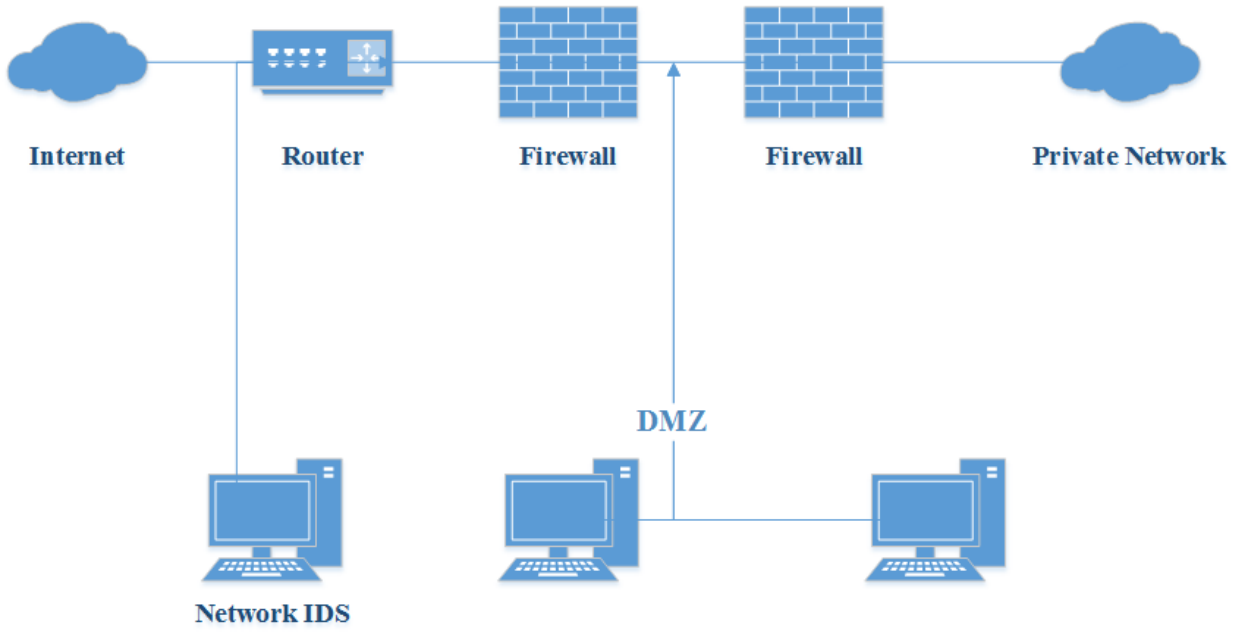

Figure 4 Network IDS placed before the Gateway Firewall [34]

2.5.7.2 In the DMZ (De-Militarized Zone):

Putting IDS inside the DMZ empowers it to screen the activity which is as of now incompletely filtered off through the gateway firewall as portrayed in figure 2 [34]. This diminishes the burden on the IDS but too limits its visibility.

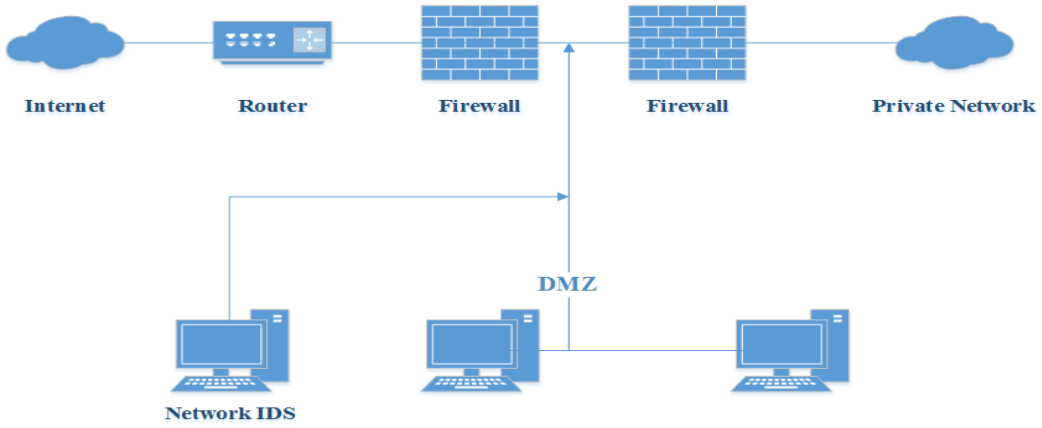

Figure 5 Network IDS in the DMZ [34]

2.5.7.3 inside the private corporative network

The final possibility where NIDS can be positioned is inside the corporate network as appeared in figure 3. Such a area points at checking the assaults developing from the neighborhood systems additionally those which are transmitted by means of firewall. As the number of assaults possible in this place is lesser than the preceding cases, this makes the application requests littler. In this case IDS creates few untrue cautions. The scope of visibility is restricted to inside the corporate network, in this way will not be able to distinguish the failed attacks as within the past cases.

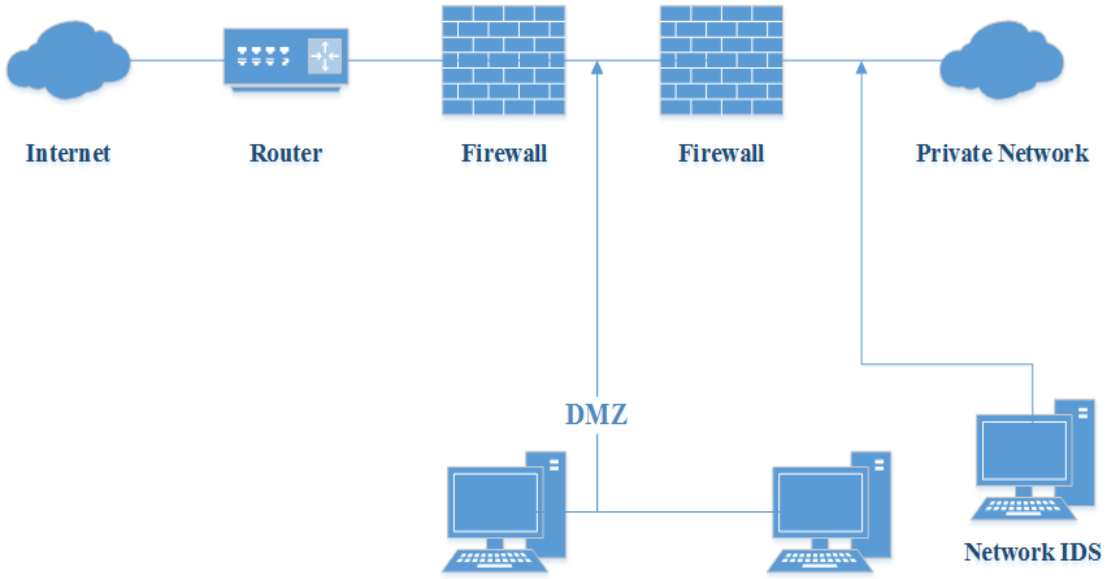

Figure 6Network IDS within the private network [34]

2.6 Deep learning methods and algorithms

Deep learning (too known as deep structured learning) is portion of a broader family of machine learning strategies based on artificial neural systems with representation learning. Learning can be supervised, semi-supervised or unsupervised Deep learning models such as deep neural systems, profound conviction systems, repetitive neural systems and convolutional neural systems have been connected to areas counting computer vision, machine vision, discourse acknowledgment, characteristic dialect handling, sound acknowledgment, social organize sifting, machine interpretation, bioinformatics, medicate plan, restorative picture investigation, fabric

review and board amusement programs, where they have delivered comes about comparable to and in a few cases outperforming human master execution [30]. It has a few layers of manufactured neural systems that carry out the ML handle. The primary layer of the neural network forms the crude information input and passes the data to the second layer. The second afterward until the desired result is achieved.

Artificial neural systems (ANNs), usually called neural systems (NNs), are computing frameworks enigmatically inspired by the biological neural systems that constitute creature brains. An ANN is based on a collection of associated units or hubs called artificial neurons, which freely show the neurons in a natural brain. Each association, just like the neural connections in a organic brain, can transmit a flag to other neurons [31]. An counterfeit neuron that gets a flag at that point forms it and can flag neurons associated to it. The "signal" at a association could be a genuine number, and the yield of each neuron is computed by a few non-linear work of the whole of its inputs. The associations are called edges. Neurons and edges ordinarily have a weight that alters as learning continues. The weight increments or diminishes the quality of the flag at a association. Neurons may have an edge such that a flag is sent as it were in the event that the total flag crosses that edge. Ordinarily, neurons are amassed into layers. Diverse layers may perform diverse changes on their inputs. Signals travel from the primary layer (the input layer), to the final layer (the yield layer), conceivably after navigating the layers different times. Agreeing to Liu machine learning can be classified as takes after, the taking after figure appears Liu’s classification of machine learning [29]

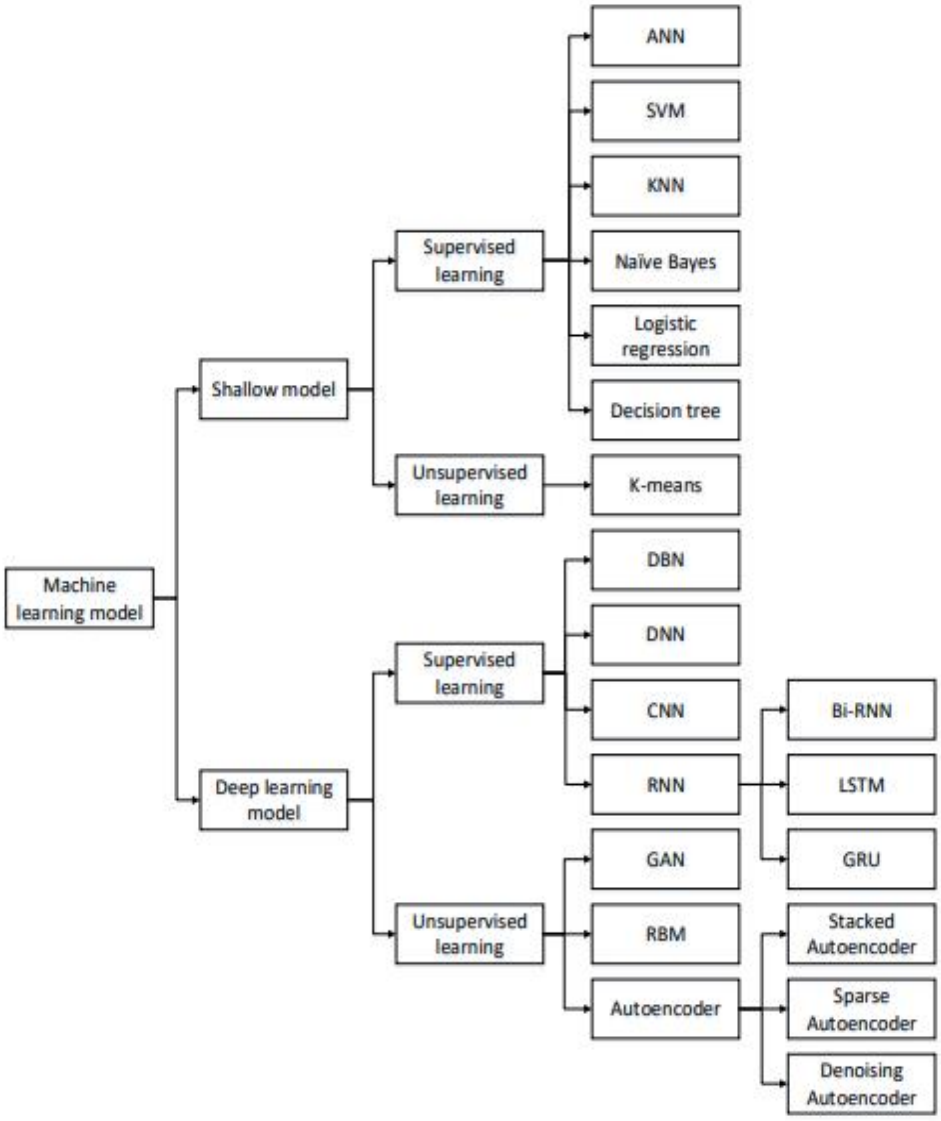

Figure 7 taxonomy of machine learning algorithm [29]

