



JIMMA UNIVERSITY

Jimma Institute of Technology

Faculty of Computing and Informatics

Anomaly Based Intrusion Detection system In IoT Using Deep Learning Techniques

By

Hadush Gebremariam

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Networking

Advisor: Fisseha Bayu (Ph.D. Cand)

Co-Advisor: Gemechu Birhanu (MSc)

Jimma, Ethiopia

August 10, 2021

Submitted by:

Hadush Gebremariam
Student Name

Signature

Date

Approved by:

We the examiners' board approve that this thesis proposal has passed through the defense and review process.

1. Melkamu Deressa (PhD)

Signature

Examiner1

Signature

Date

2. Kebebew Aabau (Ass.Prof)

Signature

Examiner2

Signature

Date

3. Sofi Alemu (MSc)

Chairman

Signature

Date

4. Fisseha Bayu (PhD Cand)

Signature

Advisor

Signature

Date

5. Gemechu Birhanu (MSc)

Signature

Co-Advisor

Signature

Date

6. _____

Dean, Head of Department

Signature

Date

Declaration

I, therefore, affirm that this master's thesis is my original work and that I have properly recognized all sources of materials utilized in the Project. My master's thesis is dedicated to my family, specifically my mother, father, sisters, brothers, and girlfriend, for their unwavering encouragement and support during my master's studies at Jimma University.

Acknowledgments

First and foremost, I'd like to express my gratitude to my advisor, Fisha Bayu (Candidate Ph.D.), for his unwavering encouragement, input, and revisions to the report. I'd also like to thank Gemechu Birhanu (MSc), my co-advisor, for his continuous advice, encouragement, and feedback on my thesis. Second, I'd like to express my gratitude to Kebebew Ababu (MSc and Assistant Proposer) for his frequent encouragement and insightful feedback on the completion of my thesis. Thirdly, I'd like to express my gratitude to all of the teachers and staff at JIT for their assistance and support during my studies. Finally, I'd like to express my gratitude to all of my friends for their good wishes for me.

Abstract

The Internet of Things is the network of tiny objects. The main goal of IoT is to connect the objects to the internet for sharing and to communicate with each other without human interference to improve the quality of human life. IoT applications are widely applied in smart homes, smart healthcare, smart city, and smart logistics. The IoT network is simply faced with cyber security challenges. Since the devices of the IoT are tiny objects and they are resource-constrained to install advanced security mechanisms. The designers didn't consider security prevention but their main goal is addressing the IoT system to the whole world.

The anomaly-based intrusion detection system is a mechanism used to monitors the network activity and it alerts an alarm if any deviation is passed from the normal behaviors of the thresholds. However, when it is applied in the IoT network it requires huge computational processing, and battery power, as well as the false alarm rate, is high. The purpose of this study is to develop an anomaly-based intrusion detection system in IoT using deep learning techniques.

Recent researchers proved that the combination of the intrusion detection system with a deep learning mechanism is efficient and accurate in countermeasures the limitations of the traditional IDS for IoT systems. To develop the models the used algorithm is a deep neural network(DNN) which creates multiple hidden layers. Deep neural networks can learn in multiple levels, corresponding to different levels of abstraction from the dataset. The dataset used for learning and testing is collected from the IoT network which is combined_IoT3. The combined_IoT3 dataset is comprised of both normal traffic, and DoS attack traffic. The dataset is splitting into training and testing. The new model is generated after learning and testing by the DNN algorithm which is anomaly-based IDS. The result indicates that the accuracy of the model is 99.99 percent and the false alarm rate is decreased to zero percent. The new study outperformed in all metrics from the existing study. According to the results, the model is novel inaccuracy and false alarm rate. Therefore the deep neural network algorithm by combining with IDS is robust and effective with the prominent accuracy for securing the IoT network environment.

Keywords: IoT, IDS, Deep Learning, Anomaly-based IDS, DNN

Table of Contents

| | |
|---|-----|
| Declaration | ii |
| Acknowledgments..... | iii |
| Abstract | iv |
| List of Figure..... | ix |
| List Tables..... | ix |
| List of Acronyms | x |
| Chapter 1: Introduction | 1 |
| 1.1 Background | 1 |
| 1.2 Statement of The problem | 4 |
| 1.3 Motivation | 4 |
| 1.4 Objective of the Study..... | 5 |
| 1.4.1 General Objective of the Study | 5 |
| 1.4.2 Specific Objective of the Study | 5 |
| 1.5 Methodology | 5 |
| 1.5.1 Literature Review | 5 |
| 1.5.2 Deep Learning | 6 |
| 1.5.3 Data Collection..... | 6 |
| 1.5.3.1 Introduction toNodeMCU ESP8266 WiFi Module..... | 11 |
| 1.5.4 Data Cleaning and Data Transformation | 11 |
| 1.5.5 Tools used for Implementing..... | 11 |
| 1.6 Scope and Limitation | 14 |
| 1.7 Significance of the Study | 15 |
| 1.8 Document Outline | 15 |
| Chapter 2: Literature Review | 16 |
| 2.1 The Internet of Things: An Overview | 16 |
| 2.2 Characteristics of Internet of Things..... | 17 |
| 2.2.1 Interconnectivity..... | 17 |
| 2.2.2 Things-Related Services | 17 |

| | |
|--|----|
| 2.2.3 Heterogeneity..... | 17 |
| 2.2.4 Dynamic Changes..... | 17 |
| 2.2.5 Enormous Scale | 17 |
| 2.2.6 Safety | 17 |
| 2.2.7 Connectivity..... | 18 |
| 2.2.8 Overall Perception | 18 |
| 2.2.9 Reliable Transmission | 18 |
| 2.2.10 Intelligent processing..... | 18 |
| 2.3 Application of Internet of Things..... | 18 |
| 2.3.1 Smart Home..... | 18 |
| 2.3.2 Smart Grid | 19 |
| 2.3.3 Industrial Internet of Things | 19 |
| 2.3.4 Smart Retail | 19 |
| 2.3.5 Smart Healthcare | 19 |
| 2.3.6 Smart City..... | 20 |
| 2.4 Security Challenges in Internet of Things..... | 22 |
| 2.4.1 Vulnerabilities in Internet of Things | 23 |
| 2.4.2 Counter Measurement for IoT security Challenges..... | 25 |
| 2.4.2.1 Security Mechanisms for Perception layer | 25 |
| 2.4.2.2 Security mechanisms for Network Layer | 25 |
| 2.4.2.3 Security mechanisms for Application Layer | 27 |
| 2.5 Security Attack in the Internet of Things | 28 |
| 2.5.1 Wormhole Attack | 28 |
| 2.5.2 Sinkhole Attack | 28 |
| 2.5.3 Selective Forwarding Attack | 28 |
| 2.5.4 Sybil Attack | 28 |
| 2.5.5 Hello Flood Attack | 28 |
| 2.5.6 Denial of Service (DOS) Attack..... | 28 |
| 2.6 Requirement of Security..... | 29 |
| 2.7 IoT Architecture | 30 |
| 2.8 Introduction to Intrusion Detection System | 31 |

| | |
|--|----|
| 2.8.1 Classification of Intrusion Detection System | 33 |
| 2.8.2 Classification of IDS in Detection Techniques | 36 |
| 2.8.2.1 Signature-Based Methods..... | 36 |
| 2.8.2.2 Specification Based Methods | 36 |
| 2.8.2.3 Anomaly-Based Methods | 37 |
| 2.8.2.4 Hybrid Methods..... | 37 |
| 2.8.3 Placement Strategy of Intrusion Detection System | 37 |
| 2.8.3.1 Centralized IDS for IoT | 38 |
| 2.8.3.2 Distributed IDS for IoT | 38 |
| 2.8.3.3 Hybrid IDS for IoT | 39 |
| 2.9 Deep Learning: An Overview | 39 |
| 2.9.1 Deep Learning (DL) Techniques for IDSs | 39 |
| 2.9.1.1 Convolutional Neural Network (CNN) | 40 |
| 2.9.1.2 Recurrent Neural Networks (RNNs) | 41 |
| 2.9.1.3 Deep Autoencoders (AEs)..... | 42 |
| 2.9.1.4 Restricted Boltzmann Machine (RBM)..... | 42 |
| 2.9.1.5 Deep Belief Network (DBN)..... | 42 |
| 2.9.1.6 Generative Adversarial Network (GAN)..... | 43 |
| 2.9.1.7 Ensemble of DL Networks (EDLNs) | 44 |
| 2.9.1.8 Deep Neural Network..... | 44 |
| Chapter3: Related Work | 46 |
| Chapter 4: Proposed Solution | 50 |
| 4.1 Introduction | 50 |
| 4.2 System Architecture | 50 |
| 4.2.1 Components of the Proposed IDS | 51 |
| 4.2.1.1 DNN Training..... | 51 |
| 4.2.1.2 Centralized based Deployment Anomaly-based DNN Model | 52 |
| 4.2.1.3 Dataset collection | 52 |
| 4.2.1.4 Data Preprocessing | 52 |
| 4.2.1.5 Deep Learning Model Training..... | 52 |
| Chapter 5: Implementation and Performance Evaluation..... | 54 |

| | |
|---|----|
| 5.1 Overview | 54 |
| 5.2 Tools used | 54 |
| 5.3 Dataset..... | 54 |
| 5.3.1 Dataset Description..... | 54 |
| 5.3.2 Dataset Features..... | 56 |
| 5.4 Data preprocessing | 56 |
| 5.5 Training..... | 57 |
| 5.6 Implementation of the Components | 58 |
| 5.6.1 DNN Training Module | 58 |
| 5.6.2 Anomaly-based IDS Model..... | 58 |
| 5.6.2.1 Anomaly Monitoring..... | 58 |
| 5.6.2.2 Anomaly Classification | 59 |
| 5.6.2.3 Anomaly Alarm | 59 |
| 5.7 Experiments and Result..... | 59 |
| 5.8 Performance Evaluation | 61 |
| 5.8.1 Performance Evaluation: Accuracy | 63 |
| 5.8.2 Performance Evaluation: Precision | 63 |
| 5.8.3 Performance Evaluation: Recalls..... | 63 |
| 5.8.4 Performance Evaluation: F1 | 63 |
| 5.8.5 Performance Evaluation: Completeness..... | 64 |
| 5.8.5 Performance Evaluation: False Alarm Rate(FAR)..... | 64 |
| 5.8.5 Performance Evaluation: False Negative Rate(FNR)..... | 64 |
| 5.9 Discussion Results..... | 64 |
| Chapter 6: Conclusion and Future Work | 69 |
| 6.1 Conclusion..... | 69 |
| 6.2 Future work | 70 |
| Bibliography | 71 |

List of Figure

| | |
|--|----|
| Fig 1: Local area wireless Connection..... | 7 |
| Fig 2: IoT Network | 8 |
| Fig 3: NodeMCU and D-Link DWA | 9 |
| Fig 4: IP and MAC address of NodeMCU..... | 10 |
| Fig 5: NodeMCU IP address..... | 11 |
| Fig 6: Research Methodology..... | 14 |
| Fig 7: Application of IoT | 22 |
| Fig 8: IoT Architecture | 31 |
| Fig 9: IDS general deployment..... | 33 |
| Fig 10:Network-Based (Network IDS.[41])..... | 34 |
| Fig 11: Host-Based (HIDS)[41]..... | 35 |
| Fig 12: Classification of IDS Based On Detection Methods | 37 |
| Fig 13: Placement Strategy of Intrusion Detection System | 39 |
| Fig 14: DL techniques for IoT IDS..... | 40 |
| Fig 15: Illustration of convolution neural network working[11] | 41 |
| Fig 16: Illustration of deep belief network working[11] | 43 |
| Fig 17: General Structure of Deep Neural Network | 45 |
| Fig 18: proposed Architecture | 51 |
| Fig 19: Deployed Anomaly-based IDS Model | 52 |
| Fig 20: Structure of Deep neural Network..... | 53 |
| Fig 21: Accuracy of the DNN..... | 59 |
| Fig 22: The Accuracy DNN Model Graph | 60 |
| Fig 23: DNN Loss Graph..... | 61 |

List Tables

| | |
|--|----|
| Table 1: Dataset Distribution | 55 |
| Table 2: Features of Dataset | 56 |
| Table 3: parameter setting..... | 58 |
| Table 4: Confusion Matrix of The DNN Model | 62 |
| Table 5: The summary result of the proposed system | 65 |
| Table 6: The results of the existing papers | 66 |
| Table 7: Comparing the new study with existing paper [74]..... | 67 |
| Table 8: Comparing the new study with existing paper [78]..... | 67 |

List of Acronyms

IoT: Internet of Things

M2M: Machine to machine

WiMAX: Worldwide Interoperability for Microwave Access

UMTS: Universal Mobile Telecommunications System

CPU: Central Processing Unit

IDSs: Intrusion Detection Systems

ML: Machine Learning

DL: Deep Learning

Colab: Collaboratory

GPU: Graphics processing unit

CCTV: Closed-Circuit Television

DoS: Denial of service

DDoS; Distributed Denial Of Service

CPU: Central Processing Unit

3G: Third Generation

WiFi: Wireless Fidelity

IDES: Intrusion Detection Expert System

HIDS: Host-Based Intrusion detection System

OSSEC: Open Source Host-based Intrusion Detection System

AIDE: Advanced Intrusion Detection Environment

NIDS: Network IDS

HIDS: Host-Based IDS

CNN: Convolutional Neural Network

RNNs: Recurrent Neural Networks

AEs: Deep Autoencoders

RBM: Restricted Boltzmann Machine

DBN: Deep Belief Network

GAN: Generative Adversarial Network

EDLNs: Ensemble of DL Networks

ANN: Artificial Neural Network

BP: Backpropagation

NNs: Neural Networks

NodeMCU :Node MicroController Unit

USB: Universal Serial Bus

PCAP: Packet Capture

PKI: A Public Key Infrastructure

SDN: Software-Defined Networking

XACML: eXtensible Access Control Markup Language

Chapter 1: Introduction

1.1 Background

The Internet of Things (IoT) is a network of networked devices that can sense, act, and communicate with one another as well as with the outside world (i.e., smart things or smart objects), as well as sharing information and acting autonomously in response to real/physical world events, triggering processes, and creating services with or without direct human intervention[1]. The Internet of Things (IoT) is a network of everyday physical objects that can link to the Internet to digital devices or sensors on the Internet of "Things" which use existing network resources to communicate and synthesize data. These objects are the interconnected digital devices or sensors that are capable of collecting data and sharing this information over the globe of the internet. New applications and services are generated by such interactions between sensors, connectivity, and people and processes. The Things in the Internet of Things refer to these digital devices or sensors. IoT networks are essentially created by the interconnection of IoT devices that lie within the range of individual users, typically within a 10-meter range, and have an inconsistent topology that can change over time dynamically[2].