2.6.1 Classification of deep learning

2.6.1.1 Supervised deep learning

Supervised learning depends on valuable data in labeled information. Classification is the foremost common assignment in supervised learning (and is additionally utilized most habitually in IDS); in any case, labeling information physically is costly and time devouring. Consequently, the need of adequate labeled information shapes the most bottlenecks to administered learning. A few of the directed profound learning calculations are examined underneath:

2.6.1.1.1 DBN (deep brief networks)

ifferent RBM can be stacked to create a Deep Belief Network. They see precisely like Completely Associated layers, but they vary in how they are prepared. That's since they prepare layers in sets, taking after the preparing prepare of RBMs (portrayed some time recently) Be that as it may, DBNs and RBMs have kind of abandoned by the logical community in favor of Vibrational Auto encoders and GANs.

2.6.1.1.2 DNN (deep neural networks)

Neural networks composed by numerous hidden layers are known as deep neural network. Without a doubt, one of the quality focuses of DNN is their progressive organization. That layered organization permits sharing and reusing data. It is conceivable to choose particular highlights and dismiss futile subtle elements.

2.6.1.1.3 CNN (convolutional neural networks)

Convolutional Neural Networks utilize a work called convolution .The concept behind them is that rather than interfacing each neuron with all the following ones, we interface it with as it were a modest bunch of them (the open field). In a way, they attempt to regularize feedforward systems to maintain a strategic distance from overfitting (when the show learns as it were pre-seen information and can't generalize), which makes them exceptionally great in distinguishing spatial connections between the information. That's why their essential utilize case is Computer Vision and applications such as picture classification, video acknowledgment, therapeutic picture investigation and self-driving cars where they accomplish truly superhuman performance.They are moreover perfect to combine with other sorts of models such as Recurrent Networks and Autoencoders. One such case is Sign Language Recognition

2.6.1.1.4 RNN (recurrent neural networks)

Recurrent Neural Network (RNN) are a sort of Neural Network where the yield from past step are encouraged as input to the current step. In conventional neural systems, all the inputs and yields are autonomous of each other, but in cases like when it is required to anticipate the following word of a sentence, the past words are required and thus there's a got to keep in mind the previous words. In this way RNN came into existence, which illuminated this issue with the assistance of a Covered up Layer. The most and most vital include of RNN is Covered up state, which recollects a few data around a sequence.

RNN have a "memory" which recalls all data almost what has been calculated. It employments the same parameters for each input because it performs the same assignment on all the inputs or covered up layers to deliver the yield. This decreases the complexity of parameters, not at all like other neural systems.

Recurrent networks are perfect for time-related information and they are utilized in time arrangement estimating. They utilize a few shape of criticism, where they return the yield back to the input. You'll be able think of it as a circle from the yield to the input in network to pass data back to the network. In this manner, they are competent to keep in mind past information and utilize that data in its prediction. To achieve better execution analysts have adjusted the first neuron into more complex structures such as GRU units and LSTM Units. LSTM units have been utilized broadly in natural language processing in tasks such as language interpretation, speech generation, and text to speech synthesis.

Advantages of Recurrent Neural Network

2. An RNN recalls data all through the time. It is valuable in time series prediction as it was since of the highlight to remember past inputs as well. Usually called Long Brief Term Memory.
3. Recurrent neural network are indeed utilized with convolutional layers to amplify the successful pixel neighborhood.

Disadvantages of Recurrent Neural Network

1. Gradient vanishing and exploding problems.

2. Training an RNN may be an exceptionally troublesome task.
3. It cannot handle exceptionally long group of sequences if using tanh or relu as an activation function.

2.6.1.2 Un Supervised deep learning

Unsupervised learning extracts important feature data from unlabeled information, making it much easier to get training information. Be that as it may, the detection performance of unsupervised learning strategies is usually inferior to those of supervised learning strategies.

2.6.1.2.1 GAN (Generative Adversarial Network)

Generative Adversarial Systems (GANs) are a capable class of neural systems that are utilized for unsupervised learning. GANs are essentially made up of a framework of two competing neural organize models which compete with each other and are able to analyze, capture and duplicate the varieties inside a dataset. Generative Adversarial Systems (GANs) can be broken down into three parts:

- **Generative:** To learn a generative model, which explains how raw information is generated in terms of a probabilistic model?
- **Adversarial:** The training of a model is practiced in an adversarial setting.
- **Networks:** Use deep neural networks as the artificial intelligence (AI) algorithms for training purpose.

In GANs, there is a **generator** and a **discriminator**. The Generator produces fake tests of data(be it an picture, sound, etc.) and tries to trick the Discriminator. The Discriminator, on the other hand, tries to recognize between the genuine and fake tests. The Generator and the Discriminator are both Neural Systems and they both run in competition with each other within the preparing stage. The steps are rehashed a few times and in this, the Generator and Discriminator get superior and way better in their individual employments after each reiteration. The working can be visualized by the chart given underneath:

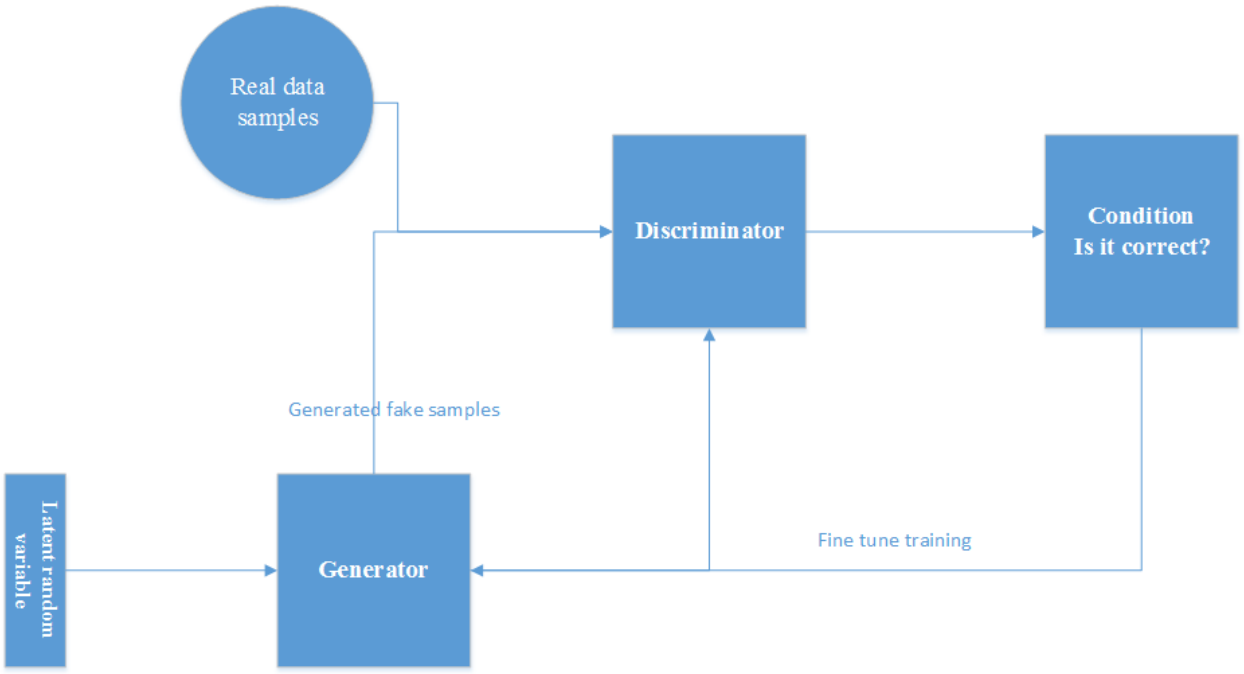

Figure 8 GAN operation principle [43]

Here, the generative model captures the distribution of information and is prepared in such a way that it tries to maximize the likelihood of the Discriminator in making a botch. The Discriminator, on the other hand, is based on a model that estimates the likelihood that the test that it got is gotten from the training data and not from the Generator.

The GANs are defined as a mini-max diversion, where the Discriminator is attempting to minimize its remunerate and the Generator is attempting to minimize the Discriminator's compensate or in other words, maximize its loss

2.6.1.2 .2 RBM (Restricted Boltzmann Machines)

Boltzmann machines are non-deterministic (or stochastic) generative Profound Learning models with as it were two sorts of hubs — covered up and obvious hubs. There are no yield hubs! This could seem unusual but typically what gives them this non-deterministic highlight. They don't have the ordinary 1 or sort yield through which designs are learned and optimized using Stochastic Angle Plunge. They learn designs without that capability and usually what makes them so extraordinary! One contrast to note here is that not at all like the other conventional

systems (A/C/R) which don't have any associations between the input hubs, a Boltzmann Machine has associations among the input hubs.

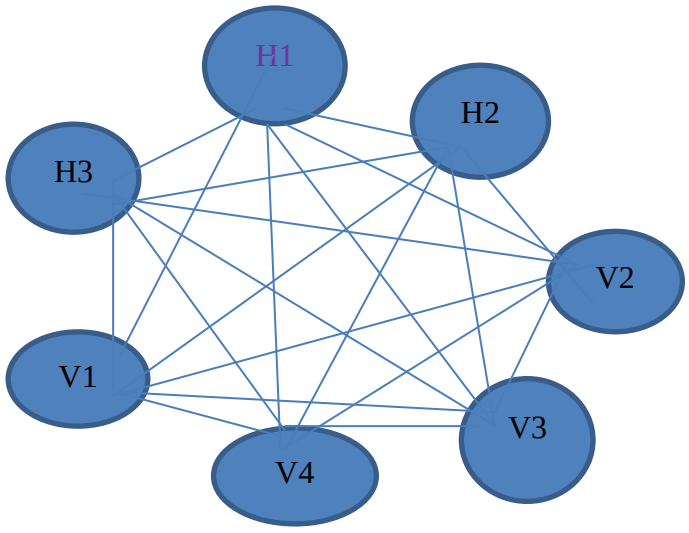

Figure 9Boltzman machine nodes [43]

This permits them to share data among themselves and self-generate ensuing information. We as it were degree what's on the unmistakable hubs and not what's on the covered up hubs. When the input is given, they are able to capture all the parameters, designs and relationships among the information. This is often why they are called Profound Generative Models and drop into the RBMs (Limited Boltzmann Machines) are a two-layered artificial neural network with generative capabilities. They have the capacity to memorize likelihood dissemination over its set of input. RBMs were concocted by Geoffrey Hinton and can be utilized for dimensionality diminishment, classification, relapse, collaborative sifting, include learning, and point modeling. Limited Boltzmann Machines are stochastic neural systems with generative capabilities as they are able to memorize a likelihood dispersion over their inputs. Not at all like other systems, they comprise of as it were input and covered up layers (no yields). RBMs are a extraordinary course of Boltzmann Machines and they are confined in terms of the associations between the obvious and the covered up units. This makes it simple to actualize them when compared to Boltzmann Machines. As expressed prior, they are a two-layered neural network (one being the obvious

layer and the other one being the covered up layer) and these two layers are associated by a completely bipartite chart. This implies that each hub within the unmistakable layer is connected to each hub within the covered up layer but no two hubs within the same bunch are associated to each other. This confinement permits for more productive preparing calculations than what is accessible for the common lesson of Boltzmann machines, in specific, the gradient-based contrastive divergence calculation.

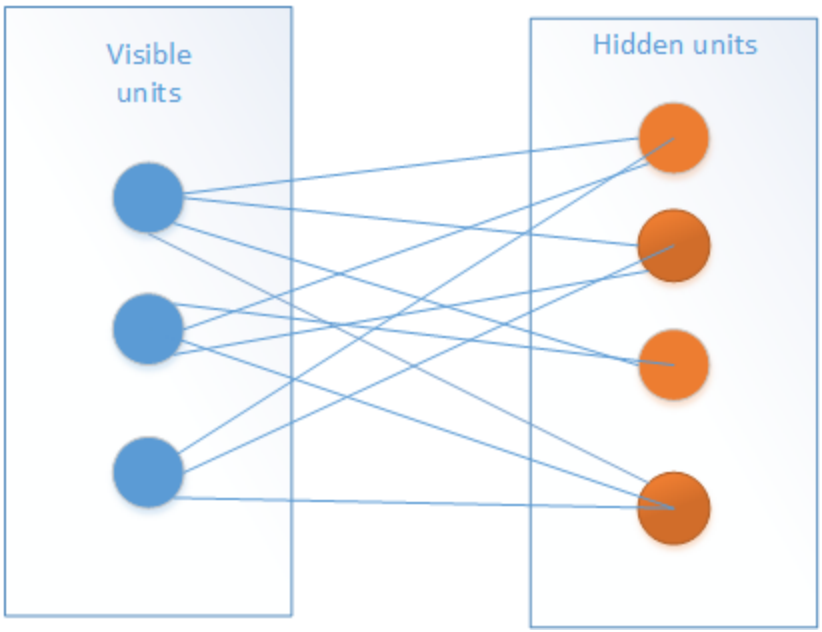

Figure 10Restricted Boltzmann Machines[43]

In the forward part of the training it takes input and produces a representation of it. In the backward pass they reconstruct the original input from the representation.

2.6.1.2 .3 Autoencoders

The network is first trained on the given input. The network tries to reproduce the given input from the features it picked up and gives an estimation to the input as the yield. The training step includes the computation of the mistake and back propagating the mistake. The ordinary

architecture of an Auto-encoder resembles a bottleneck. The schematic structure of an auto encoder is as follows:

Input layer output layer

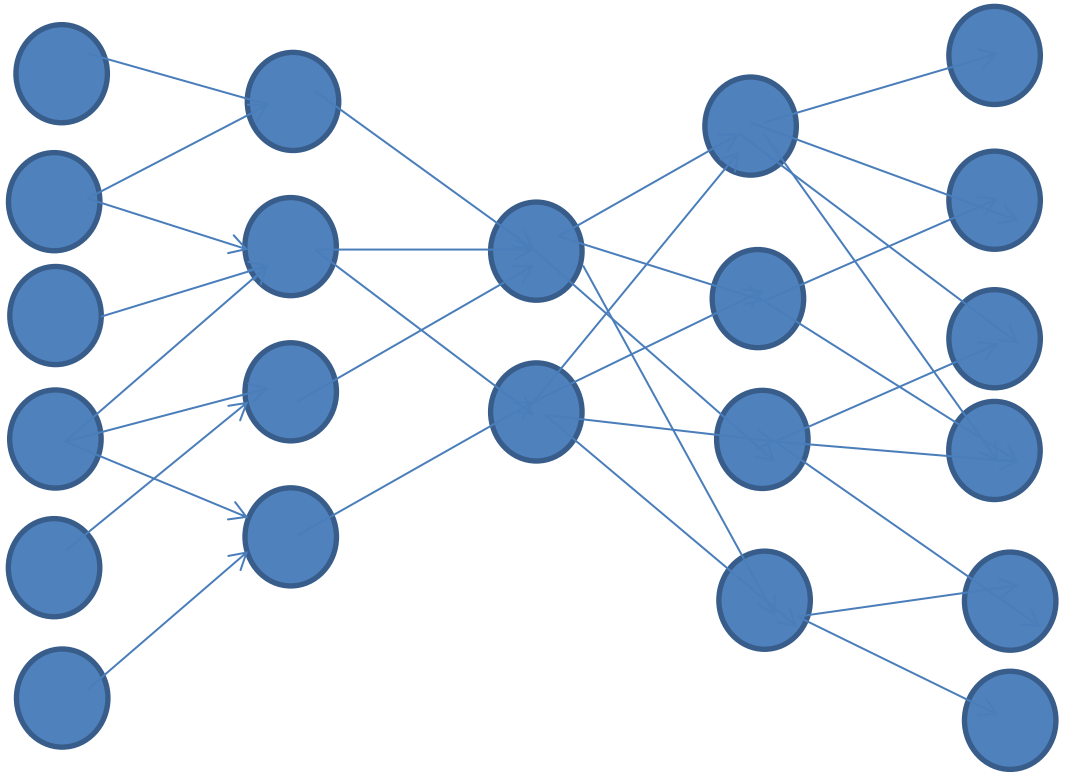

Figure 11 autoencoder operation [42]

Steps to training autoencoders involve the following:

Step 1: Encoding the input data

The Auto-encoder first tries to encode the data using the initialized weights and biases.

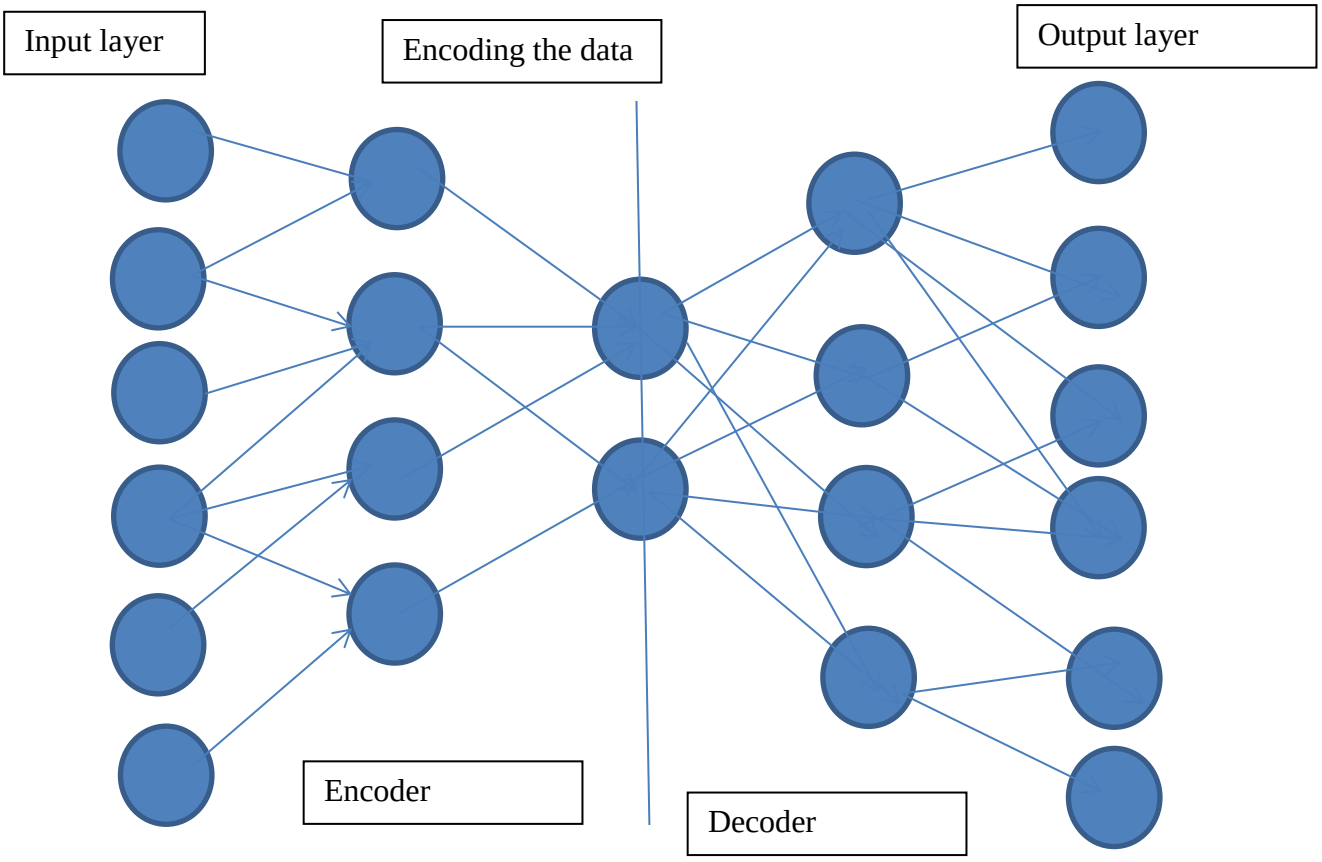

Figure 12 Encoding the input data [42]

Step 2: Decoding the input data

The Auto-encoder tries to reconstruct the original input from the encoded data to test the reliability of the encoding.