The Internet of Things (IoT) is transforming the IT industry and will be the next significant technological leap after the Internet. The Internet of Things market is predicted to increase from over 15 billion devices in 2015 to more than 75 billion by 2025. According to this estimate, each person on the planet will have at least 25 personal IoT gadgets. As a result, IoT is projected to have a significant impact on our lives in the not-too-distant future. During this time, WSNs will be integrated into IoT, and thousands of sensor nodes will connect to the Internet to perceive and monitor their surroundings. Soon, IoT will increasingly use WSNs to facilitate interaction between people and the environment. For example, improved environmental awareness will benefit our planet as a result of this integration. The goal of the Internet of Objects is to connect people and smart things at any time, in any place, to anything and anybody, via any network and service. As a result of adhering to this goal, IoT application areas will grow steadily and drastically in every part of life. We may now remotely feel and act on conditions in our homes or offices thanks to the various installations of IoT devices. WSN and IoT security is a major concern, especially if they are used for mission-critical operations. For example, in tactical military applications where a network security breach could result in friendly force losses on the battlefield. Another example

(IoT applications) from the health-care sector: According to a recent study, the majority of currently in-use systems lack effective security services that could protect patient privacy. None of the patients would be pleased if their personal health information was leaked owing to malfunctioning nodes or system failures. WSNs are susceptible to a variety of attack methods, posing serious security risks. Active and passive attacks are the two basic types of such attacks. Attackers that use passive attacks are usually concealed (camouflaged) and either destroy the network's functional components or tap the communications channel to obtain useful information. Eavesdropping, node demolition, node malfunctioning, node outage, and traffic analysis are some of the more common sorts of passive attacks. Active attacks, on the other hand, include an attacker interfering with the targeted network's functions and operations. The end outcome of this ill-effect could be the attacker's true goal, which security systems can identify (intrusion detection). As a result of these types of attacks, network services, for example, may become vulgarized. Active assaults include jamming, flooding, DoS, black holes, wormholes, sinkholes, and Sybil[3].

A network of things called IoT, which is powered by sensors, identifiers, software intelligence, and internet connectivity. IoT is therefore based on several previous technologies, including omnipresent information systems, sensor networks, and embedded computing. IoT may simply be regarded as the intersection of the internet, things, and data. Things/objects may be anything, such as appliances, vehicles, persons, livestock, or plants. The number of items would be enormous in the IoT, according to company and technological experts. As a new paradigm, IoT enables physical objects to share information for monitoring and/or control functions over the internet with the manufacturer, user, or any connected computer. In summary, IoT is a worldwide network that connects different objects via the internet at any time and anywhere. IoT's main objective is to monitor and control things from anywhere in the world. As a new communication paradigm, IoT can lead to the expanded connection of devices, systems, and services beyond machine-to-machine (M2M) communications.

In general, transport, environmental monitoring, medical and healthcare, home automation, energy management, media, agriculture, and security are the use cases of the Internet of Things. It should be noted that anyone or any machine may carry out monitoring and control of IoT services. A homeowner may, for example, track his or her home using a mobile device. In this simple instance, it is clear that security is a major concern for the prevention of unauthorized access and hackers[4].

IoT networks are also becoming more vulnerable to security threats with the growth in popularity. Cybersecurity assaults are becoming one of the most serious IoT security threats. Such attacks occur in different ways, targeting several IoT devices for various resources. In an IoT network, these attacks appear to compromise one or more devices that can be used as a "resource" or "platform" for attacks such as distributed denial-of-service and fraudulent activities such as ransomware, theft of opportunistic services, and ex-filtration of information[2].

While IoT technologies are important for enhancing real-world smart systems such as smart cities, smart homes, smart healthcare, and smart industry, the huge scale and ubiquitous nature of IoT systems has created new security challenges. For a variety of reasons, the Internet of Things is highly vulnerable to cyber-attacks. The first is that IoT components have limited capabilities in terms of both energy and processing resources, making it impossible for them to install and run security methods[5]–[7]. Because of their limited energy and compute capabilities, IoT devices cannot afford to install advanced security measures on top of these security concerns. New attack surfaces emerge regularly as a result of the IoT's interconnected and interdependent settings[8], [9].

IoT has several restrictions and limitations in terms of components and devices, including restricted processing capability, memory, and power consumption, as well as the heterogeneous and omnipresent nature of IoT, which adds to the problems[10]. Intrusion detection is one technique to improve the security problems of the Internet of things. Traditional intrusion detection system techniques, on the other hand, are less effective or inadequate for the security of IoT systems due to the aforementioned special characteristics, namely limited energy, ubiquitousness, heterogeneity, limited bandwidth capacity, and global connection. Various studies have examined various strategies for developing IDS for IoT systems, but the majority of the aforementioned surveys did not include the implementation of machine learning or deep learning techniques as detection mechanisms in IoT networks and associated lightweight devices[11]. Since IoT systems are resource constrained the traditional intrusion detection is insufficient to use in the internet of things. In addition, the traditional intrusion detection techniques have several limitations such as false alarm, unable to detect the new attack, and require enough computing resources and energy to run on the internet of things. But the internet of things is resource-constrained, like computing resources, small batteries, and tiny memories. To solve those limitations I have proposed an anomaly-based intrusion detection system in IoT using deep neural algorithms.

1.2 Statement of The problem

One of the critical issues that need to be addressed for the Internet of Things is security problems. A variety of obstacles prevent IoT devices from being secured and end-to-end protection in an IoT environment from being achieved. Since networking equipment and other artifacts are still a relatively new concept, protection hasn't always been a top priority during the design process of a product. Furthermore, since the Internet of Things is still in its early stages, many product designers and manufacturers are more concerned with bringing their goods to market as soon as possible rather than taking the appropriate precautions to ensure protection from the start. A key issue with IoT security is the usage of hardcoded or default passwords, which can lead to security breaches. Even if passwords are modified, they are often not secure enough to avoid infiltration. Another problem with IoT devices is that they are frequently resource-constrained and lack the compute resources required to implement strong security. As a result, many devices lack or are incapable of providing advanced security features[12].

IDS is one of the main methods used for information systems and network system security. IDS tracks the activities of a host or network, alerting the system administrator when a security breach is identified. Despite the sophistication of IDS technology for conventional networks, existing implementations are insufficient for IoT systems due to complex IoT features that affect the creation of IDS. In conventional networks, IDS agents are installed by the system administrator in nodes with higher processing power. IoT networks are typically made up of resource-constrained nodes. Finding nodes that can support IDS agents in IoT systems is, therefore, more complex. The challenges of the traditional IDS are ineffective in detecting new attacks and variants of known attacks, Anything that does not adhere to normal behavior is considered as an intrusion, has a high false-positive rate, and is vulnerable to error[13].

1.3 Motivation

Since IoT systems are limited energy, ubiquitous, heterogeneity, and limited bandwidth capability the traditional IDS is insufficient to secure the IoT system. Deep Learning techniques with high compute capacity are better suited to sophisticated IoT applications. Deep Learning (DL)-related approaches have lately found traction in a popular application for detecting network threats, such as those affecting IoT networks. This is because deep learning methods can detect both benign and abnormal behavior in IoT environments.

The study was concerned with an anomaly-based intrusion detection system for the internet of things based on deep learning techniques. Since an anomaly-based IDS is preferred over the signature and specification-based IDS on its ability to detect new attacks or zero-day attacks, for detecting cyber-attacks on IoT network systems, this is the preferable method. However, the existing intrusion detection system used on the traditional network is not sufficient for IoT networks, the IoT system is a constraint of many things. Deep learning techniques are prominent approaches for securing the resource constraints of IoT systems and these techniques can classify intrusions in high accuracy and with low false positives.

1.4 Objective of the Study

1.4.1 General Objective of the Study

The study's major purpose is to build anomaly-based IDS in IoT using Deep learning techniques to better and solve the Internet of Things' cyber-security issues. The proposed intrusion detection system controls the network traffic in the IoT network and classifies it as a DoS attack or normal traffic.

1.4.2 Specific Objective of the Study

To achieve the main objective I have performed the following specific objectives. Those are:- preparing and preprocessing the dataset for training and testing. Training and testing by using deep neural networks, design new architecture for IoT systems, develop a classifier engine or model and evaluate the model.

1.5 Methodology

The methodology is referred to the technique or procedure to overcome the problem definitions. The method is begun by reading and analysis of the existing state of security and privacy in the IoT. Next, the selected IDS is an Anomaly-based intrusion detection system for IoT using Deep Learning algorithms. To accomplish this study the required methods or techniques were defined in the following.

1.5.1 Literature Review

Reviewing the existing materials to achieve the main goals concerned on the Internet of Things, challenges of the Internet of Things, Intrusion Detection Systems, Machine Learning, and Deep Learning mechanisms.

1.5.2 Deep Learning

Deep learning is a form of machine learning in which a computer creates multiple layers as an output from a hierarchy of data based on experience. Both supervised and unsupervised deep learning techniques are available. Data can be categorized using supervised deep learning, while data patterns can be analyzed using unsupervised deep learning. Deep learning is linked to artificial intelligence, in which computers learn from experience and eventually replace human intelligence. Deep learning uses artificial neural networks to analyze large volumes of data with the aid of algorithms created by human intelligence. Artificial neural networks with different deep layers that allow them to learn are referred to as "deep learning." Each neural node in a neural network calculates the weighted values received from the previous layer and transfers the output values to the next layer. The final results obtained by the neural networks from the raw data can be called the result value of the last layer[14]. The used algorithm for this study is a deep neural network which consists of the input layer, hidden layer, and output layer.

1.5.3 Data Collection

The required materials to collect the dataset are:-Local area wireless IoT connection, NodeMCU(Node MicroController Unit), Wireshark, laptop, VMware, Kali Linux operating system, Ubuntu operating system, and smartphone. The dataset is collected by creating a local area wireless internet of Things connection. The local connection is created by using D-Link DWA-125 Wireless N 150 USB Adapter (rev.A2).

Status

Network status



No Internet access

Your device is connected, but you might not be able to access anything on the network. If you have a limited data plan, you can make this network a metered connection or change other properties.

Fig 1: Local area wireless Connection

And the devices are connected to the local area wireless connection through Connectify hotspot. Those devices connected to local area connections are:-NodeMCU(ESP_C62591), Huawei laptop, Toshiba Laptop, HUAWEI smartphone, and Techno Smartphone. The NodeMCU is connected to the created Local Area wireless connection by using the Arduino setup and by installing the esp8266 library. Next by calling the code in the Arduino setup, the NodeMCU is connected to the local area wireless Connection. After all the devices were connecting to the local area wireless connection the dataset is collected by tracking using Wireshark. The dataset which is consists of both normal and Dos attack traffic which is a PCAP Wireshark file. The Dos attack is collected by using Kali Linux through performing Dos attack by the command hping3 tool. Next, the PCAP file is changed to Comma-separated Value by using commands in Ubuntu.



Fig 2: IoT Network

The NodeMCU and D-Link DWA-125 Wireless N 150 USB Adapter (rev.A2) are shown in the figure below.



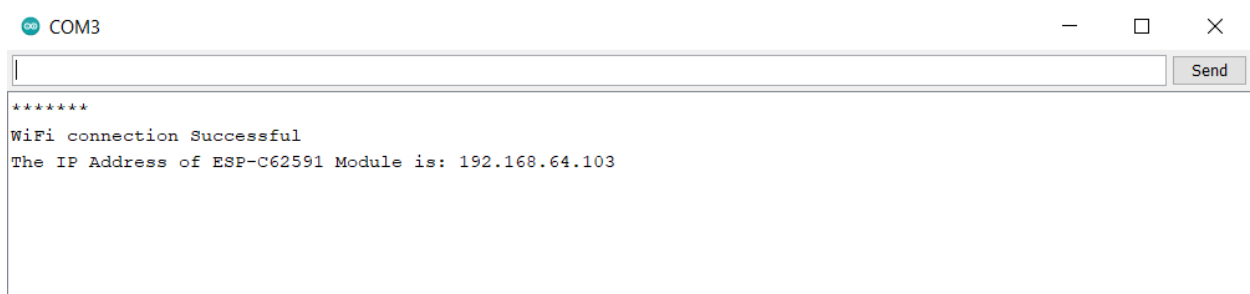
Fig 3: NodeMCU and D-Link DWA

After connected the NodeMCU the IP and MAC address is shown in the following figure.



Fig 4: IP and MAC address of NodeMCU

The following figure shows after the NodeMCU is connected successfully to the local area wireless connections and displayed on the Aurdino setup with its IP address of 192.168.64.103.

A screenshot of a terminal window titled 'COM3'. The window contains the following text: '*****', 'WiFi connection Successful', and 'The IP Address of ESP-C62591 Module is: 192.168.64.103'. There is a 'Send' button in the top right corner of the terminal area.

```
*****
WiFi connection Successful
The IP Address of ESP-C62591 Module is: 192.168.64.103
```

Fig 5: NodeMCU IP address

The total collected dataset record is 83225 from this 67.06% is a denial of service and 32.04% are normal traffic datasets.

1.5.3.1 Introduction to NodeMCU ESP8266 WiFi Module

The Node MicroController Unit is an open-source software and hardware development environment based on the ESP8266, a low-cost System-on-a-Chip. The Espressif Systems ESP8266 features all of the essential components of a computer, including a CPU, RAM, networking (WiFi), and even a current operating system and SDK. As a result, it's a great fit for a variety of Internet of Things (IoT) projects. And it's built on the ESP8266, which can connect things and allow data to be transferred via Wi-Fi. The ESP8266 module is a wifi-enabled system on a chip. It is used to create embedded Internet of Things (IoT) applications. It uses a 32-bit RISC CUP that runs at 80 MHz. Data RAM is 96 KB, boot ROM is 64 KB, and instruction RAM is 64 KB. The ESP8266 module is a low-cost wireless transceiver that can be used in Internet of Things end-point applications[15]–[18].