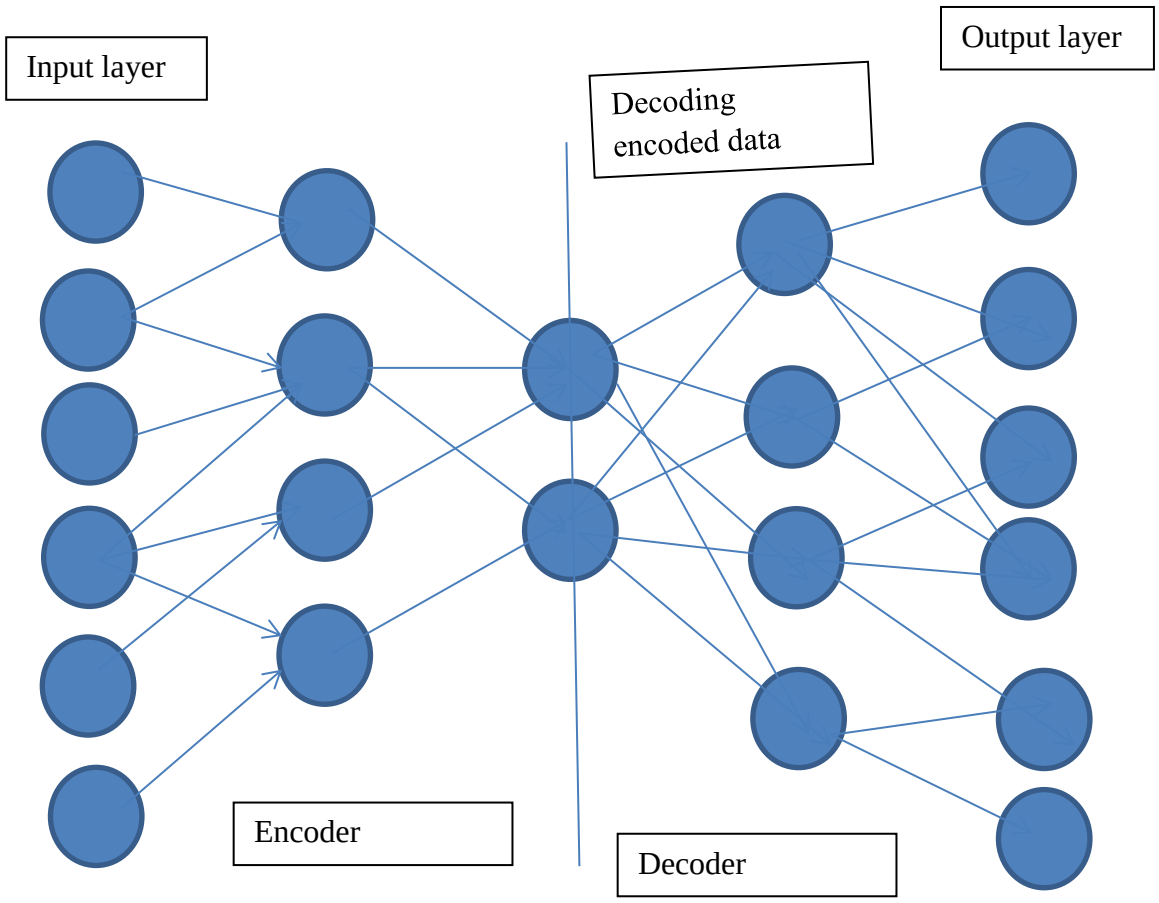

Figure 13 decoding input data [42]

Step 3: Back propagating the error

After the reconstruction, the loss function is computed to determine the reliability of the encoding. The error generated is back propagated.

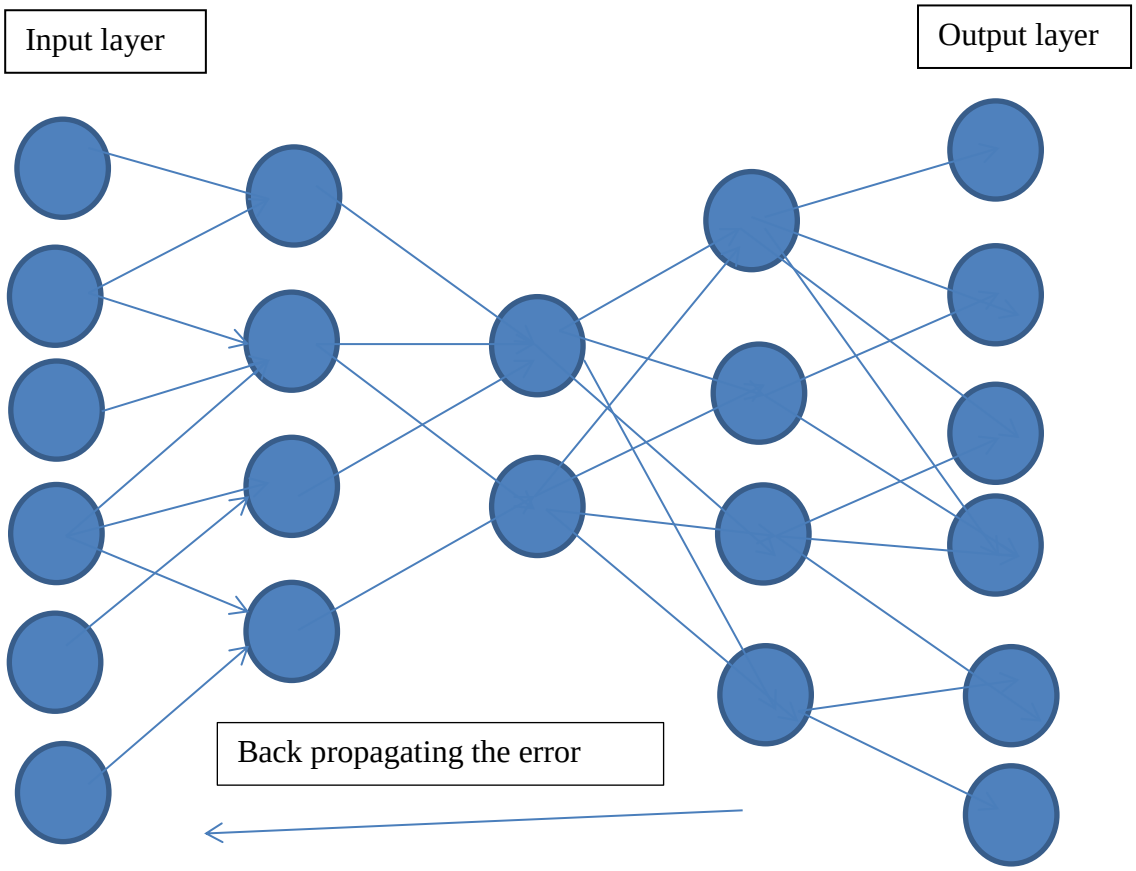

Figure 14 backward propagation error [42]

The above-described training process is reiterated several times until an acceptable level of reconstruction is reached. After the training process, only the encoder part of the Auto-encoder is retained to encode a similar type of data used in the training process.

Chapter 3: Related work

Intrusion detection systems have significant role in all networks and information system globally to get the required security guarantee. IDS is one of proposed methods to reduce malicious attacks [10]. This chapter discusses important topics about Network Intrusion Detection Systems which are related to my research area.

3.1 Anomaly based works

In this section research papers that are done on anomaly based intrusion detection system in iot network will be discussed.

In [26] they developed an IoT network attack detection system they used decision tree machine learning classifier to achieve accuracy of 82.71%. Their approach was evaluated using NLS-KDD dataset. Even though this dataset is another version of the KDD data set, it still suffers from various issues reviewed by McHugh [27].I believe the dataset should not be used as an effective benchmark dataset for IOT intrusion detection dataset because the data is collected from traditional network. Besides how they preprocess the data is not discussed, hence the dataset contains empty values and duplicate entries.

In [7] Suresh conducted his research on designing intrusion detection for iot network to maximize the accuracy of detection. He used kaggle dataset that has 41 features as basis for his research. He used weighted random forest for feature selection and trained the dataset on matlab. Finally an accuracy of 97% is achieved on this paper. Even though he got a good accuracy, it is not tested in real network environment. And no feature extraction technique is used while building the model.

In [34] network intrusion detection and classification system for iot network was implemented. They built machine learning classifier using KNN, Random Forest and Gaussian Naive Bayes algorithm. In KNN they got an accuracy of 94.4%, using Random Forest they got 88.8% and using Gaussian Naive Bayes they got 77.78%. They were successful in classifying doss, telnet and scanning attack. But they did not discuss how they collected their dataset and the number of

dataset. Data preprocessing, feature selection and feature extraction techniques are not clearly pointed out.

In [35] comparison of deep learning models and shallow are evaluated for using them in iot intrusion detection. They used NSL-KDD dataset for training their model. He dataset consists of four classes of data namely Dos, Probe, R2l and U2R.When using shallow model they got an average accuracy of 95.22% and when using deep model they got 99.2% concluding that deep models are better in accuracy. But they didn't pointed out which specific deep model algorithm they used and which shallow model algorithm used. And additionally using NSL-KDD dataset for iot network IDS is not recommended.

3.2 Signature based works

In this section researches that are done on signature based intrusion detection system in iot network environment will be discussed.

In [36] signature based intrusion detection system for iot network has been developed. They used optimized pattern recognition algorithm to detect a network attack. Even though signature based systems are accurate in detecting known attacks, they cannot detect zero day attack. This paper work is evaluated using KDD99 dataset but iot network dataset must be used to test iot IDS model. The test result is not stated in terms of accuracy or precision or percentage, only total dataset number and false positive result is stated.

In [37] they implemented signature based IDS in iot network scenario. But this paper didn't clearly showed the architecture or methodology. And it doesn't clearly show how they designed the signature or patter recognition and detection .It only stated their method was tested in cooja network simulator. And they did not quantified result of their research in terms of percentage or accuracy.

3.3 Hybrid based works

In [5] the researchers used hybrid approach to improve the accuracy of intrusion detection classifier. The dataset they used to build their model was NSL-KDD dataset which is the improved version of KDD dataset. They tried to compare accuracy obtained while using single

algorithm and when using hybrid algorithm. When using SVM algorithm they got 92.89% accuracy. When using ANN they got 80.33% accuracy and when using combined GA-DBN they got 98.82%. They used hybrid approach out performed individual algorithm. But using traditional network log NSL-KDD dataset for iot network intrusion detection is not better technique since iot network has different network characteristics and requirements. And if tested on real iot network Environment the performance of the model might degrade.

In [28] hybrid intrusion detection system that combines unsupervised learning technique with supervised deep learning based system is proposed. In their work they used Autoencoders unsupervised pre-training on the data to provide compressed and less noisy representation of the input space, while the final dense neural network functioned as the supervised classifier for their experimental intrusion detection scenario. They used publicly available wifi network log data which is called Aegean Wifi Intrusion Dataset (AWID). They trained their model using hybrid of the above learning algorithms. Their result showed that 99.8% accuracy in detecting Dos, impersonation and injection attacks. But it is better to test in real iot network environment than running and testing on matlab.

In [38] they conducted research on how to improve the security of WSN by developing hybrid intrusion detection system. They used NSL-KDD dataset to train both anomaly and misuse model. In this paper optimum-path forest algorithm is used to train anomaly based model and supervised optimum-path forest algorithm (MOPF) is used to train misuse detection model. The system was trained and tested on MATLAB R2014a. The accuracy of Anomaly detection module on this work was 80.95% and Misuse detection module was 96.20 %. But how they integrated anomaly based module with misuse detection module was not discussed.

Table 1 related work summary table

Reference	**Approach(method)**	**accuracy**	**Iot Dataset?**	**Test on real machine?**
[26]	Anomaly based used decision tree algorithm	87.1 %	X	X
[7]	Random forest	97%	✓	X
[34]	KNN	94.4%	Not stated	X
[35]	Deep learning but not specific algorithm is stated	99.2%	X	X
[36]	Signature based	Not stated	X	X
[37]	Signature based	Not stated	Not stated	Not stated
[5]	Hybrid IDS	98.8%	X	X
[28]	Hybrid IDS	99.8%	✓	X
[38]	Hybrid IDS	96.2%	X	X
This paper	Hybrid IDS	99.96%	✓	✓

Chapter 4: Proposed Solution

4.1 Introduction

In this Chapter, the proposed architecture for intrusion detection will be discussed. Different components of the proposed hybrid IDS architecture are discussed with their importance and techniques used to build those components. The Chapter presents the architecture with implemented algorithms. A hybrid intrusion detection system is a system that is composed of both signature based and anomaly-based IDS.A signature-based IDS captures and analyzes the network traffic for patterns that match a known rule or signature. These signature based systems are composed of many elements that help to find a pattern from network traffic. On the other hand anomaly based IDS try to identify anomalous behavior on the system. The IDS must be trained or taught to identify normal or legitimate. After that, the system alert notification if network behavior is outside normal.

4.2 System Architecture

In this Section, I have proposed architecture for hybrid intrusion detection system that can increase the detection performance of the intrusion detection system. It has different components as shown in Figure

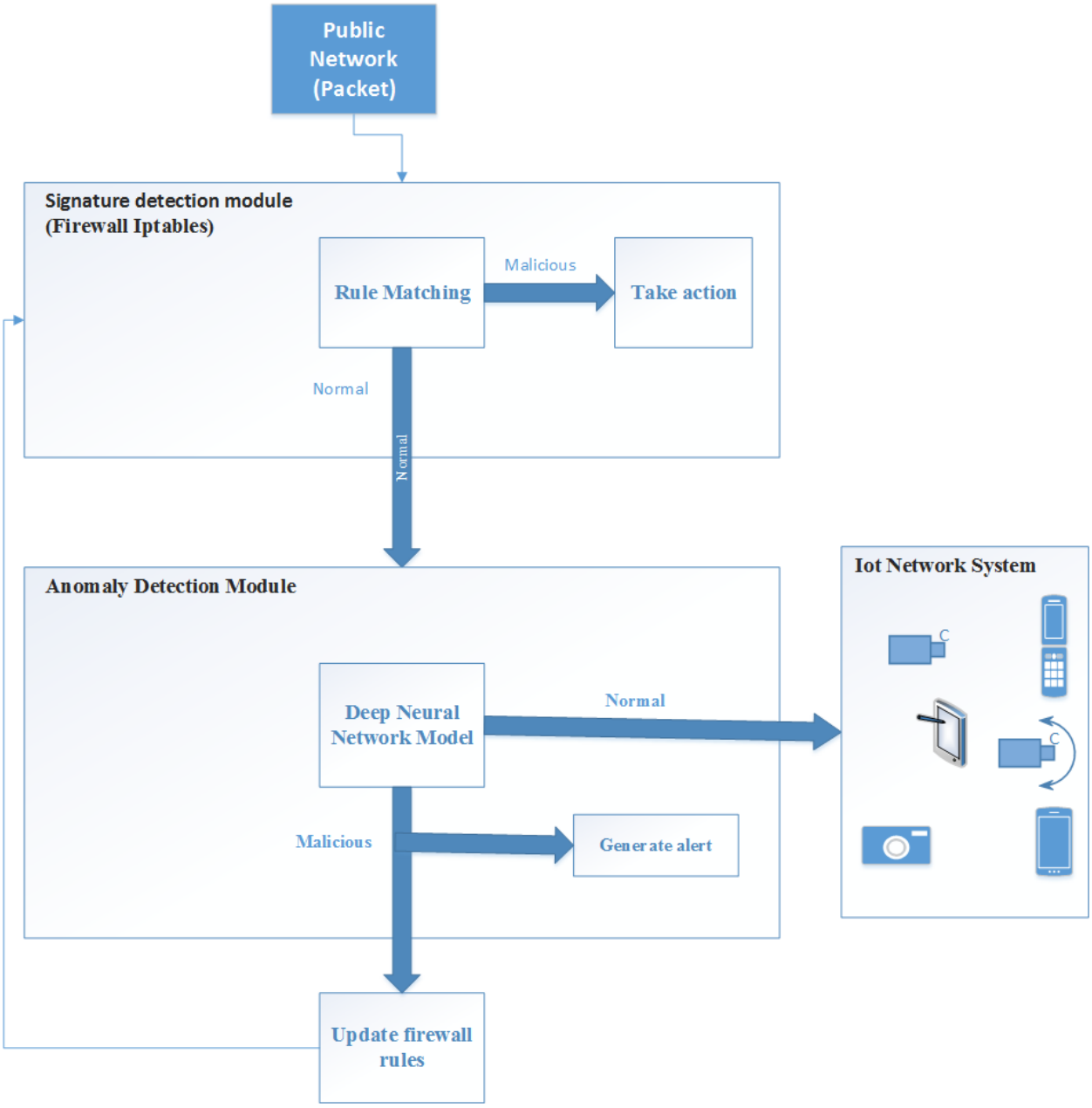

Figure 15 proposed architecture

The network packet will be captured from network interface and then signature based module will accept those packets and checks with pattern listed from the firewall policies. If it matches with rule action will be taken on the packet .If not the packet will pass to anomaly detection module .The anomaly detection module checks the incoming packet behavior with ANN model

if it is malicious alert will be generated and the malicious traffic will be written on firewall IPtables policies or chain. If the detected packet is normal then it will be ignored.

4.2.1 Components of the Proposed IDS

4.2.1.1 Signature Detection Module

The signature detection module is the first module that accepts packet from network interface card and filters the packets. The accepted packets will be checked with iptables rules if it matches with listed rules then action will be taken. If it doesn't match to any policies no action will be taken and it will be passed to next module.

As soon as network packet come to network layer the packet is passed to prerouting raw table, the raw table set mark on packets so that they should not be handled by connection tracking system. The aim is to do processing on the packet before kernel tracks its state. Then the packet is passed to prerouting chain of mangle table. The mangle table is used to alter ip headers of the packets (ex TTL) .This table marks an internal kernel mark on the packet6 for further processing in other tables. It doesn't touch the packet rather it adds a mark to the kernel representation of the packet. Once mangle table task is done it is passed to nat table prerouting chain. The nat table is used to implement network address translation rules. As the packet enters the network stack, rules in this table determine whether to modify packet source or destination ip address to impact the way that packet or any response traffic is routed. Now decision should be made if the current host (firewall hosting pc) is the final destination then it continue to mangle table input chain. After doing some processing on ip headers then it pass to filter table input chain where major filtering tasks are done. This table decides whether to let continue the packet continue to its destination or to deny its request. Once the packet passes this step it enters to security table input chain. Security table used to apply MAC (mandatory access control) to the packets. Once the it is applied the packet is passed down to network socket of the host application(network program).Now the firewall task is done and the host does whatever it want on the packet. If the current host is not the final destination now the inbound packet from nat table prerouting chain passes to mangle table forward chain. The mangle table task is the same as previous mangle table task.as the mangle table is done with packet it passes to filter table forward chain .After filter table done with the packet it pass to security table forward chain after MAC is applied on the

packet now decision is made. Is the packet routable? Or not. If it is not routable it is dropped if it is routable then it is forwarded to destination host by passing 2 tables. Those are mangle table post routing chain and followed by nat table post routing chain. The finally send the packet to data link layer for outbound interface.

4.2.1.2 Anomaly Detection Module

Anomaly detection module accepts network packet that has been filtered as normal from signature detection module. Once receive the packet this module lode the packet into Deep Neural Network model and then the trained model classify the input as anomalous or normal. If the packet is classified as normal then it will be ignored. But if the packet is classified as anomalous alert will be generated and IPtables firewall rules will be updated based on the type of attack. The following figure shows anomaly detection module.

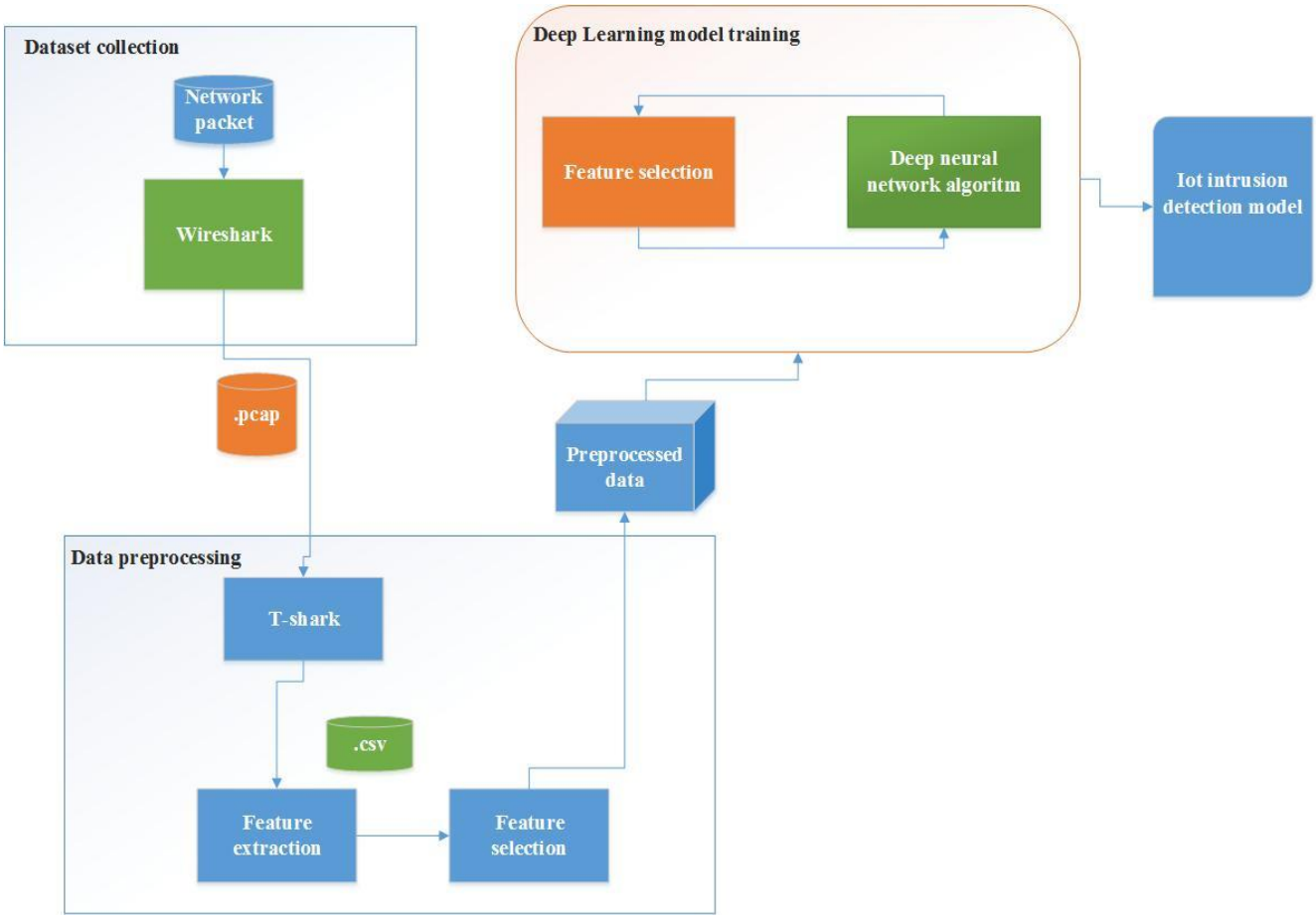

Figure 16 anomaly detection module.