1.5.4 Data Cleaning and Data Transformation

The practice of correcting or deleting incorrect, corrupted, improperly formatted, duplicate, or incomplete data from a dataset is known as data cleaning. The preparation of datasets from one structure to suitable another format for learning and testing purposes in deep learning is called data transformation. I have prepared the dataset in the correct format which is called comma-separated values. Finally, it is combined into one file which is called combined_iot3.

1.5.5 Tools used for Implementing

The proposed model has been implemented using Colaboratory, or Colab for short is a Google research product that enables developers to write and execute Python code directly from their browser. For deep learning activities, Google Colab is an excellent tool.[19]. Colab is a free text

editor that uses to run the Python programming language to teach machine learning and deep learning methodologies. It comes with a deep learning-ready runtime as well as free access to a powerful Graphics Processing Unit (GPU)[20]. Here first I have uploaded the dataset to google drive, I have imported pandas as np, NumPy as np I have imported all libraries of the Keras. The following describes the implementation steps.

1. Get the Dataset: To create the deep learning model the first required is the dataset as the deep learning model is working on data. The collected dataset is prepared in comma-separated values format. Its name is combined_iot.csv which is comprised of both Dos and Normal traffic of the IoT networks.
2. Importing Libraries:- The imported libraries are Numpy library, pandas library, Matplotlib, read_csv, train_test_split, LabelEncoder, OneHotEncoder, sequential, dense, and EarlyStopping.
3. Importing the Datasets: The collected dataset is uploaded to google drive. Here the google Colaboratory and google drive are connected. Now, to import the dataset, by using the Pandas library's read CSV() function, which reads a CSV file and performs different actions on it.
4. Encoding Categorical data:- encoding is changing the categorical dataset into a numerical value, because the deep learning algorithm understands the dataset only the numerical values. To convert the categorical dataset to numerical values the used techniques are LabelEncoder and OneHotEncoder.
5. Splitting the Dataset into the Training set and Test set:- The dataset is separated into a training set and a testing set in deep learning data preprocessing. The training set is a subset of the dataset used to train the deep learning model. The testing set is a subset of the dataset used to test the deep learning model, and the model predicts the outcome using the test set. This is an important step in data preprocessing since it improves the performance of the deep learning model. For training, the used dataset is 80%, which is 66580, and for testing is 20%, which is 16645 records. 11250 records are Dos and 5395 records are Normal traffic.
6. Generate DNN Model:- The generated model is a sequential Keras layer consist of the input layer, hidden layer, and output layer. The input layer and output layers have 20 and 1 neurons respectively. There are five hidden layers from hidden layer one up to hidden layer five. The number of neurons from the hidden layer one to hidden layer five are

26,22,21,25, and 10 respectively. The used activation functions are rectified linear unit(ReLu) and sigmoid function. Rectified linear unit is used for the input layer and the hidden layer whereas sigmoid function is used for the output layer.

7. Compile Keras Model:- Compilation is the last stage in the model creation process. After the compilation is completed then the training phase can be started. In this study, the used loss function is cross-entropy. This loss is known as binary_crossentropy in Keras and is used to solve binary classification issues. Optimization is a crucial technique in deep learning that optimizes the input weights by comparing the prediction and loss function. optimizers are used to update weights and biases i.e the internal parameters of the model to reduce the error. The used optimizer for this study is adam. Adaptive Moment Estimation is abbreviated as Adam. Adam is a deep neural network-specific adaptive learning rate optimization technique[21], [22].
8. Fit Keras Model:- This phase is the training phase of the deep neural network through loading the dataset. By using the fit() function on the model, the model is trained by using the training dataset. Training takes place in epochs, with each epoch divided into batches. The phrase "batch size" is used in deep learning. It refers to how many training examples are used in a single iteration. An epoch is a word used in deep learning that refers to the number of passes the deep learning algorithm has made across the full training dataset. In this study, the utilized numbers of the epoch are 50, and the batch sizes are 32.
9. Evaluating the Model:- After the model is created the model is evaluated by accuracy, precision, recall, F1 score, and false-positive from the output of the confusion matrix.
10. Make Predictions:- For making predictions, the predict() method is used. The sigmoid activation function is used to activate the output layer. The prediction's result is the probability in the range of 0 to 1 which means Dos and Normal.

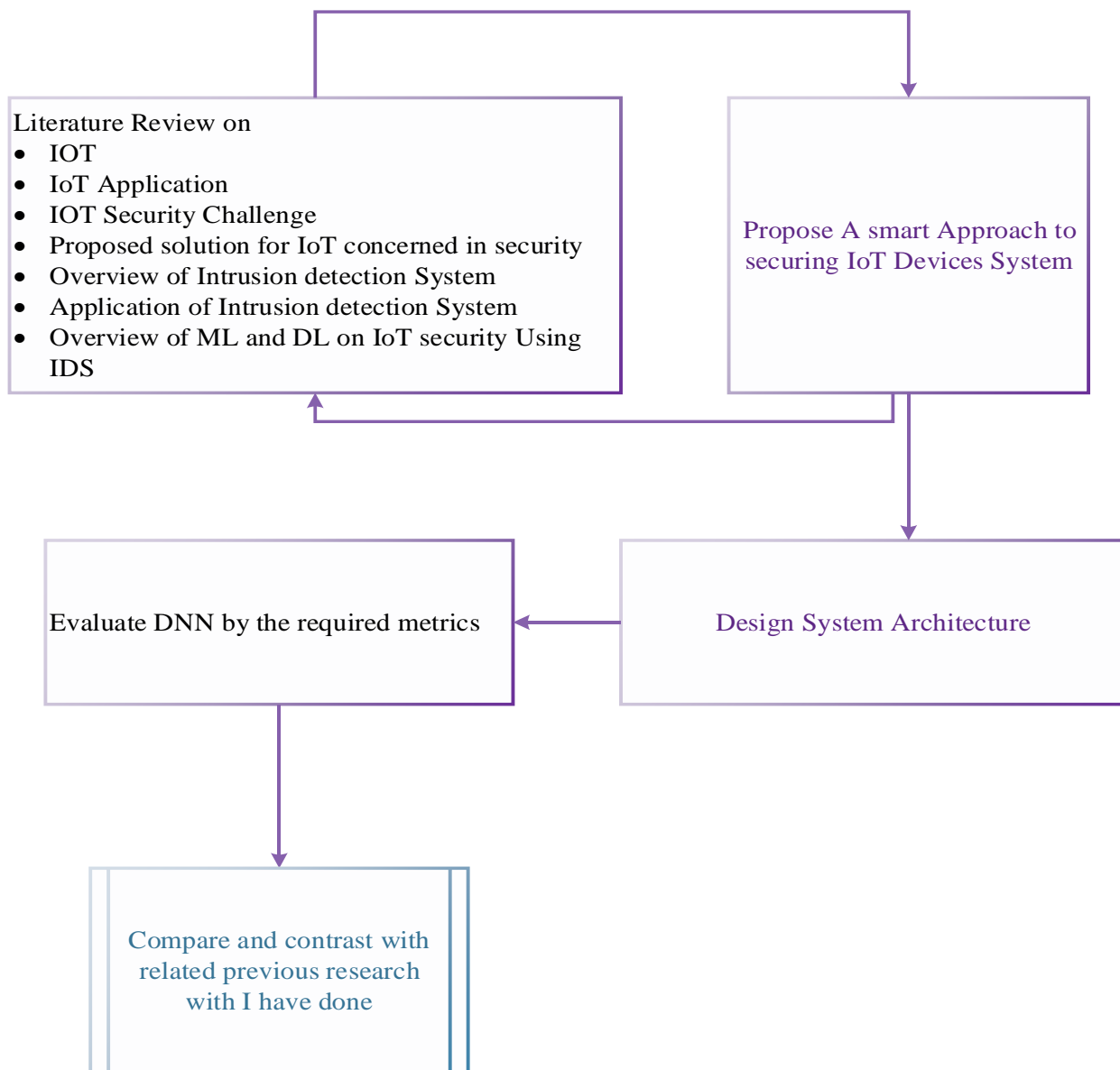


Fig 6: Research Methodology

1.6 Scope and Limitation

The scope of the study is focused on solving cybersecurity in the Internet of Things by developing anomaly-based intrusion detection systems using deep neural network methods. The specific scope

of this study preprocessing the dataset, training the dataset using deep neural network algorithms, testing the model by using the testing dataset, evaluating the model, design a new system architecture of the model. The limitation of this study is limited on intrusion detection system for the Internet of things that means it can't take an action like an intrusion prevention system to block the packet however, it alerts an alarm when it detects Dos attack.

1.7 Significance of the Study

The significance of this study is to secure the internet of things from cyber-attack during communication, data sharing, and service providing for the users in different sectors.

1.8 Document Outline

This paper is organized into the following chapters: - chapter one is about problem identification, the goal of the study, the Research Methodology, and the Scope of the study. Chapter 2, it is discussed the literature Review of IoT, Machine Learning, Deep Learning. Chapter 3, it is discussed related work. In chapter 4, it describes the proposed solution or methodology. When we see chapter 5, describes the Implementation and Performance Evaluation. In chapter 6, it is about the conclusion and Future Work.

Chapter 2: Literature Review

2.1 The Internet of Things: An Overview

The Internet of Things was conceived in 1999 by a member of the Radio Frequency Identification development group, and it has only lately been increasingly applicable to the real world, owing to the advent of mobile devices, embedded and ubiquitous networking, cloud computing, and data analytics. The Internet of Things refers to a type of network based on stipulated protocols to connect something to the Internet via information sensing equipment to share and communicate information to achieve smart identification, positioning, tracking, monitoring, and administration. The common concept of the Internet of Things is known as: The Internet of Things (IoT) is a physical entity network. The internet is not only a computer network, but it has grown into a network of devices of all kinds and sizes, cars, smartphones, home appliances, toys, cameras, medical devices and industrial systems, animals, people, and homes are all connected, interacting, and exchanging information according to pre-determined protocols to carry out wisely reorganizations, location, tracing, and control.[23].

In the world of computer networks, the Internet of Things (IoT) is an evolving concept that enables communication through the Internet between all kinds of objects. RFID tags, sensors, actuators, cell phones, etc. may be such objects; they use a single addressing mechanism to communicate and collaborate to achieve a shared purpose. The IoT enables the development of universal computing to integrate all sorts of communications, all the time, for everyone, and on any entity. It will cover a wide variety of applications and touch on almost all aspects of everyday life that we face[10]. Latest advances in communications and information technology, such as the Internet of Things (IoT), have far exceeded the conventional understanding of the surrounding world. The production of systems that can enhance the quality of life has been encouraged by IoT technologies[11]. The Internet of Things (IoT) is a network system that connects real-world items with unique identities, such as computing equipment, animals, digital machines, objects, animals, and people, so that such objects can transfer data over a network without involving humans or computers. [24].

2.2 Characteristics of Internet of Things

With a variety of functions, the IoT is a complex device. Its features differ from one domain to another. Some of the features of the general and main are the following[23], [25].

2.2.1 Interconnectivity

Concerning the IoT, the global knowledge and communication system can be interconnected with everything.

2.2.2 Things-Related Services

Under the constraints of things, the IoT is capable of offering thing-related services, such as private security and semantic continuity between physical things and their related virtual things. Both the technology in the real world and the information world can evolve to include things-related services under the constraints of things.

2.2.3 Heterogeneity

IoT devices are heterogeneous, as they are built on several hardware platforms and networks. Via various networks, they may communicate with other devices or service platforms.

2.2.4 Dynamic Changes

Device context includes position and speed, as well as dynamic changes in device state, such as sleeping and waking up, connected and/or unplugged. In addition, the number of devices can change dynamically.

2.2.5 Enormous Scale

As the number of the device connected to the internet on the internet of Things is scaled up from time to time, it needs to be handled and enabled to communicate each other.

2.2.6 Safety

We must not forget about protection as we reap benefits from the IoT. We must design for protection as both the designers and receivers of the IoT. This requires the protection of our personal information and the security of our physical well-being. It entails developing a security paradigm that can scale to secure endpoints, networks, and the data that flows between them.

2.2.7 Connectivity

Connectivity ensures usability and compatibility with networks. Accessibility is accessed on a network, while compatibility offers the common capacity to absorb and generate information.

2.2.8 Overall Perception

In the different sensing systems, the Internet of Things is widely applied. Whenever and wherever possible, the sensor obtains real-time data and continuously updates data according to the periodic collection of environmental information using the two-dimensional code method RFID sensor to obtain material information.

2.2.9 Reliable Transmission

The Internet of Things is a sort of ubiquitous internet-based network, an essential cornerstone, and the Internet of Things' core technology is still the Internet. The Internet of Things will relay the data of subjects to the data center in real-time, securely, and reliably.

2.2.10 Intelligent processing

Not only does the Internet of Things provide the sensor with a network link, but it can also do intelligent processing, intelligent control of things. The Internet of Things integrates sensors with intelligent processing, i.e. the use of cloud computing, fuzzy recognition, and smart computing technologies, vast data collected for analysis and processing in the data center, to obtain meaningful data, to provide feedback and basis for the implementation of intelligent decision-making.

2.3 Application of Internet of Things

Let's turn our attention now to various types of application scenarios that can benefit from the revolution of IoT. Typical and future applications, such as healthcare (e.g. patient management and surgery), smart energy, smart cars (e.g. autonomous vehicles), industrial automation, etc. [26]. The following are some of the applications of the Internet of Things[27].

2.3.1 Smart Home

Smart Homes would become as popular as smartphones as a result of the advantages they bring, such as energy conservation, cost savings, and decreased time spent performing tasks. Remotely

regulated air conditioning, heating, lighting, and other household equipment are only a few of the features available in a smart home.

2.3.2 Smart Grid

Another common IoT application in the world is the Smart Grid. Data is automatically collected and analyzed to better understand user behavior and energy consumption trends to increase the device performance and gain insight into the economics of electricity usage.

2.3.3 Industrial Internet of Things

The use of IoT in manufacturing, logistics, transportation, utilities, mines and metals, aviation, and other industrial sectors is known as the Industrial Internet of Things (IIoT). Industrial IoT is a transformative strategy for manufacturing that leads to improved quality, safety, and productivity. IIoT applications include the sharing of stock information between manufacturers and retailers in real-time, monitoring products and automatic distribution, etc. For quality management and sustainability, it also has a great reach.

2.3.4 Smart Retail

In the fields of supply chain management and logistics, the Internet of Things is important in retail. It aids in the enhancement of customer service through digital signage, self-checkout, and smart mirrors, among other items. With the interoperable components given by IoT, one can easily scale as the business changes. By sensing, knowing, and acting on IoT data with analytics, retailers aim to have a frictionless customer experience.