Dataset collection

Network packet from LAN is collected using Wireshark which is network analyzer tools used in windows operating system. This tool can captures live network packet, save packet or export in csv (comma separated value) or pcap file format. In my case I exported pcap file not csv because Wireshark not allow exporting selected features. Hence we don't want all the features it need further preprocessing.

Data preprocessing

The exported pcap file from wire shark contains raw data. The required features (columns) can be extracted from pcap file using t-shark which is network analyzer tool in Ubuntu operating system Now the selected features from pcap file is exported into csv file.

Deep learning model training

The preprocessed data is fed into machine learning algorithm after feature extraction is done. Feature extraction begins from an initial set of quantified measured data and builds it features which are expected to be explanatory and non-duplicate, enhance learning and generalization. Feature extraction is related to minimizing dimensions.

When input information to an algorithm is too big to be processed and analyzed and it is suspected to contain duplicate entries, then it can be converted into a minimized form of features called feature vector. Finding smaller representation of initial features is called feature selection. The minimized representation of selected features contains vital and important information from input data, so that the necessary task can be performed on this reduced representation of the data instead of the original data. Finally, dataset with extracted features are trained to produce IOT intrusion detection model.

Chapter 5: Implementation and Performance Evaluation

5.1 Overview

I used tshark which is Ubuntu dependency for network packet data collection. And it is a network protocol analyzer. It allows capturing network packet data from real-time network traffic or reading previously captured packet. And it displays the data to standard output or write packets to file [32].It took place on one personal computer, one desktop computer, two rasp berry pi, one CCTV camera and one smart phone. Dataset preprocessing and training took place on cloud platform called Google Collaborator.

5.2 Tools used

Different tools have been used for implementing the proposed architecture. The tools are listed below:

- Wireshark to capture network packets
- The Linux tshark packet capturing tools to extract the needed features captured from Wireshark and then to csv data.
- Iptables to filter incoming network used as signature based detection.
- Google Collaborator to train my model.
- Python programming language to feed the csv dataset into DNN model.

5.3 Data collection

I used Wireshark and tshark to capture live network packet data from my local network. The data collection scenario includes one personal computer, one desktop computer; two rasp berry pi, one CCTV camera and one smart phone. The raw data collected is in pcap file format. The data collected includes both normal data as well as doss attack from live kali Linux terminal from desktop computer. The following figure shows data collection scenario:

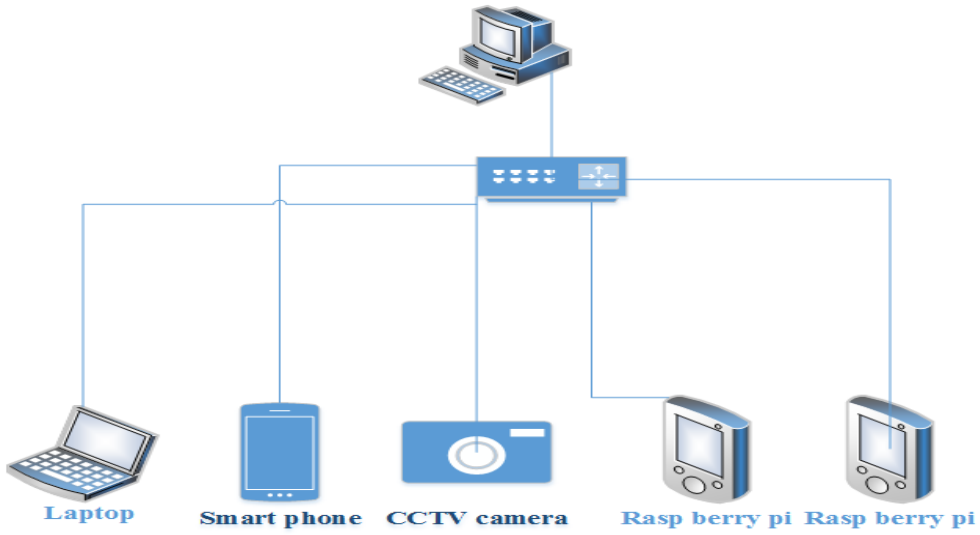

Figure 17 data collection scenario

5.3.1 Dataset description

The dataset consists of 92,079 data packet from which 76,186 doss attack anomalous data whereas the rest 15,892 is normal network packet data.

Table 2 Dataset distribution.

Attack type	Record	Percentage
Syndoss	15599	16.9%
Ssdpflood	19688	21.38%
Sslrenegotiation	40,899	44.42%
Normal	15897	17.26%
Total		92,079

Table 3 training dataset distribution

Attack type	Record	Percentage
Syndoss	10892	11.83%

Ssdpflood	13784	14.97%
Sslrenegotiation	28636	31.10%
Normal	11132	12.09%
Total		64,444

Table 4 testing dataset distribution

Attack type	Record	Percentage
Syndoss	4603	5.07%
Ssdpflood	5893	6.4%
Sslrenegotiation	12246	13.3%
Normal	4769	5.18%
Total		27511

Out of 92,079 seventy percent of the dataset is used for training purpose where as 30 % of the dataset is used for testing purpose.

5.3.2 Dataset Features

The collected pcap data has many features from all the selected features used in my study are the following listed in table below.

Table 5 packet capture features

Feature name	**Feature description**
No	The number of the packet in the capture file
Time	The timestamp of the packet.
Source	The address where this packet is coming from.
Destination	The address where this packet is going to

Protocol	The protocol name in a short
Length	The length of each packet
Src port	Source port address
Dst port	Destination port address
Frame number	The id of the frame
Frame length	The length of the frame
Eth.src	Source mac address
Eth.dst	Destination mac address

5.4 Data preprocessing

Data preprocessing or data cleaning is an important part of building neural network. It is preparing the data in way that can be fed into data processing algorithms.

The collected data is in raw format in 4 different pieces. So merging of all csv file in one combined dataset is crucial. Therefore; I combined them using windows command line by issuing the following command "copy *.csv dataset.csv" without quotation mark. The second task is filling empty values with zero, converting text values into numbers, dropping duplicate values and removing rows which has only single entry and removing entry that has low variance are done on cloud platform which is Google Collaborator.

5.5 Training

The preprocessed dataset is fed into deep neural network with one input layer, one output and one hidden layer. The input layer has 12 nodes with rectified linear activation function and the hidden layer has 24 nodes with sigmoid activation function and the output layer has one node with sigmoid activation function. The model used is deep neural network because it is able to learn nonlinear relationship but single layer neural network is only able to learn linear relationships among the data. Finally the model is compiled using binary cross entropy loss type

and Adam optimizer. Detail codes of training are explained on appendix2. Parameter setting is explained on the table below.

Table 6 parameter setting

Parameter	**Value**
Activation function	Sigmoid
Loss	Binary cross entropy
Optimizer	Adam
Epochs	250
Batch size	32

5.6 Implementation of the components

Signature based module is the baseline of this work using Iptables as first line signature detection. Iptables are configured as firewall on Ubuntu operating systems to filter the incoming network traffic. The configuration of IPtables is tiresome due to its dependency on command line commands no graphical user interface is available to configure IPtables.

5.6.1 Signature Based module

Signature based module is implemented using Linux iptables firewall. First iptables is installed on Linux machine. It is followed by defining or adding rules on a chain. Finally persisting all changes made on iptables permanent. Detail configurations are defined in appendix1.

5.6.2 Anomaly Detection

The anomaly detection module is implemented using deep neural network model. Once this model is deployed on the Linux machine the following steps were followed

Step1. Capturing data from firewall filtered packet

The following command captures the packets and writes them in file called ids.pcap

Tshark –w ids.pcap

Step2. Converting the pcap file into csv file:

tshark -r ids.pcap -T fields -E header=n -E separator=, -e ip.src -e ip.dst -e ip.proto -e ip.len -e tcp.srcport -e tcp.dstport -e tcp.len -e udp.srcport -e udp.dstport -e frame.len > /home/aschalew/Desktop/testing/keras/tshark/ids.csv

Step 3. Activating python virtual environment to run python code

source /home/aschalew/virtualenviroment/project1/bin/activate

Step 4. Now run IDS python script to detect an attack

python deploy_ids.py

5.7 Experiments and Results

This section discuses experiments performed while doing the thesis. The experiment involves getting better accuracy be altering the training parameters and dataset.

Experiment1

Using sigmoid activation function in the hidden and output layer the following is the result

```
Epoch 1/250
1928/1928 - 3s - loss: 0.0724 - accuracy: 0.9895 - val_loss: 0.0127 - val_accuracy: 0.9994
Epoch 2/250
1928/1928 - 3s - loss: 0.0074 - accuracy: 0.9996 - val_loss: 0.0055 - val_accuracy: 0.9995
Epoch 3/250
1928/1928 - 3s - loss: 0.0038 - accuracy: 0.9996 - val_loss: 0.0039 - val_accuracy: 0.9995
```

Figure 18 accuracy experiment2

Table 7 confusion matrix experiment2

TN	**25,173**	**FP**	**14**
FN	**0**	TP	5,199

Precision

Precision is the total number of true positives divided by the sum of true positives and false positives.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5.7.4}$$

$$= \underline{0.9973}$$

Recall

Recall is the capability of the model to identify all important points from the entire dataset. It can be directly defined as the total number of true positives divided by the sum of false negatives and true positives. False negatives are data entries that are recognized as negative by the model but they are actually positive(wrong) .True positives are data entries that are recognized as positives by the model and it is actually positive(right)

$$\text{Recall} = \frac{TP}{TP+FN} \tag{5.7.5}$$

$$= \underline{1}$$

F1 score

The F1 score is the harmonic mean of precision and recall it takes into consideration of both measurements.

$$\text{F1} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.7.6}$$

$$= \underline{0.998}$$

Experiment2

Using softmax activation function in the hidden and output layer the following is the result

```
Epoch 1/250
1928/1928 - 3s - loss: 12.6050 - accuracy: 0.1734 - val_loss: 12.6401 - val_accuracy: 0.1711
Epoch 2/250
1928/1928 - 3s - loss: 12.6050 - accuracy: 0.1734 - val_loss: 12.6401 - val_accuracy: 0.1711
Epoch 3/250
1928/1928 - 3s - loss: 12.6050 - accuracy: 0.1734 - val_loss: 12.6401 - val_accuracy: 0.1711
Epoch 4/250
```