2.3.5 Smart Healthcare

Know that the Internet of Things is making significant development in the healthcare sector and that it fulfills responsibilities in this industry. Medical professionals may also use IoT to gain quick access to patient information that has already been processed on the internet. It helps to provide medical services to patients in remote areas where doctors are unable to access them physically, and also helps senior citizens as they are limited owing to their mobility. IoT-based healthcare creates an unprecedented opportunity to increase the quality and efficacy of medical services, leading to better patient health.

2.3.6 Smart City

The Smart city model is based on a centralized architecture in which numerous peripheral devices are interconnected across the city and produce and collect data, which is then transmitted through a variety of communication technologies to a centrally controlled hub, where the necessary processing and analysis are performed, and a decision is made about a specific application. The smart city seeks to maximize the use of existing resources, a convenient living atmosphere, and a wide range of customer services.

The components of a smart city are listed below.

2.3.6.1 Smart Homes

Because of its multiple benefits, such as productivity and cost control, Smart Homes will become the most common IoT application. Anything in a smart home can be controlled remotely using a smartphone and voice commands. Remotely controlled air conditioning, heating, ventilation, control of other household appliances such as refrigerators, washing machines, dishwashers, etc. are some of the features available in a smart home. Household appliance management helps to better maintain and operate the home efficiently, along with saving resources. When the room is not in use, the occupancy sensor changes the temperature and turns off the lights. Water management is provided by an automated sprinkler control device that interacts in real-time with local weather data. Some of the smart homes' other advanced features are addressed as follows.

- Smart Bedroom: smart beds use 'responsive air' technology that detects movement at night and automatically adjusts firmness, bed comfort. Smart pillows use sleep monitoring devices to gather data during the night to provide you with sleep cycles, snoring, and restfulness statistics that can be analyzed by a mobile application.
- Smart Kitchen: Coffee in a Smart Kitchen Smart Optimum Brew lets you set regular brewing schedules and warns you when it's time to refill with coffee or water. The Thermos Hydration Bottle, another smart kitchen gadget, comes with a sensor tube on its Smart Lid that monitors your daytime water intake while offering real-time temperature measurements. It also connects via Bluetooth to the smartphone to calculate your proposed hydration goal based on personal data.
- Smart Bathroom: Wi-Fi connects to the smart shower and allows you to adjust your ideal water temperature and shower length remotely, saving time and water.

2.3.6.2 Smart Transportation and Traffic

In cities, traffic congestion is a normal phenomenon. In this question, smart traffic signals come to the rescue. Real-time traffic data can be collected using GPS-equipped cars, road sensors, and traffic cameras, advising the driver to take a less congested route, saving fuel and time. Smart Parking uses sensors and smartphones that provide real-time data that enables a person to know the parking slots in advance. It directs the driver to a suitable parking space, optimizes the parking space area, makes parking hassle-free, and contributes to a greener environment by reducing CO2 emissions, resulting in a smooth driving experience.

2.3.6.3 Smart Water Management

For our life, water is very important. Water conservation, water quality, and public health all benefit from smart water management. Using the IoT-based monitoring system, safe and routine supply and maintenance of water quality are carried out. Sensors are used to calculate water level and flow in real-time, and a full water management system is developed. Because of inefficient systems, smart water meters detect leakage, provide data integrity and minimize revenue loss. Consumers can also have gates where they can have access to their water intake in real-time.

2.3.6.4 Smart Waste Disposal

The collection and disposal of waste is an important service needed in every region. Smart garbage bins have alarms mounted and transmit details to the appropriate authority when the bin is full—all of this is possible thanks to the Internet of Things. Smart waste management results in the proper use of manpower, reduction of fleet costs, and reduction of greenhouse gas emissions, increased cleanliness, and environment-free pollution.

2.3.6.5 Smart Energy Usage

Conservation of energy is very critical as the resources that provide energy are rapidly depleting. The IoT applications in the energy sector range from basic temperature control in a room to offering full energy-saving solutions for an entire building at a higher level. Energy savings, decreased downtime, improved energy efficiency, lower power bills, and increased convenience and comfort for customers are all advantages of real-time energy use, smart meters, and smart grids.

2.3.6.6 Smart Surveillance

Security and surveillance will help society resolve the threats raised by terrorism and other illegal activities. There is a possibility of data being stolen as data is openly exchanged over the internet. The Internet of Things (IoT) plays an important role in data protection. Motion detection cameras, alarm burglars, CCTV cameras are all IoT activated to send the customer information about any mishap. Smart surveillance systems based on IoT will make homes, towns, and businesses safer.

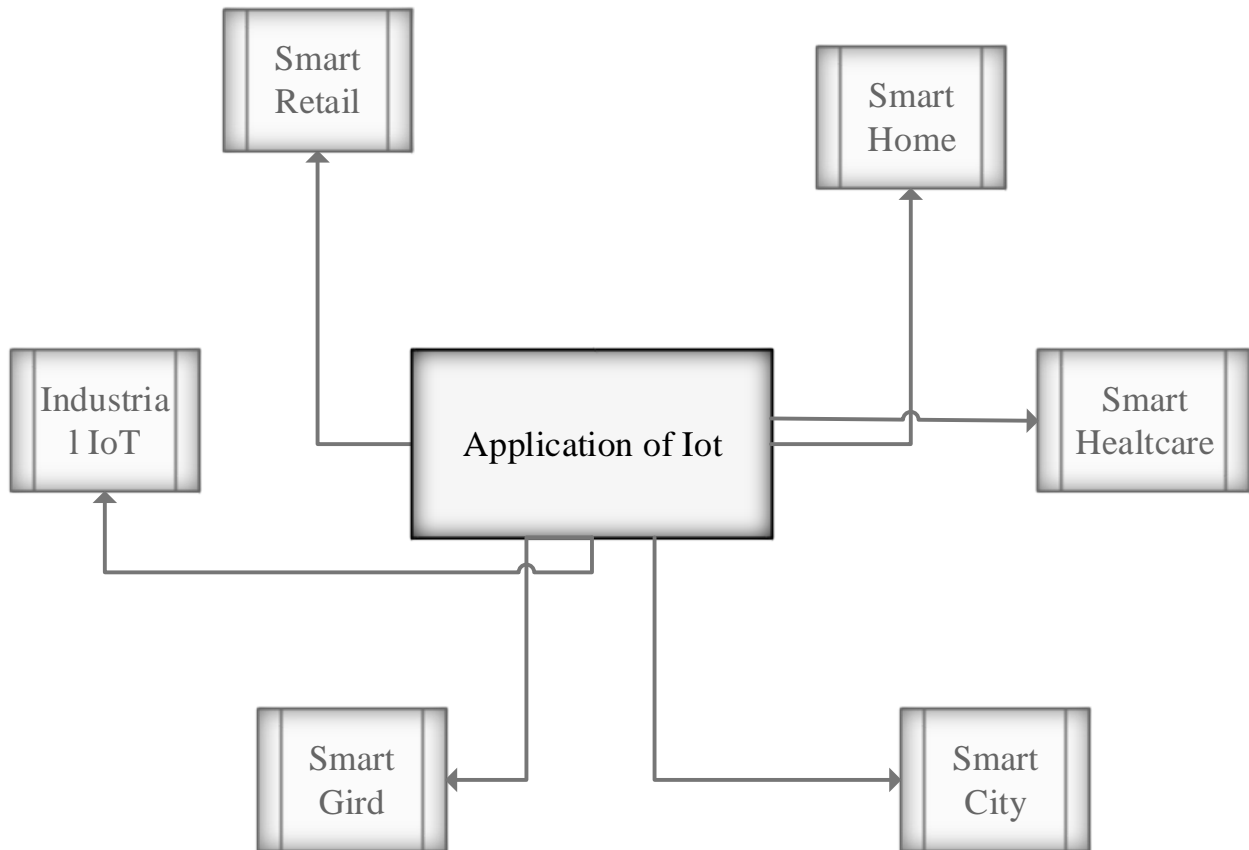


Fig 7: Application of IoT

2.4 Security Challenges in Internet of Things

A variety of obstacles prevent IoT devices from being secured and end-to-end protection in an IoT environment. Since networking equipment and other artifacts are still a relatively new concept, protection hasn't always been a top priority during the design process. Furthermore, because the Internet of Things is still in its infancy, many product designers and manufacturers are more concerned with getting their products to market as quickly as possible rather than taking the

necessary precautions to ensure security from the start. A key issue with IoT security is the usage of hardcoded or default passwords, which can lead to security breaches. Even if passwords are updated, they are often insufficient to avoid unauthorized access. Another problem with IoT devices is that they are often resource-limited and lack the computing resources needed to enforce strong protection. As a result, many devices lack or are unable to have advanced security features[12].

Since IoT devices are often deployed in hostile and unstable environments, the crucial data shared between them are more vulnerable to attacks. As a result, security solutions are needed to protect IoT devices from attacks by intruders[28].

2.4.1 Vulnerabilities in Internet of Things

IoT devices are vulnerable due to a lack of built-in protection to defend against attacks. Users, in addition to technological factors, play a role in the devices' susceptibility to attacks. The following are some of the reasons why these smart devices are still at risk[29][30]:

- Limited computational abilities and hardware limitations: These devices perform complex tasks that necessitate minimal computing capacities, leaving little space for robust security and data protection mechanisms.
- Heterogeneous transmission technology: A variety of transmission technologies are commonly used by devices. It may be difficult to develop uniform security methods and protocols as a result of this.
- Components of the device are vulnerable: Millions of smart devices are impaired by insecure basic components.
- Users with insufficient security awareness: Due to a lack of user security awareness, smart devices may be vulnerable to assaults.
- Insecure Internet of Things Network Interface: The majority of Internet of Things (IoT) network users rely too heavily on default passwords, poor passwords, or password recovery features that are too easy. Cross-site scripting, cross-site request forgery, and SQL injection are some of the other features that contribute to an unstable web interface.
- Insufficient Authentication/Authorization: Anyone who uses the web interface, whether it's a human or a bot, is putting themselves at risk. Unauthorized users accessing the network interface is one of the most well-known security risks. Authentication and

authorization must be dramatically enhanced to properly secure information on the internet of things network. When required, this authentication and authorization should be revoked. It's crucial to make sure that unique authentication tokens or session keys are used for application, system, and server authentication.

- **Insecure Internet of Things Network Services:** All open ports on network devices must be assessed and validated as not vulnerable to cyber-attacks to protect internet of things network services.
- **Lack of Transport Encryption/Integrity Verification:** To mitigate this risk, network traffic, mobile applications, and other connections should not send any explicit text over the transport layer. To do so, encryption protocols and protected sockets must be in operation. The secure socket layer should be used to secure and establish encryption, and the transport layer security should be upgraded and adjusted.
- **Insecure Cloud Interface:** To protect the cloud interface, network users should avoid using default usernames and passwords, block user accounts that are unable to log in after a predetermined number of attempts, and search all cloud interfaces for other vulnerabilities.
- **Insecure Mobile Interface:** Two-factor authentication protection concept can be used to ensure improved security for mobile interfaces to avoid data leakage when linked to wireless networks.
- **Inadequate security setup:** For easy intrusion detection, security configuration should be enhanced utilizing a variety of ways, such as allowing different degrees of access and privileges to different users, encryption, the use of a very strong password, and the recording of different security events.
- **Insecure Software/Firmware:** One of the first goals should be to ensure that the IoT system can be modified and that the update files are encrypted and sent over an encrypted link. This update must be signed and checked, as well as the update server.
- **Poor Physical Security:** Physical protection can also make an IoT network more vulnerable. The storage medium should be protected against easy removal, stored data should be secured, bad actors should not be able to gain access to the ports, and the system should not be easily disassembled.

2.4.2 Counter Measurement for IoT security Challenges

To protect the IoT system, it needs to take security measurements. The security counter measurement can be discussed in the three types of layer architecture [31].

2.4.2.1 Security Mechanisms for Perception layer

The following techniques are used to defeat the perception layer from malicious attacks.

- Hashed Based Encryption:- Hashed-based encryption security is a type of encryption that converts a message into an anonymous form known as ciphertext. When a message is transmitted from a sender, it is changed into a different format using a key that only authentic users can decipher. The length of the message is used to produce a key. It always has a key that is twice as long as the message. As a result, breaking a key is a difficult task. The receiver receives the key as well. The key can be used by the receiver to turn the ciphertext into an original message.
- Public Key Infrastructure protocol:- A Public Key Infrastructure (PKI)-like protocol mechanism combines all of the methods including authorization, authentication, and intrusion detection, and is implemented at the recognition layer of IoT architecture. It's preferable to employing various mechanisms one by one. A network is formed by many nodes that are connected. It must keep the community safe. As a result, it has no confidence in anyone sending a message. The public key and private key are both encrypted with the RSA encryption method. A base station stores the public key and distributes the private key to each node.
- Lightweight cryptography: Lightweight cryptography is a subclass of cryptography that tries to provide secure solutions for low-resource devices, such as IoT devices. The terms symmetric key lightweight cryptographic algorithm, public key lightweight cryptographic algorithm, and hash functions are used to describe three different forms of lightweight cryptography methods. While symmetric-key encryption is appropriate for communication security, one lightweight symmetric-key encryption strategy has been devised that has proven to be particularly effective in securely transferring data in IoT networks.

2.4.2.2 Security mechanisms for Network Layer

Network layer architecture is used for transmitting and forwarding data collected from the perception layer. To safely transmit the packet to the destination recipient the following countermeasures should be considered.