Figure 19 accuracy experiment 1

That means 0.17 accuracy. The following Table shows the confusion matrix

Table 8 confusion matrix experiment1

TN 0	**FP 25,187**
FN 0	TP 5,199

Precision

Precision is the total number of true positives divided by the sum of true positives and false positives. Equation 5.7.1 shows the formula how to calculate precision.

$$\text{Precision} = \frac{TP}{TP + FP} \qquad (5.7.1)$$

$$= \underline{0.17}$$

Recall

Recall is defined as capability of the model to identify all important points from the entire dataset. It can directly be defined as the total number of true positives divided by the sum of false

negatives and true positives. False negatives are data entries that are recognized as negative by the model but they are actually positive(wrong) .True positives are data entries that are recognized as positives by the model and it is actually positive(right). Equation 5.7.2 shows the formula how to calculate recall

$$\text{Recall} = \frac{TP}{TP+FN} \qquad (5.7.2)$$

$$= \underline{1}$$

F1 score

The F1 score is the harmonic mean of precision and recall it takes into consideration of both measurements. The following equation shows how to calculate F1score

:

$$F1 = 2 * \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})} \qquad (5.7.3)$$

$$= \underline{0.29}$$

Experiment3

I modified experimental variables such as epochs and number nodes and batch size but the result is unchanged the following figure shows accuracy graph with respect to epochs. It is shown in the figure20 below.

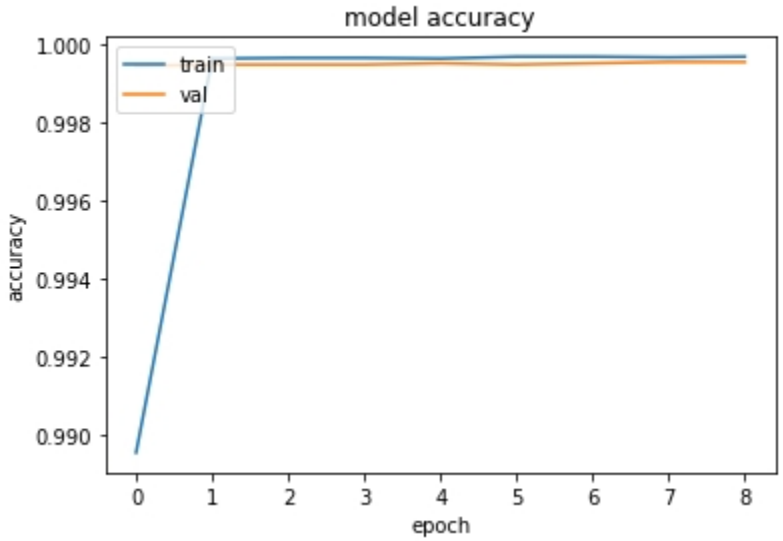

Figure 20 accuracy graph with epochs

5.8 Performance Evaluation

In this section, I measured the performance of the proposed hybrid intrusion detection system. To conduct this evaluation, I consider the requirements of hybrid intrusion detection system. Therefore, the evaluation includes those measures like accuracy, performance, completeness and scalability. Standard metrics for measuring the performance of IDS is evaluated by computing measures from the value in the confusion matrix shown in Table 5.5.

Table 9 Confusion matrix

		Actual Class	
		Negative class(Normal)	Positive class(Malicious)
Predicted Class	Negative class(Normal)	25,173	14
	Positive class(Malicious)	0	5,199

5.8.1 Performance Evaluation: Accuracy

As discussed previously the effectiveness of hybrid intrusion detection system is measured in terms of accuracy in which it identifies how much do the IDSs classify the coming packet as normal and attack. The accuracy of the proposed system is calculated using Equation 5.8.1.

$$\text{Accuracy} = \frac{TP+TN}{TN+TP+FP+FN} \tag{5.8.1}$$

In measuring the accuracy of anomaly based module system I used 30,386 total test dataset from which I got TN=25, 173, TP=5, 199, FP=14 and FN=0 from this we can calculate the accuracy as 99.95%.

5.8.2 Performance Evaluation: Performance

The main objective of this performance evaluation is to identify whether it add noticeable overload or not to the IDS. In particular, the following measures will be used to assess the IDS's performance including accuracy

$$\text{False Positive Rate (FPR) or False Alarm Rate (FAR)} = \frac{FP}{TN+FP} \tag{5.8.2}$$

$$\text{Precision} = \frac{TP}{TP+FP} \tag{5.8.3}$$

$$\text{Recall or True Positive Rate or Detection Rate (DR)= Recall} = \frac{TP}{TP+FN} \tag{5.8.4}$$

$$\text{F1score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.8.5}$$

The other issues which are useful to measure the performance of IDS are training time and testing time in which training time is the time needed by the algorithm to build the model and testing time is the time needed by the classifier to classify new example using a given model. Even if these issues are dependent on data set size doing it fast enough is important to avoid the slowness or overload on the network. But the most commonly used are detection rate and false alarm rate. From this we can calculate each for detection approaches. For proposed approach I get FPR = 0.00056, Precision = 0.997, Recall = 1 and F1score = 0.998.

5.8.3 Performance Evaluation: Completeness

The aim of this completeness is to minimize false alarm rate. When the false alarm rate decreases we can assume that the IDS will starts examining both known and unknown attacks in a network. So for our case false alarm rate for the hybrid detection is less than the single detection technique so from this we can conclude that the proposed IDS is better detecting network attack and relatively it is complete

5.8.4 Performance Evaluation: Scalability

Scalability is the other issue which responsibly identify whether the IDS works in large scale network or not. And I have tested it using dataset having 30,386 instances and I got an accuracy of 99.97%. The proposed system easily accepts the incoming data in .pcap format and convert it into csv file and classify it.

5.9 Discussion

The previous two chapters explained the techniques I used how to improve the performance of IDS in IOT environment. This paper goal is to achieve the objective listed in the chapter1 of this thesis. The objective of the study was to develop intelligent hybrid deep learning based intrusion detection system for IOT network by improving performance measurements using new IDS architecture.

I can say that this work have answered the research questions rose in this paper and achieved the objective of this research. Beginning from data collection I collected raw dataset from IOT network from my office LAN that contains IOT devices. The collected data is carefully preprocessed and trained with deep learning algorithm. The obtained model is combined with signature based iptables firewall. And it was evaluated using different performance measurement techniques. The combined result was promising giving both the functionality of intrusion detection and intrusion prevention system. As compared to using signature based system alone for intrusion detection, using combined anomaly based and signature based resulted in good performance IDS.

6. Conclusion and Future Work

6.1 Conclusion

Nowadays embedded systems and smart devices are rising in the everyday activities of our lives. But the network intrusion attacks are also changing and new attacks are emerging so IDS in IoT networks should also be improved continuously.

In this thesis I tried to develop an intelligent hybrid intrusion detection system that improves the performance of the existing Intrusion detection system in the IoT network. This work divides the entire work into two modules anomaly-based module and signature-based module. Signature-based modules detect attacks based on rules specified and only detect known attacks. Whereas anomaly-based modules can detect a new attack. Attacks not included in the signature-based module can be detected in the anomaly based module. The anomaly-based module is generated from deep learning trained model. If this model is trained with a sufficient and clean dataset with appropriate optimization techniques it can minimize false positivity and maximize the accuracy of intrusion detection.

In this paper, we have developed IDS in the IOT network that combines anomaly-based with the signature-based system. The anomaly bases system is built on top of iptables firewall. Signature detection is achieved using iptables rules. The combination of both modules resulted in the new architecture of IDS in the IOT network.

The main contribution of this thesis is to build a novel hybrid intrusion detection system that has high accuracy and low false positivity and prevent intrusions in IoT networks.

6.2 Future work

Even though this thesis result is encouraging and achieved its objective, it needs to be improved for the future. Future work that will be a continuation of this paper is listed below.

- This work is limited to detect doss attacks only but in the future, another type of attack will be included.
- Increasing accuracy of detection by using larger datasets and more data collection test buds (more IoT devices).

References

[1] K.Ansam, G.Iqbal, V.Peter, K.Joarder and A.Alazab, “A novel Ensemble of Hybrid Intrusion Detection System for Detecting Internet of Things Attacks,” Electronics, vol. 8, no. 11, p. 1210, Oct. 2019.

[2] Alghuried, A. "A Model for Anomalies Detection in Internet of Tings (IoT) Using Inverse Weight Clustering and Decision Tree" M.S Thesis, Dublin Institute of Technology,Dublin, 2017.Accessed on:October 25,2019.[online]. Available:https://arrow.dit.ie/scschcomdis /21427/D7WK7S.pdf

[3] T. Gethapriya and C.Shiven ,“Towards Deep-Learning-Driven Intrusion Detection for the Internet of Things”, University of Washington Bothell, Bothell, USA ,April 27, 2019.

[4] D.Zak, Cyber-attacks On IOT Devices Surge 300% In 2019, Sep 14, 2019. Accessed on: Dec 1, 2019. [Online]. Available: https://forbes.com/sites/zakdoffman/2019/09/14/dangerous-cyberattacks-on-iot-devices-up-300-in-2019-now-rampant-report-claims/#4aca6be65892.

[5] Z.Ying,L. Peisong and W. Xinhen, ”Intrusion Detection for IoT Based on Improved Genetic Algorithm and Deep Belief Network”, International Journal Of Electrical Electronics Engineers,2019.

[6] K. Ansam and G. Iqbal,” A novel Ensemble of Hybrid Intrusion Detection System for Detecting Internet of Things Attacks”, Internet Commerce Security Laboratory, Federation University Australia, Mount Helen, Australia, October 23, 2019.

[7] B. Suresh ,M. Venkatachalam and M. Saroja,” Towards Improved Random Forest based Feature Selection for Intrusion Detection in Smart IOT Environment”, International Journal of Innovative Technology and Exploring Engineering,vol. 8,no. 11,pp. 749-757,2019.

[8] ”Security in IoT Applications”, November 30,2016.Accesses on: October 23,2019.[Online].Available: https://www.eletimes.com/security-iot-applications.

[9] A. Eirini, W. Lowri, S. Malgorzata, T.Georgios and B. Peter .” A supervised intrusion detection system for smart home IoT devices”. IEEE Internet of Things, 2019.

[10] A. Ali Azawi,T. Sufyan,A. Belal."Survey on Intrusion Detection Systems based on Deep Learning", Periodicals of Engineering and Natural Sciences, vol. 7,no. 3,pp. 1074-1095,September 2019.

[11] N. Shone, T. N. Ngoc, V. D. Phai, et al., "A deep learning approach to network intrusion detection," IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 2, no. 1, pp. 41–50, Feb. 2018

[12]. Wikipedia, the free encyclopedia, "Intrusion Prevention System", Accessed on October 28 2020, http://en.wikipedia.org/wiki/ Intrusion detection system.