- Identity management framework:- For each device to communicate with each other, authentication is required. It allows users to verify the validity of devices before transferring data. The identity management framework is a technique presented to meet this need. The environment, sensors, and receiver, as well as the network, make up the system. The environment refers to the area where sensors and things are linked. The sensors are in charge of gathering data from objects. It delivers all of the data to computers, which make the final decision. After using computers to make a choice, the receiver receives the result. The network, which is utilized to convey information, is the final component. Sensors send information to computers, while computers send information to receivers. There are two parts to the proposed framework: identity manager and service manager. The identity manager verifies that the sensors and receivers have the necessary permissions to send and receive data. After receiving authentication permission from identity management, the service manager distributes services to devices.
- Software-Defined Networking with IoT:- Sensors send information to computers, while computers send information to receivers. There are two parts to the proposed framework: identity manager and service manager. The identity manager verifies that the sensors and receivers have the necessary permissions to send and receive data. After receiving authentication permission from identity management, the service manager distributes services to devices. SDN can keep track of network traffic and detect malicious activity. It locates the infected nodes and isolates them from the rest of the network. It was effective at detecting attacks and did not impose any overhead on the controller, but it was unable to diagnose other types of attacks.
- Cooperation of Node Communication protocols:- Security-conscious ad-hoc can protect against attacks from within the network; the goal of the proposed protocol is to identify nodes that can disrupt the entire network due to their bad conduct. The monitor, reputation system, path manager, and trust manager are the four components that make up the system. Nodes' information and suspicious behavior are communicated to a trust manager whenever suspicious conduct is observed from any nodes in the network. A trust manager sends out an ALERT message to all nodes in its range. The ALERT message contains information on the address of a reporter node, the attacker node's address, and packet loss. Through a reputation system, the node that gets an ALERT message examines whether the

reporter node is authenticated or not, as well as packet loss. If the information provided is correct, the path manager creates a new path from the source node to the destination node to ensure consistency.

2.4.2.3 Security mechanisms for Application Layer

To defend the application layer from malicious attacks, the following security techniques should be used.

- Special policies and permissions:- For accessing and operating the IoT infrastructure, specific policies and permissions must be followed. Policies are made up of a set of rules, such as eXtensible Access Control Markup Language (XACML) terminology. They are composed of a condition and an effect, for example, allow and deny. Outgoing and incoming traffic, as well as the system's access request, are all subject to access control lists, which can accept or refuse them.
- Anti-virus, anti-adware, and anti-spyware:- All of these tools can ensure the security, consistency, confidentiality, and reliability of the IoT region.
- Risk Assessment Techniques:- Techniques for assessing risk Risk assessment techniques detect risks to the IoT system, thus risk assessment should be used to secure the application layer. Update the firmware of the system devices to strengthen security measures in this case.

The above techniques are the crucial security counter measurement of the Internet of Things for security challenges. However, Attackers break down and penetrate these techniques to access the confidential identity of the Internet of Things. This leads to critical issues, that users can't access the IoT services like smart home, healthcare service, and Smart city. Another counter measurement is an intrusion detection system that controls and monitors the network traffic and alerts an alarm when the intruder is found. IoT systems consist of tiny objects with resource-constrained to support and install advanced security systems. The traditional intrusion detection system requires enough resources for installing. Therefore, applying this traditional intrusion detection system for the Internet of Things is not sufficient due to the lack of resource computing. To answer this limitation I have proposed an anomaly-based intrusion detection system in the Internet of Things using deep learning techniques. Because deep learning intrusion detection systems are lightweight and highly accurate to identify intruders.

2.5 Security Attack in the Internet of Things

Attacks are actions taken using different techniques and methods to damage a device or interrupt regular operations by leveraging vulnerabilities. Attackers initiate assaults, either for personal satisfaction or compensation, to achieve goals[32]. Inside and outside attacks are the most common forms of attacks. An outside attack is started by a network node that is not a member of the network, whereas an inside assault is initiated by network nodes that are compromised or malevolent. Some examples of cyber-attacks in IoT applications are described below[33].

2.5.1 Wormhole Attack

The adversary node establishes a virtual tunnel between two ends during this attack. Between two actual nodes, a person node functions as a forwarding node. The two malicious nodes appear to be one hop away from the bottom station on occasion. By relaying packets between two distinct nodes, the wormhole attack can also be used to persuade two distinct nodes that they are neighbors.

2.5.2 Sinkhole Attack

During the attack, the malicious node draws network traffic to itself. To carry out these attacks, a malicious node entices all nearby nodes to forward their packets through it by displaying its lowest routing cost. An attack is launched by inserting a false node into a network.

2.5.3 Selective Forwarding Attack

In this attack, the malicious node appears to be a regular node but drops some packets selectively. The simplest form of selective forwarding attack is the black hole attack, in which the malicious node bears all packets.

2.5.4 Sybil Attack

The node has several identities during this attack. A malicious node will compromise the routing protocol, detection algorithm, and cooperation processes.

2.5.5 Hello Flood Attack

The routing protocol broadcasts a hello message to its neighbors in a very sensing element network to announce its presence. When a node receives a hello message, it will assume that the source node is within its contact range and add it to its list of neighbors.

2.5.6 Denial of Service (DOS) Attack

The availability of services will be harmed as a result of this assault. When this attack is launched, legitimate users will be unable to access services. DDoS attacks are those that are initiated by

several malicious nodes. This attack can have an effect on network resources, bandwidth, and CPU time, among other things.

2.6 Requirement of Security

In the Internet of Things, all devices and people are linked together to provide services at any time and in any place. Most internet-connected devices lack effective protection protocols, making them vulnerable to various privacy and security concerns such as confidentiality, integrity, and authenticity, among others. Some protection specifications for the IoT must be met to protect the network from malicious attacks. Some of the most important features of a secure network are briefly discussed here[34], [35].

- Confidentiality: An attacker can easily intercept a message sent from the sender to the receiver, exposing personal information and modifying the content. As a result, secure message transmission is essential in the Internet of Things.
- Integrity: The message must not be tampered with in transit; it must arrive at the receiver node in the same condition as it left the sender node. Integrity ensures that the message has not been tampered with while in transit.
- Availability: When data or services are needed, they must be accessible. Attackers can saturate a resource's bandwidth to degrade its availability. Malicious attacks such as denial of service (DoS), flooding, black hole attacks, jamming attacks, and others can compromise availability.
- Resilience to attacks: If the device crashes during data transmission, it should be able to recover. A server in a multiuser environment, for example, must be intelligent and strong enough to defend itself against intruders or eavesdroppers. If it goes down, it will automatically restore without informing users.
- Data Authentication: Authentication of the data and associated information is needed. Only authentic devices are allowed to transmit data using an authentication mechanism.
- Access control: Access control is granted only to those who have been given permission. The system administrator must manage users' usernames and passwords, as well as define their access privileges so that various users can only access the database or programs that are important to them.

- Client privacy: The information and data should be kept secure. To protect the privacy of clients, personal data can only be accessed by approved personnel. It means that no unauthenticated device user or another kind of client can access the client's private information.
- Authenticity: The ability to confirm one's identity is referred to as "authenticity."
- Users should be able to recognize the identity of the individual with whom they are communicating. It can be checked via the authentication process, ensuring that an unauthorized party is prevented from participating in the correspondence.
- Non-Repudiation: Non-repudiation means that neither the sender nor the recipient can deny sending or receiving the message.
- Data Freshness: When data is needed, it must be current. It ensures that an adversary can not be able to replay previous messages.

2.7 IoT Architecture

The Internet of Things (IoT) is a network of interconnected objects that allows people and objects to communicate and build smart environments in areas such as transportation, agriculture, healthcare, electricity, and towns. The IoT architectural model is made up of three layers[10].

- Perception layer: This layer contains devices that detect and collect data from the environment, which it then sends to the network layer.
- Network layer: Data transmission is carried out at the network layer using some of the most recent technologies, such as WiFi, Bluetooth, 3G, Zigbee, and so on. Devices are linked to the cloud internet through an Internet of Things gateway.
- Application layer: This layer includes the services that consumers need, such as services for smart homes, health care, and so on. The data's authenticity, honesty, and confidentiality are all ensured at this layer.

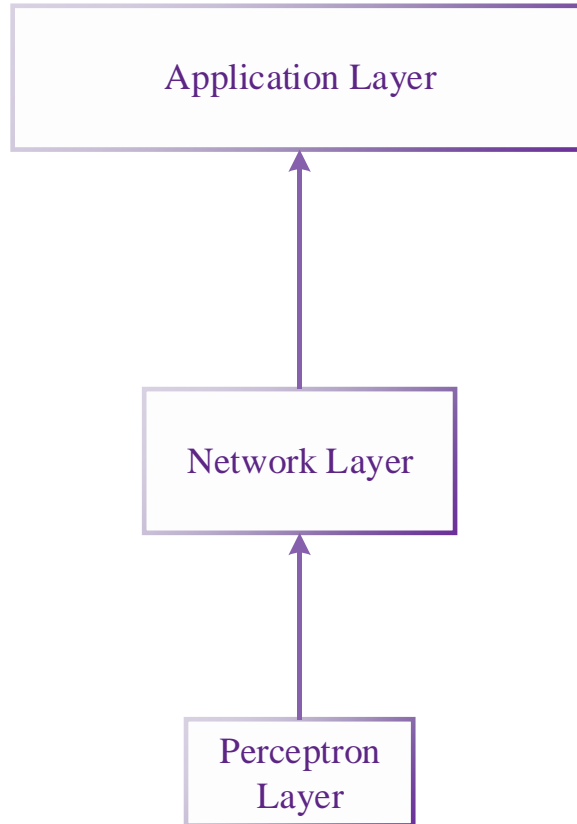


Fig 8: IoT Architecture

2.8 Introduction to Intrusion Detection System

Jim Anderson first proposed the idea of keeping track of user activity through logs and computer records in 1980. It was to prevent unauthorized external or internal users from accessing information, as well as misfeatures, or users who abused their power. IDDES, developed by SRI International's Computer Science Laboratory, is the world's first real-time intrusion detection system. IDDES is a stand-alone real-time intrusion detection system that uses mathematical algorithms for anomaly detection and an expert system for rule-based detection. The IDDES model, which was introduced in 1986, was intended to be general-purpose, and the framework could be used to construct a much more robust and efficient IDS. Because of the high demand for IDS technology in the industry, it has become a very popular and well-researched topic. IDS' performance and accuracy have improved as a result of the unwavering interest in this technology. Since the formation of IDDES, the industry has seen a significant increase in the number of intrusion detection devices available for consumers to choose from and introduce. Kevin Richards evaluated the efficiency of five different IDS products in a manufacturing environment in 1999[36].

An Intrusion Detection System (IDS) analyzes network and system behavior to detect intrusions or attacks against them. The Intrusion Detection System (IDS) is used to keep track of malicious traffic in a node or network. It can protect the network from intruders by acting as a second line of protection. Intrusion is a malicious or unwanted operation that harms sensor nodes. IDS is available in both software and hardware versions. IDS may examine and investigate machines and user behavior, detect well-known attack signatures, and recognize malicious network activity. The aim of an intrusion detection system (IDS) is to monitor networks and nodes, detect various types of network intrusions, and notify users when intrusions are discovered. The IDS functions as an alarm or network observer, preventing device harm by issuing a warning before the attackers launch their attack. Intrusion detection systems can identify threats from the inside as well as the outside. Internal attacks are initiated by malicious or compromised network nodes, while external attacks are initiated by third parties. IDS examines network packets to see if they are from intruders or legal users[13], [35].

When it senses a security breach, the IDS tracks the activities of a host or network and alerts the system administrator. The IDS is made up of three main sections[37].

- **Monitoring:** This component is primarily responsible for tracking traffic patterns, internal events, and resource use.
- **Analysis and detection:** This is the key component that uses an algorithm to detect intrusions.
- **Alarm:** When an intrusion occurs, this part produces an alert.

IDS is responsible for detecting malicious activity by tracking network environments and processes, and it involves both software and hardware mechanisms. To put it another way, an intrusion detection system (IDS) detects cyber-attacks, and issues are found it alerts an alarm. IDS serves as a protection for networks and applications in general. In most cases, an intrusion detection system (IDS) is deployed after the firewall and utilized in tandem with an intrusion prevention system. IDS is not a new concept in the field of IoT protection and privacy research. In recent years, a large number of publications have appeared. For quite some time, cybersecurity experts have been worried about the security and privacy of IoT environments. To combat cyber-attacks, the idea of IDS embedding has been introduced into IoT architectures and devices. Researchers are mainly involved in inventing new techniques and models to counter intruders in traditional

network protocols. Traditional IDS mechanisms, on the other hand, are incompatible with IoT devices that are connected via IPv6 and other complicated network structures. Additional information For IDS to secure and protect privacy in IoT, research on the use of machine learning methods is required[38].

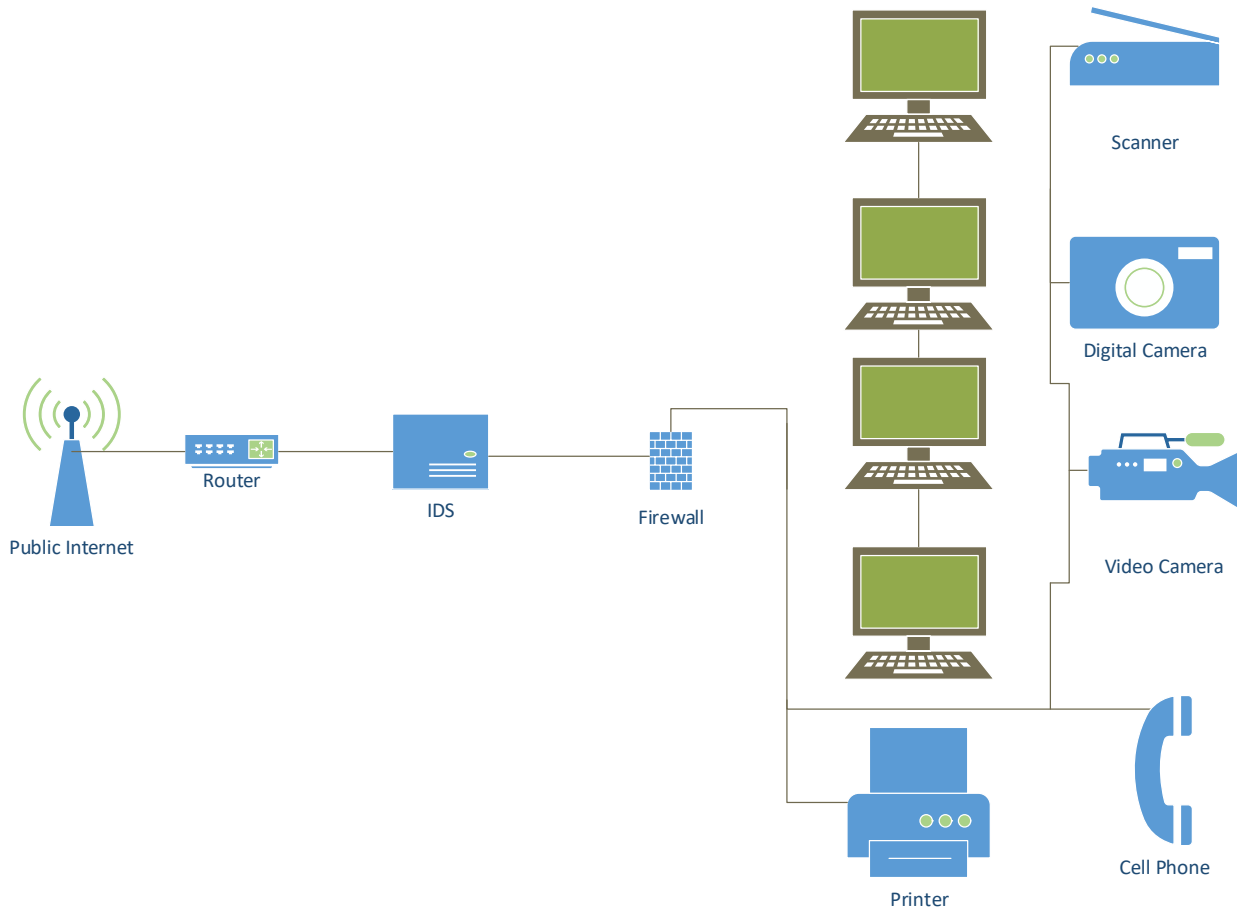


Fig 9: IDS general deployment

2.8.1 Classification of Intrusion Detection System

IDS come in a variety of forms, ranging from antivirus applications to hierarchical structures that commonly control entire backbone networks are traffic. The following the most classifications [39], [40].

- Network-Based (Network IDS): Network-based intrusion detection tries to detect unauthorized, illegal, and anomalous activities based only on network traffic. A network

intrusion detection system (IDS) gathers packets that pass via a network tap, bridge port, or hub. The IDS system processes and flags any unusual traffic based on the collected data. The major distinction between an intrusion prevention system and an intrusion detection system is that the IDS is unable to limit network traffic. A network IDS' function is passive, consisting solely of collecting, identifying, recording, and alerting.

SNORT is an example of a Network IDS.

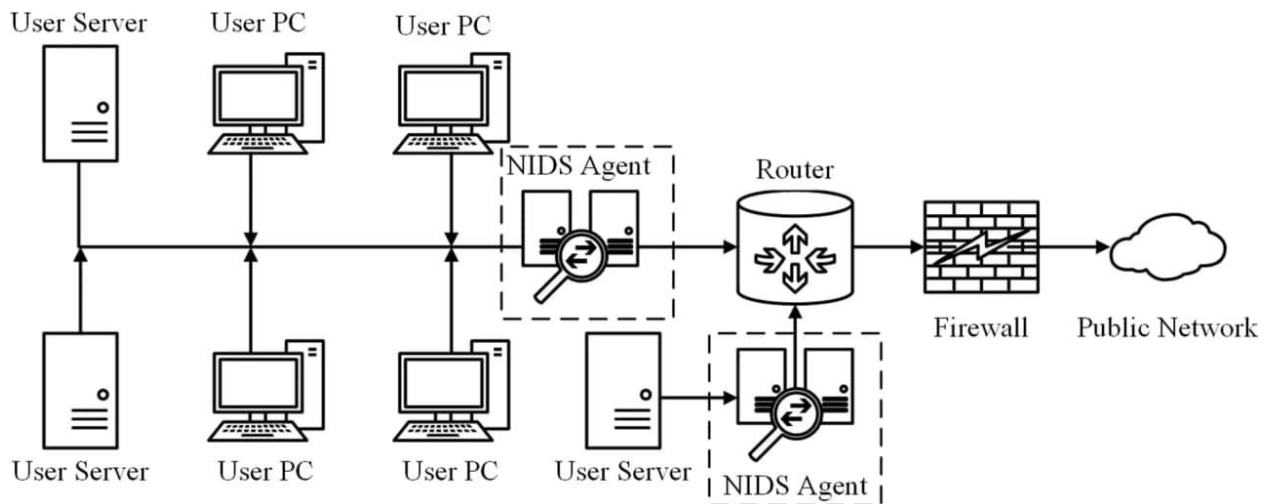


Fig 10:Network-Based (Network IDS).[41]

- **Host-Based (HIDS):** Host-based intrusion detection, also known as HIDS, tries to detect illegal, unlawful, or unusual activity on a particular computer. In most cases, HIDS includes installing an agent on each device that monitors and alerts on local OS and application operation. Using a combination of signatures, rules, and heuristics, the configured agent can detect unlawful conduct. A host IDS' function is largely passive, consisting of only gathering, identifying, recording, and alerting. HIDS includes tools like OSSEC (Open Source Host-based Intrusion Detection System), Tripwire, AIDE (Advanced Intrusion Detection Environment), and Prelude Hybrid IDS.

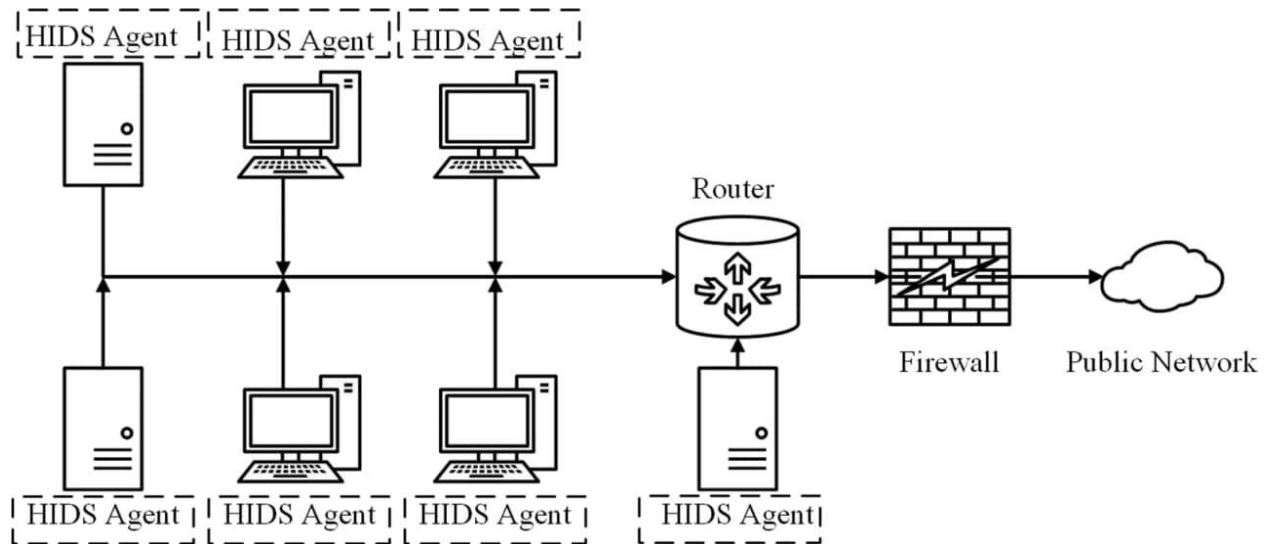


Fig 11: Host-Based (HIDS)[41].

- Physical (Physical IDS): The act of detecting threats to physical systems is known as physical intrusion detection. Physical intrusion detection is frequently viewed as a set of physical safeguards put in place to protect the CIA. Physical intrusion detection systems frequently serve as both detection and prevention systems. Guards, security cameras, access control systems (card, biometric), firewalls, man traps, and motion sensors are just a few examples of physical intrusion detection systems.
- Hybrid IDS: The administration and alerting from both network and host-based intrusion detection devices are the natural equivalent of NID and HID - central intrusion detection control. Network and host-based IDS each have their own set of benefits and drawbacks. Network-based IDS are easier to set up and maintain, as well as less expensive to buy and maintain. Their performance, however, is dependent on known security exploits and signatures. If the IDS is unaware of a new vulnerability, the device will be unable to detect the attack. The security administrator responsible for maintaining and monitoring a host-based IDS is only as good as the IDS itself. Learning how to use, maintain, and track this program can be difficult. The optimal method is to combine a combination of the greatest characteristics of Network and Host-based IDS to improve attack resistance and provide better flexibility. Hybrid IDS is a term used to describe this strategy.

2.8.2 Classification of IDS in Detection Techniques

Intrusion detection methods are divided into four groups based on the existence of different types of intrusion attacks: signature-based methods, specification-based methods, anomaly-based methods, and hybrid methods[42], [38].

2.8.2.1 Signature-Based Methods

Signature-based approaches begin by scanning the network data and comparing it to a function database. If the scanned data matches the signature database's features, the data would be classified as an intrusion. It has the advantage of accurately determining the type of attack. It's simple to use, and the demand for resources is low. Signature-based techniques identify attacks when the activity of a device or network matches an attack signature stored in the IDS internal databases. An alert will be triggered if any system or network activity matches the stored patterns/signatures. This method is reliable and very successful at detecting known threats, and its process is easy to understand. However, this method is ineffective to detect new attacks and variants of established attacks, since a matching signature for these attacks is still unknown

2.8.2.2 Specification Based Methods

System administrators must set rules and thresholds in advance using specification-based methods. IDS monitors the current state of the system and network based on the rules and thresholds set by administrators. If the threshold is exceeded or the rules are broken, the IDS will identify a problem and take appropriate action. The expected behavior of network components such as nodes, protocols, and routing tables is described by a collection of rules and thresholds known as specifications. When network activity deviates from specification, intrusions are detected using specification-based approaches. As a result, specification-based detection serves the same role as anomaly-based detection: detecting deviations from the standard. However, there is one major difference between these methods: in specification-based approaches, each specification's rules must be manually specified by a human expert. In contrast to anomaly-based detection, manual-defined requirements normally yield lower false-positive rates. Furthermore, specification-based detection systems do not need any training because they can begin operating immediately after the specifications have been set up. Manually specified requirements, on the other hand, do not conform to various environments and can be time-consuming and error-prone.

2.8.2.3 Anomaly-Based Methods

Anomaly-based approaches focus on detecting irregular patterns and comparing traffic patterns to come up with a solution. This approach has the benefit of allowing the identification of new and unknown intrusions. The system, on the other hand, has a high rate of false positives, which is a major drawback. To increase the robustness of anomaly-based intrusion detection approaches, researchers are now focusing on using machine learning algorithms. Anomaly-based intrusion detection methods use machine learning algorithms to track ongoing intrusion footprints and compare them to existing databases to identify possible future attacks.

2.8.2.4 Hybrid Methods

In the same IDS, hybrid methods refer to the use of any combination of the above-mentioned detection methods. This approach will help to resolve the limitations of a single process, thus improving the overall IoT system's reliability. The obvious downside is that the entire IDS can increase in size and complexity. This will make operating the system more complex and will necessitate more money. The intrusion detection method can consume a lot of resources and time, particularly if there are a lot of protocols in the IoT framework.

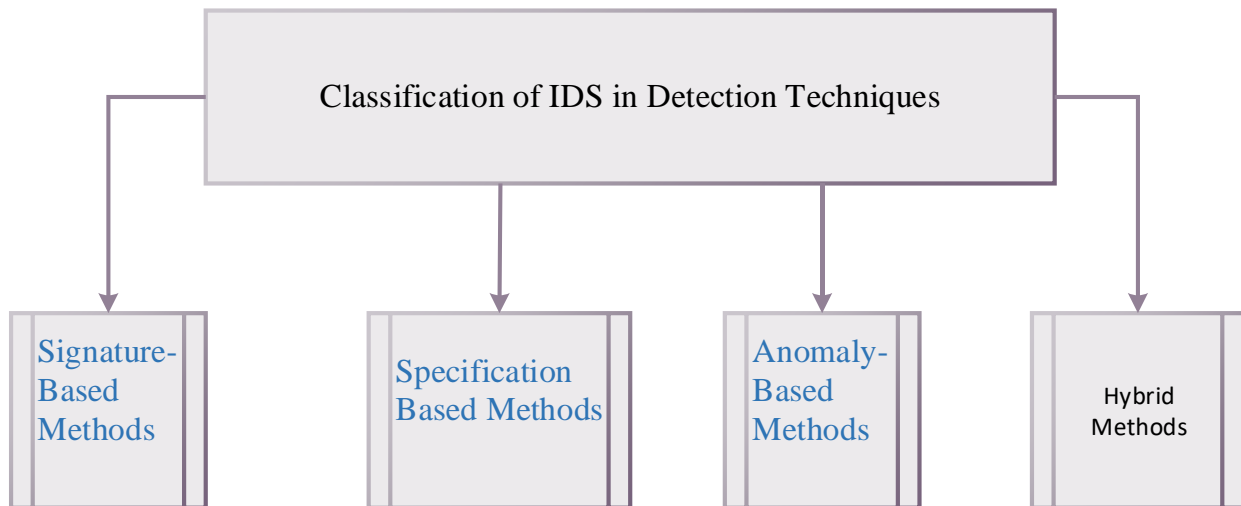


Fig 12: Classification of IDS Based On Detection Methods

2.8.3 Placement Strategy of Intrusion Detection System

IDSs can be classified as centralized, distributed, or hybrid, depending on placement strategies[43].

2.8.3.1 Centralized IDS for IoT

IDS typically deploys its agent in a master node with a lot of processing power and a lot of storage/memory space. This master node must also be able to track all network activities so that the deployed IDS agent can access real-time network activity data and identify intrusions by monitoring network activities in real-time. Since the IoT device naturally has an edge node (i.e. boundary router) that links the IoT network to the Internet, centralized IDSs are simple to implement. Network incursions by external attackers can be better detected by the centralized IDS agent since all outside packets must be forwarded to the edge node, and the attack can be neutralized by simply dropping the destructive packets sent by the malicious intruders at the edge node. Detecting intrusions from internal intruders, on the other hand, can be difficult for centralized IDSs because it allows the IDS agent to closely track the behavior of resource-constrained sensor nodes over lossy networks. In the meantime, even though border routers typically have a more powerful processor than the sensor nodes connected to them, computation cost, energy consumption, and memory use remain the most important factors to consider when IDSs are deployed in the IoT. In addition to their ability to detect external intrusions, centralized IDSs can also detect internal attacks like the selective forwarding attack, in which malicious nodes selectively forward packets to interrupt routing paths.