[13] Britannica, "computer-security", Accessed on October 28, 2020. .[Online].Available: https://www.britannica.com/technology/computer-security.

[14] IOT Agenda,, "IoT-security", Accessed on October 28 2020.[Online].Available: https://internetofthingsagenda.techtarget.com/definition/IoT-security-Internet-of-Things-security

[15] Ahmad, K., Verma, S., Kumar, N., & Shekhar, J ,"Classification of Internet Security Attacks", India, 2011.

[16] Force Point, "intrusion-prevention-system", Accessed on October 28 2020. .[Online].Available: https://www.forcepoint.com/cyber-edu/intrusion-prevention-system-ips.

[17] Cisco, "What is firewall", Accessed on October 28 2020.[Online].Available: https://www.cisco.com/c/en/us/products/security/firewalls/what-is-a-firewall.html

[18] H.Dawit, "Integrating Descriptive Modelling with Case Based Reasoning in Network Intrusion Detection", unpublished M.Sc. thesis, college of natural science, Addis Abeba University, Addis Abeba, 2015.

[19] K. Dinakara, "A Master's Thesis on Anomaly Based Intrusion Detection", Computer Science and Engineering Indian Institute of Technology, Kharagpur -721302, India, May 2007.

[20] A.Khraisat, I.Gondal, P. Vamplew, et al. Survey of intrusion detection systems: techniques, datasets and challenges. February 20, 2019.

[21] Kreibich C, Crowcroft J Honeycomb: creating intrusion detection signatures using honeypots. SIGCOMM Comput Commun Rev 34(1):51–56,2004.

[22] AirDefense "AirDefense",accessed on October 31,2020 .[Online].Available: http://www.lever.co.uk/airdefense-wireless-intrusion-detection-system-ids-uk.html

[23] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, E. Vázquez, Anomaly-based network intrusion detection: Techniques, systems and challenges,Computers & Security, Vol. 28, no. 1–2 ,pp. 18-28, 2009.

[24] Snort,"What is snort",accessed on October 31,2020.[Online].Available: https://www.snort.org/

[25]Openmeniak,"Snort_inline", accessed on October 31,2020.[Online].Available: http://openmaniak.com/inline.php

[26] S.Moreira, H.Marques, M.Lima, "DEA: Anomaly Detection in Smart Environments using Artificial Intelligence", Lanoms, 2019

[27] McHugh, J. Testing Intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. ACM Trans. Inf. Syst. Secur. 2000, 3, 262–294.

[28] An efficient deep learning model for intrusion classification and prediction in 5G and IoT networks.

[29] Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey

[30]Wikipedia,"Deep learning", accessed on November 1,2020.[Online].Available: https://en.wikipedia.org/wiki/Deep_learning.

[31]Wikipedia," Artificial neural network", accessed on November 1,2020.[Online].Available: https://en.wikipedia.org/wiki/Artificial_neural_network.

[32]Wireshark,"Tshark",accessed on November 12,2020. .[Online].Available: https://www.wireshark.org/docs/man-pages/tshark.html

[33] K. Dinakara, *" Anomaly Based Intrusion Detection"*, M.Sc. thesis, Computer Science and Engineering Indian Institute of Technology, Kharagpur ,India, May 2007.

[34] A. Kumar and T. J. Lim, "EDIMA: Early Detection of IoT Malware Network Activity Using Machine Learning Techniques," 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland, pp. 289-294, 2019

[35] A.A. Diro, N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things", Future Generation Computer Systems ,Vol. 82,pp. 761-768,2017.

[36] N.Uddin, H.Rahman, S.Vikram, H.Alqahtani, A Lightweight Signature-Based IDS for IoT Environment, arxiv e-prints,November,2018, accessed on: October 30, 2020.[online]. Available:https://ui.adsabs.harvard.edu/abs/2018arXiv181104582U/abstract.

[37] P. Philokypros , V.Vassilios, D.Ioannis,D.Michael,"A Signature-based Intrusion Detection System for the Internet of Things",Information and Communication Technology Forum(ICTF),Graz,Austria,Jul.2018.

[38] S.Mansour,"A Hybrid Intrusion Detection Architecture for Internet of Things", in international symposium on telecommunication, Tehran,Iran,2016.

[39] H.Fatima, H.Rasheed, A.Syed,M.Ayash, "Machine Learning in IoT Security: Current Solutions and Future Challenges".????

[40] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," IEEE Communications Surveys Tutorials, vol. 17, pp. 2347–2376, Fourthquarter 2015.

[41] B. Susilo and R. F. Sari, "Intrusion Detection in IoT Networks Using Deep Learning Algorithm," Information, vol. 11, no. 5, p. 279, May 2020.

[42] Geeksforgeeks, "ML | Auto-Encoders" accessed on: Dec. 25, 2020. [Online].Available: https://www.geeksforgeeks.org/ml-auto-encoders/.

[43] Geeksforgeeks, "Types of Boltzmann machines", accessed on: Dec. 25, 2020. [Online].Available: https://www.geeksforgeeks.org/types-of-boltzmann-machines/.

[44] Geeksforgeeks, "Generative Adversarial Network (GAN)", accessed on: Dec. 25, 2020. [Online].Available: https://www.geeksforgeeks.org/generative-adversarial-network-gan/.

Appendix 1

Iptables installation and configuration

Step 1. Installing IPtables by issuing the following command.

```
sudo apt-get update
sudo apt-get install iptables
```

Step 2. Defining Chain Rules

```
sudo iptables -A <chain> -i <interface> -p <protocol (tcp/udp) > -s <source> --dport <port no.>  -j <target>
```

Step3. Enabling Connections on HTTP

```
sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

Step4. Filtering Packets Based on Source

```
sudo iptables -A INPUT -s <ipnumber> -j ACCEPT
```

Step5. Persisting Changes

The iptables rules that we have created are saved in memory. That means we have to redefine them on reboot. To make these changes persistent after restarting the server, you can use this command:

```
sudo /sbin/iptables-save
```

Appendix 2

ANN Model training code

```
from google.colab import drive
drive.mount('/content/drive')
import pandas as pd
data=pd.read_csv('/content/drive/My Drive/syn_dos_with_normal_ssdpflood.csv')
data.head(10)

import pandas as pd
from tensorflow.keras.utils import get_file
# from google.colab import files
# data_to_load = files.upload()
from pandas import read_csv
# data = read_csv('syn_dos_with_normal_ssdpflood.csv', header=None)
# data.head(10)
# count the number of missing values for each column
#num_missing = (data[[1,2,3,4,5,6,7,8,9,10]] == 0).sum()
# report the results
#print(num_missing)
# deleting extra empty columns
# del data[11]
# del data[12]
# del data[13]
data.head(10)
#print values with empty entry nan
#print(data.isnull().sum())
#print(data[0][0])
Analyaze dataset
ENCODING = 'utf-8'
```

```
def expand_categories(values):
    result = []
    s = values.value_counts()
    t = float(len(values))
    for v in s.index:
        result.append("{}:{}%".format(v,round(100*(s[v]/t),2)))
    return "[{}]".format(",".join(result))
def analyze(df):
    print()
    cols = df.columns.values
    total = float(len(df))
    print("{} rows".format(int(total)))
    for col in cols:
        uniques = df[col].unique()
        unique_count = len(uniques)
        if unique_count>100:
            print("** {}:{} ({}%)".format(col,unique_count,int(((unique_count)/total)*100)))
        else:
            print("** {}:{}".format(col,expand_categories(df[col])))
            expand_categories(df[col])

analyze(data)
# calculate duplicates
dups = data.duplicated()
# report if there are any duplicates
print(dups.any())
# list all duplicate rows
print(data[dups])
# drop rows with missing values
data.dropna(inplace=True)
```

```
#print values with empty entry nan
print(data.isnull().sum())
data.count
Show columns with nan value
data.columns[data.isna().any()].tolist()
Fill emty fields with zero
data.fillna(0, inplace = True)
data.head(10)
from pandas import read_csv
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import OrdinalEncoder
from sklearn.preprocessing import OneHotEncoder
from keras.models import Sequential
from keras.layers import Dense
import numpy as np
from tensorflow.keras.callbacks import EarlyStopping
from keras.optimizers import SGD
import io
# load the dataset function
def load_dataset(filename):
        # load the dataset as a pandas DataFrame
        # data = read_csv(filename, header=None, low_memory=False)
 #      fill empty values with zero
        #         data.fillna(0, inplace = True)
        # retrieve numpy array
        dataset = data.values
        # split into input (X) and output (y) variables
        X = dataset[:, :-1]
        y = dataset[:,-1]
```

```
        # format all fields as string
        X = X.astype(str)
        # reshape target to be a 2d array
        y = y.reshape((len(y), 1))
        return X, y
# prepare input data :convert  categorical variables(textual column values) into numerical representations
def prepare_inputs(X_train, X_test):
        oe = OneHotEncoder(handle_unknown = "ignore")
        oe.fit(X_train)
        X_train_enc = oe.transform(X_train)
        X_test_enc = oe.transform(X_test)
        return X_train_enc, X_test_enc

# prepare target:converting the labels into numeric form so as to convert it into the machine-readable form. Machine learning algorithms can then decide in a
#       better way on how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning
def prepare_targets(y_train, y_test):
        le = LabelEncoder()
        le.fit(y_train)
        y_train_enc = le.transform(y_train)
        y_test_enc = le.transform(y_test)
        return y_train_enc, y_test_enc
# load the dataset
X, y = load_dataset('syn_dos_with_normal_ssdpflood.csv')
# split into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y.ravel(), test_size=0.33, random_state=1)
# prepare input data
X_train_enc, X_test_enc = prepare_inputs(X_train, X_test)
```

```
# prepare output data
y_train_enc, y_test_enc = prepare_targets(y_train, y_test)
# define the  model
model = Sequential()
model.add(Dense(12, input_dim=X_train_enc.shape[1], activation='relu', kernel_initializer='he_normal'))
model.add(Dense(8, activation='sigmoid'))
model.add(Dense(1, activation='sigmoid'))
# compile the keras model
#model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
# fit the keras model on the dataset
monitor = EarlyStopping(monitor='loss', min_delta=1e-3,
                        patience=5, verbose=1, mode='auto')
history = model.fit(X_train_enc, y_train_enc,validation_data=(X_test_enc,y_test_enc), callbacks=[monitor], epochs=250, batch_size=32, verbose=2)
#history = model.fit(X_train_enc, y_train_enc, validation_data=(X_test_enc,y_test_enc), epochs=150, batch_size=32, verbose=2)
```

Submitted by

_____ _____ _____

Student Signature Date

Approved by

1. _____ _____ _____

Advisor Signature Date

2. _____ _____ _____

Chairman, CNOS chair Signature Date

3. _____ _____ _____

Chairman, Faculty's Signature Date

Graduate Commission

4. _____ _____ _____

Dean, Faculty of Computing Signature Date

5. _____ _____ _____

Director, Post graduate office Signature Date