2.8.3.2 Distributed IDS for IoT

A distributed IDS, unlike a centralized system, places detection agents in each sensor node. Each agent watches and analyzes the behaviors of the nodes in its radio range, and sends out alerts if it notices anything unusual. The local agent's analysis, known as individualized decision making, or the majority votes of all neighboring agents, known as cooperative decision making, can be used to determine whether or not a node is compromised. Internal intrusion detection is a strong suit for distributed IDSs. Furthermore, the distributed IDS implementation does not rely on a super-powerful central node to detect possible attacks; rather, all sensor nodes with detection agents in the IoT will collaborate to detect possible attacks, making the system more resilient in the sense that incidents causing one node to malfunction would not harm the entire wireless IoT network. However, since the IDS agents are installed in each node, distributed IDS strategies are generally inefficient in terms of energy usage. This raises the overall computing cost and introduces additional coordination costs for cooperation. The designer of distributed IDSs for IoT must account for the fact that the detection agents will be deployed on sensor nodes with restricted

resources, memory, and processing capacity must pay particular attention to the energy consumption of each agent. Since deploying detection agents will impact the battery life of sensor nodes in the IoT, the financial cost should also be weighed.

2.8.3.3 Hybrid IDS for IoT

The hybrid placement strategy seeks a tradeoff between detection accuracy, algorithm complexity, energy consumption, memory and processing power use, communication expense, and other design requirements by combining the centralized and distributed strategies. IDS agents are often hosted on nodes that are more stable and efficient.

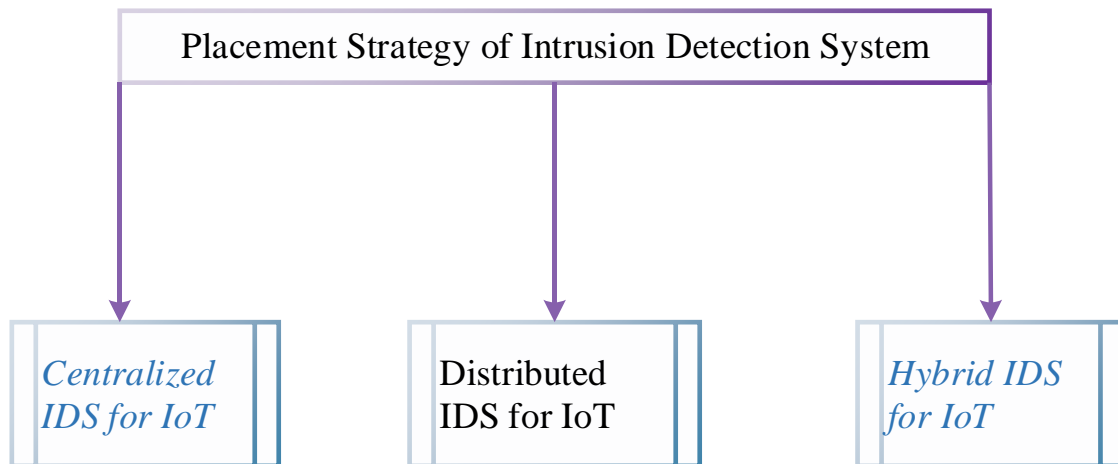


Fig 13: Placement Strategy of Intrusion Detection System

2.9 Deep Learning: An Overview

2.9.1 Deep Learning (DL) Techniques for IDSs

Deep Learning is a subset of machine learning or a specific type of machine learning. It works in the same manner that machine learning does in terms of technology, but with different capabilities and techniques. Artificial neural networks are based on the functionality of human brain cells known as neurons. When dealing with large datasets, DL algorithms outperform ML algorithms. Because IoT environments are characterized by the production of vast amounts and a variety of data, DL becomes especially relevant in IoT security applications. Furthermore, DL can automatically model complex feature sets from sample data. This allows IoT-based systems to interact automatically without the need for human intervention to perform collaborative tasks[44]. The ability of DL algorithms to enable deep linking in IoT networks is another benefit[45].

Deep Learning is a more advanced form of machine learning that uses many degrees of data abstraction and various processing layers to achieve multiple levels of data abstraction. Via back-propagation, Deep Learning can learn the complex structures in a dataset and show how the system adjusts the internal parameters at each layer. In terms of structure and learning data representations, deep learning, also known as hierarchical learning or deep structural learning, is a more structured form of machine learning. The main distinction between machine learning and deep learning is how the output changes as the size of the data grows. Deep Learning algorithms need more data to identify patterns in the network, while machine learning algorithms need fewer.[46].

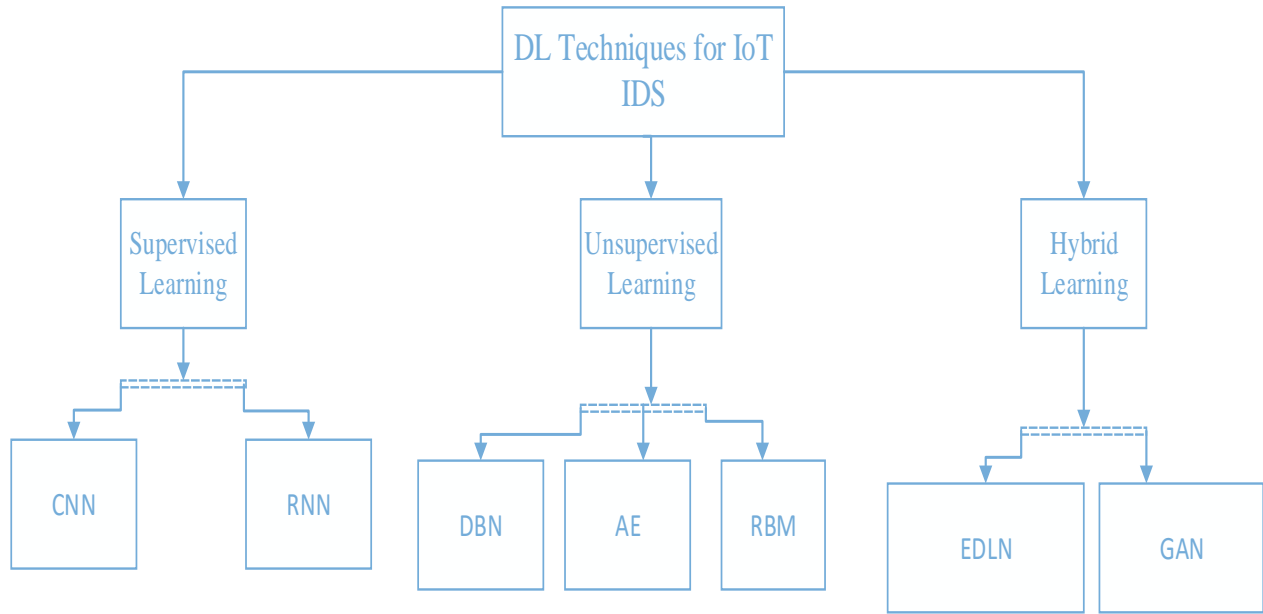


Fig 14: DL techniques for IoT IDS

2.9.1.1 Convolutional Neural Network (CNN)

CNN is a deep learning model for processing data with a grid pattern, such as photographs, that is based on the organization of the animal visual cortex and built to automatically and adaptively learn spatial hierarchies of information from low- to high-level patterns.[47]. As a result, CNN is more scalable and training takes less time. As shown in Figure 14, a CNN has three different types of layers: convolutional, pooling, and activation. Various kernels are used to convolute data inputs in the convolutional layers [185]. The pooling layers reduce sample sizes, allowing subsequent layers to be smaller. It entails two techniques: maximum pooling and average pooling, with the former being the more common. After distributing the input among distinct clusters, chooses a maximum value for each cluster of previous layers[48]–[50].

On the other hand, average pooling calculates the average values of all clusters in the previous layer. In a non-linear fashion, the activation unit can trigger an activation function on each feature in the feature set[49]. CNN is ideally suited for extracting highly effective and fast features from raw data, but it also takes a lot of computing power[51]. As a result, using CNN for security on resource-constrained IoT devices is extremely difficult. This problem is partially solved by a distributed architecture in which a lighter version of Deep NN is trained and implemented on-board with only a subset of critical output groups, while the cloud's high computing power is used to complete the algorithm's training[52].

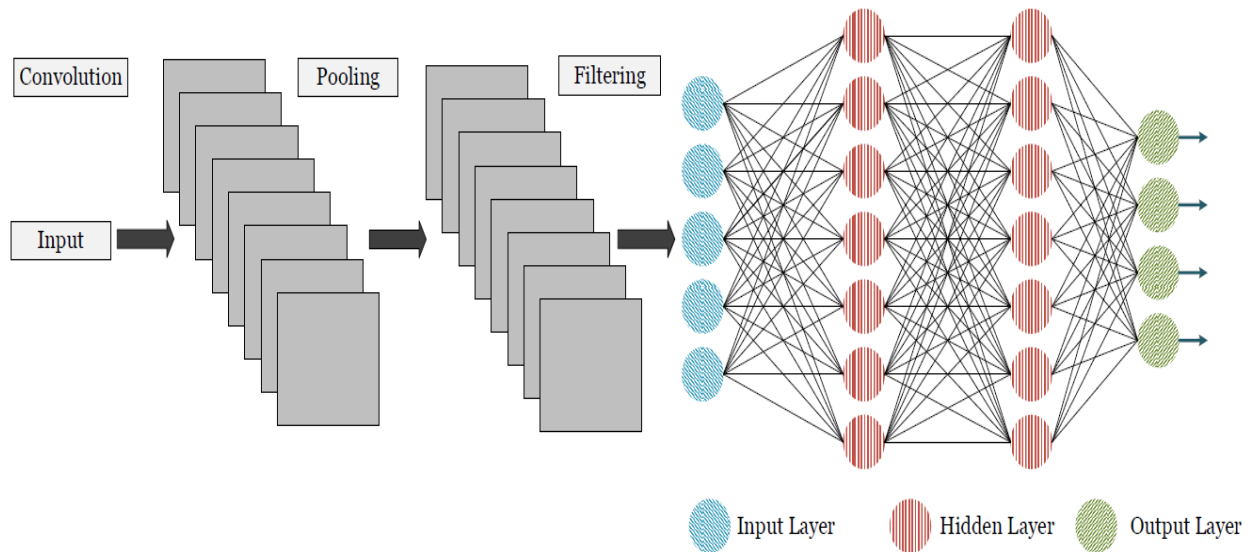


Fig 15: Illustration of convolution neural network working[11]

2.9.1.2 Recurrent Neural Networks (RNNs)

Recurrent neural networks are networks that analyze sequential input in natural language processing. Analyzing isolated data from a sequence makes little sense since the properties of sequential data are contextual. Each unit in an RNN receives the current state as well as previous states to obtain contextual information[53]–[55]. RNN is a discriminative deep learning algorithm that works best in situations where data must be processed sequentially. Its performance is based on back-propagation rather than forward propagation, as opposed to other neural networks[56]–[58].

2.9.1.3 Deep Autoencoders (AEs)

It is an unsupervised algorithm that uses a decoder function and a hidden layer that contains the concept of a code used for input representation to reproduce its input at its output[59]. The encoder function is the other function in an AE neural network, and it is responsible for converting the acquired input into code. Reconstruction errors must be minimized during training[60]. Feature extraction from datasets is one application for AE. However, they are limited by the need for a lot of computing power.

2.9.1.4 Restricted Boltzmann Machine (RBM)

It is a deep generative and undirected model that is built using an unsupervised learning-based algorithm[61]. In any layer of an RBM, there are no two nodes that are connected in any way. There are two types of layers in an RBM: visible and hidden layers. The visible layer contains known input parameters, whereas the hidden layer contains unknown potential variables. Using a hierarchical approach, features extracted from a dataset are passed on as latent variables to the next layer. For network/IoT intrusion detection systems, RBMs have been used in several studies[62], [63]. Implementing RBMs on low-powered IoT devices is difficult due to the high computational resources needed. Furthermore, Single RBM cannot represent features. However, by stacking two or more RBMs to create a Deep Belief Network, this limitation can be overcome (DBN).

2.9.1.5 Deep Belief Network (DBN)

DBNs are unsupervised learning-based generative algorithms, as they are formed by stacking two or more RBMs[64]. They perform well when each layer is trained unsupervised[65]. In the pre-training phase, each layer's initial features are extracted, then followed by a fine-tuning phase in which a softmax layer is applied to the top layer[66]. As shown in Figure 15, it is primarily made up of two layers: a visible layer and a hidden layer. Directed Graphs are the foundation of Deep Belief Networks. Individual units called Restricted Boltzmann Machines are used to build the stacks. The model is called a stochastic generative model. Deep belief networks are made up of two layers: one input layer and one hidden layer, as well as the network's specified energy and the probability of a unit's state[67].

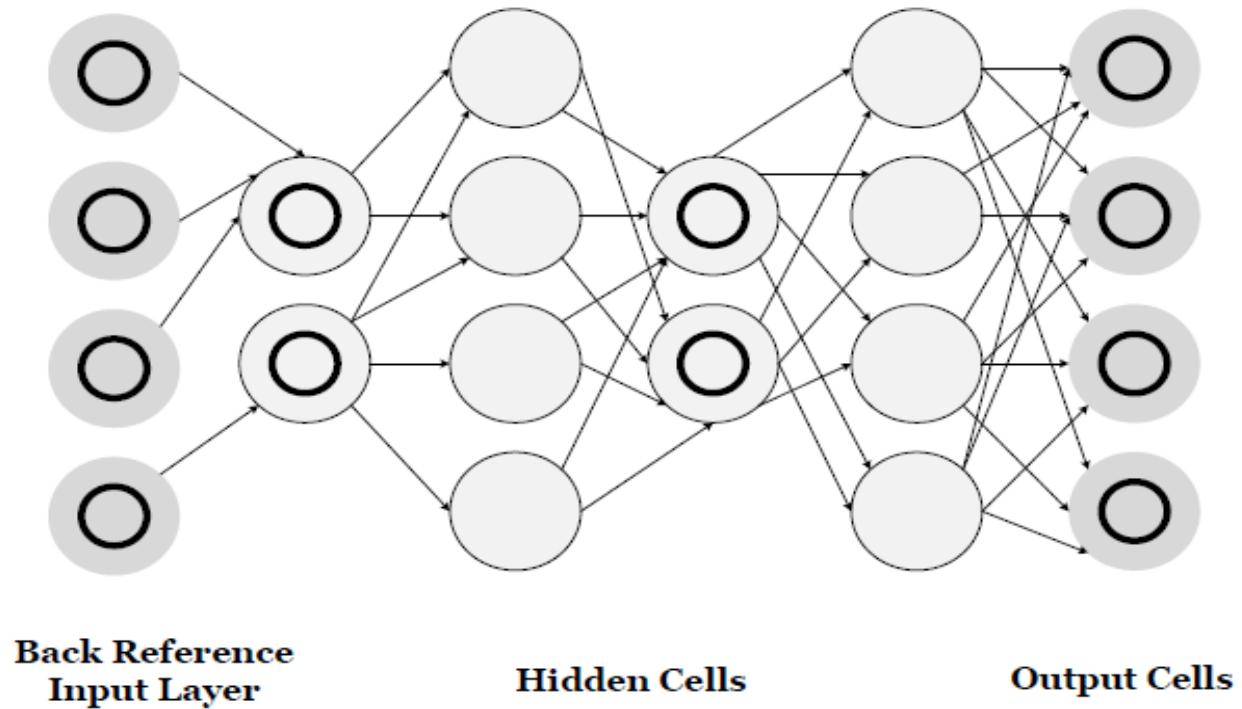


Fig 16: Illustration of deep belief network working[11]

2.9.1.6 Generative Adversarial Network (GAN)

It's a hybrid deep learning approach that trains with both generative and discriminative models. The generative model predicts the authentic origination of a given sample from a training dataset, and the discriminative model produces the dataset and sample distributions[68].

GANs, or Generative Adversarial Networks, are a type of generative modeling that employs deep learning techniques like convolutional neural networks. Generative modeling is an unsupervised learning task in machine learning that requires automatically detecting and learning regularities or patterns in incoming data so that the model may be used to produce or output new examples that could have been derived from the original dataset. Generative Adversarial Networks are a smart way to learn a generative model by framing the problem as a supervised learning problem with two sub-models: the generator model, that we learn to create new examples, and the discriminator model, which tries to categorize instances as true (from the domain) or false (from the domain) (from outside the domain). Both models are trained in an adversarial zero-sum game until the discriminator model is fooled around half of the time, indicating that the generator model is producing convincing examples. GANs are an exciting and rapidly evolving field that fulfills the promise of generative models by generating realistic examples across a variety of problem

domains, most notably in image-to-image translation tasks such as converting summer to winter or day to night photos, and in generating photorealistic photos of objects, scenes, and people that even humans can't tell are fake[69].

2.9.1.7 Ensemble of DL Networks (EDLNs)

As previously stated, the results of an ensemble of different ML classifiers are more effective than the results of individual ML classifiers. Similarly, multiple DL algorithms can be used in parallel to produce better results than each component DL algorithm by organizing them into an ensemble. Any combination of discriminative, generative, or hybrid DL algorithms can be used in EDLNs. EDLNs perform better in uncertain environments with a large number of features, making them ideal for solving complex problems. A heterogeneous EDLN is made up of classifiers from various genres, whereas a homogeneous EDLN is made up of classifiers from the same genre. Both compositions are designed to increase efficiency and produce precise results[70].

2.9.1.8 Deep Neural Network

A neural network is a form of artificial neural network (ANN) that uses a series of algorithms to simulate the human brain. A neural network is made up of four parts: inputs, weights, a bias or threshold, and an output. The algebraic formula will be something like this, similar to linear regression[71]:

$$\sum_{i=1}^m w_i x_i + bias = w_1 x_1 + w_2 x_2 + w_3 x_3 + bias$$

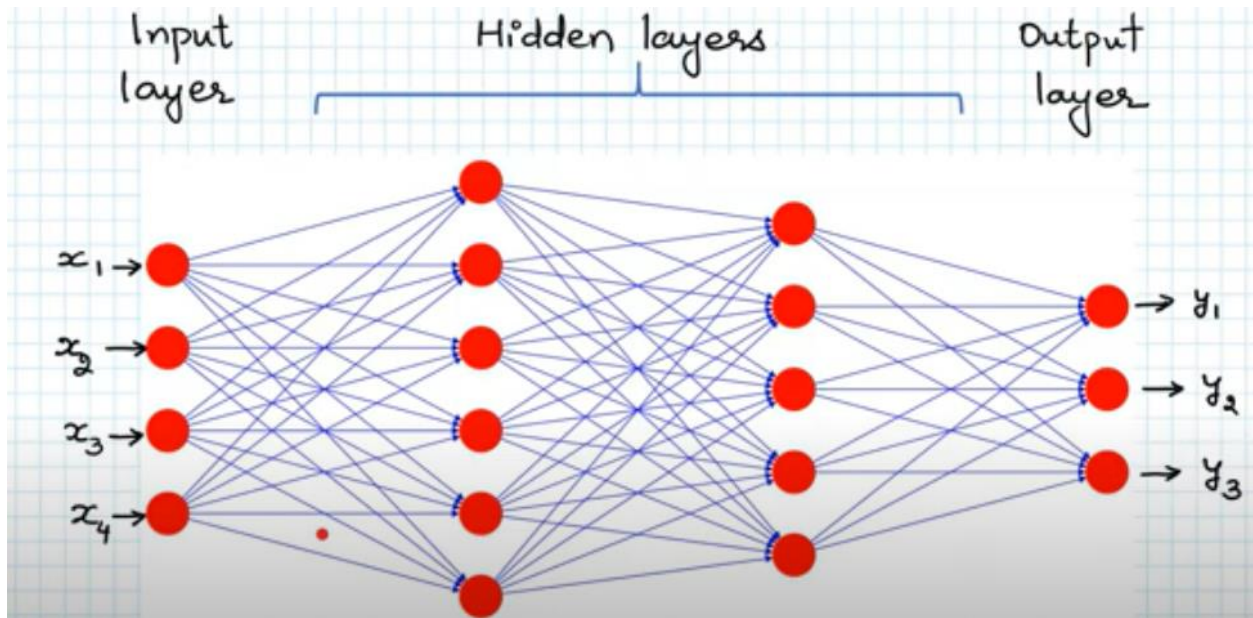


Fig 17: General Structure of Deep Neural Network

The word "deep learning" has gained popularity among researchers who work with machine learning techniques based on artificial neural networks (ANNs) in recent years. Deep learning, like ANNs, is influenced by the brain's overall structure and functionality. Deep learning is made up of several ANN layers that are stacked on top of each other. It's made up of input and output layers that form a layered data flow network. Deep learning is also known as deep structured learning or deep neural network (DNN)[72].

The secret layers are the main difference between ANN learning and deep learning. The secret layers are arranged between the input and output layers in a hierarchical order. Since it processes and computes the given inputs to a greater degree than an ANN, deep learning is more robust than a traditional ANN. In the real world, data volume is increasing, resulting in increased data complexity. The ability to learn from previous layers or a wide collection of data is a key feature of deep learning. Deep learning is a powerful candidate in the selection of machine learning models, particularly for classifying data from unlabeled sample datasets, because of this function[73].

Chapter3: Related Work

Many researchers have investigated how the IDS of the Internet of things could be improved. Some of them are the following. This research[74] presents a deep neural network-based anomaly detection technique for IoT network architecture that efficiently learns valuable complex patterns from IoT network flows to classify traffic as benign or anomalous. On the newly released IoT-Botnet 2020 dataset, the proposed methodology is put to the test. The suggested model outperformed other DL approaches with a detection accuracy of 99.01 percent and a false alarm rate of 3.9 percent, demonstrating its superiority over existing deep learning approaches. The ability of detections accuracy is very high. Since the false alarm rate is 3.9 %, this indicates that false alarm seems too high.

According to an automata model, the authors[75] suggested a new intrusion detection technique for IoT environments. Three forms of IoT attacks are targeted by this tool (jam-attacks, false-attacks, and reply-attacks). a Labelled Transformation Systems' approach is an extension of it. The placement technique of this IDS is a centralized approach since the data collected by network nodes are sent to the Intrusion detection center, which then helps to establish an Event Database. An Event Analyzer based on a specification method is used by the system to detect intrusions. By comparing the abstracted behavior flows, this IDS could detect jamming, false attacks, and reply attacks in the IoT network. It detects the attacks however still it is traditional IDS not intelligent as machine learning algorithms.

The authors [76]proposed an anomaly-based intrusion detection system by using both machine learning and deep learning algorithms, specifically are convolutional neural networks and multilayer perceptron from deep learning algorithms and Random forests from machine learning algorithms as well as they had used BoT-IoT dataset for training and testing. In terms of accuracy and An area under the curve for multiclass classification, random forests and CNN delivered the greatest results. However, the accuracy of multilayer perceptron in detecting some rare attack types is still insufficient.

An analysis of anomaly-based IDS suitable for securing IoT against DoS attacks is performed in this paper[77]. Random forests, AdaBoost, gradient boosted network, highly randomized trees, classification, and regression trees, and multi-layer perceptron are among the seven machine learning classification algorithms evaluated. A random search algorithm is used to find the best

classifier parameters. The accuracy, specificity, sensitivity, false-positive rate, and area under the receiver operating characteristic curve are all used to evaluate the classifiers' efficiency. The CIDDs-001, UNSW-NB15, and NSL-KDD datasets are used to benchmark all of the classifiers. Furthermore, Friedman and Nemenyi post hoc tests are used to find significant differences among classifiers during statistical analysis of performance measures. The average response time of all classifiers is also evaluated on the Raspberry Pi hardware unit. Based on the results of the performance tests and statistical tests, classification and regression trees, as well as the extreme gradient boosting classifier, provide the best trade-off between prominent metrics and response time, making them both ideal for creating IoT-specific anomaly-based IDS.

The primary goal of the proposed study[78] was to develop machine learning models for detecting attacks in IoT networks. IoT networks are extremely vulnerable to hacking, so strategies to secure these devices and networks must be created and tested. An IoT-based platform was developed for this analysis, and it served as a testbed for understanding and executing IoT network attacks. Data were obtained from the deployed network to use machine learning to detect network attacks. Technique The data were classified into normal and modified malicious attack data using four machine learning algorithms. They are SVM, Naïve Bayes, Decision Tree and Adaboost algorithm. From the algorithms when checked with test data, it can conclude that the decision tree has the highest accuracy of all the classifier models. SVM, Nave Bayes, and AdaBoost-based classifications have also performed admirably, with few misclassifications. And also the used data set is Sensor480 is the name of the 480-record dataset that was developed. Sensor480, a classifier created by the authors, is used to test the output of the classifiers. The algorithms had a high degree of accuracy in classifying the data. According to the results, machine learning algorithms can be used to build IDS for IoT networks. The most difficult aspect of creating an IDS based on machine learning principles is producing a practical and high-quality training dataset; data flow in the network should be of good quality during the attack process because interception is only possible with a continuous flow of data. Since the network would be used by several heterogeneous devices, the ML model will take into account a broad range of data. By overcoming these obstacles, a suitable IDS for the IoT environment can be built. To ensure that IoT devices are safe from cyber threats, security features must be considered early in the design process. The weakness side of this paper is few misclassifications.

This paper[79] proposed a multi-agent intrusion detection system model based on blockchain and deep learning. Because of the simplicity of a multi-agent scheme, this new IDS can be used in a variety of IoT environments. All communication agents' activities will be registered on the blockchain, making the system safer against threats such as data tampering and disclosure. Multi-agent reinforcement algorithms can aid the system in continuously improving its performance. The use of a neural network in intrusion detection systems is studied in this paper using this model, and simulation results show that the deep learning algorithm outperforms the conventional methods. The simulation using the NSL-KDD dataset demonstrates DNN's high accuracy in detecting intrusions on the IoT transport layer. The DNN model outperforms other machine learning approaches such as decision trees when it comes to separating anomaly from normal. The DNN model has a 98 percent overall accuracy score, demonstrating the feasibility of deep learning algorithms for IoT system IDS. However, there are a few problems that will need to be discussed in future work. While the DNN model has a high accuracy rate in separating the more common attack types, some uncommon attack types cannot yet be detected with sufficient accuracy. More research is needed in this field.

This paper[10] prepossessed a machine, learning-based lightweight intrusion-detection model. This model could identify new attacks and defend IoT nodes from both internal and external threats. To find the best classifier model, they used three lightweight feature selection algorithms to test multiple machine learning classifier models and tried to optimize the parameters of each algorithm to get an effective classifier model with high accuracy and precision, as well as a low false negative. They learned and evaluated their model using the datasets KDD99, NSL-KDD, and UNSW-NB15.

Several studies and academic papers on IDSs for IoT have been published as a result of this article[11]. However, there are numerous open research challenges and issues, especially in the use of machine learning and deep learning techniques for anomaly and intrusion detection in the Internet of Things. The problem is that there is no standard process in place to ensure that the proposed structures or methods are accurate. The majority of the research papers show the evaluation of their proposed systems using synthetic datasets and discuss a single problem that may or may not operate in the real world with real data and in the presence of other issues. As this and other similar studies on the state of the art in IDS for IoT have shown, designing an IDS that covers at least the most important aspects of an effective IDS, namely that it is deployable, online,

scalable, works effectively on real data, and satisfies all stakeholders' requirements, is extremely difficult. Instead, the majority of the published work presents evaluation results based on fictitious datasets, covers only a portion of the system, and presents results with skewed parameters. Furthermore, proving the completeness and accuracy of any proposed IDS is exceedingly difficult to identify and achieve. As a result, one of the study's findings is that designing a comprehensive IDS that can provide high precision, scalability, robustness, and security against all types of threats is extremely difficult. Listed below are some of the major issues and challenges that researchers face now and in the future. Because IoT security is still in its infancy, there is plenty of scope for research, particularly in the areas of anomaly and intrusion detection using machine learning and deep learning techniques. One of the most important IoT security techniques is a machine and deep learning-based IDS.

In this paper[12] the proposed system in intrusion detection systems are classified using the Nave Bayes algorithm and with the corresponding KDDcup99 dataset. The outcomes are then compared to those obtained by using a deep learning model. By using a deep learning algorithm instead of a machine learning algorithm, the accuracy of detecting an attack improves. When tested on test data, the Deep Learning model outperformed conventional machine learning systems in classifying network data into normal/attack. This paper gives a direction to use other machine learning algorithms deep learning algorithms, and compare the results.

Chapter 4: Proposed Solution

4.1 Introduction

The design strategy and datasets utilized in constructing an intelligent model for an IoT network are briefly described in this chapter. The new architecture designed for the Internet of Things demonstrates how the dataset is preprocessed, how the model learns from the dataset, how the generated model is tested with new datasets, and how the intrusion detection system creates a model that classifies to normal and Dos attack. The generated model was learned by a deep neural network from both normal and DoS traffic of the IoT networks. The dataset is consists of both normal and DoS traffic which is collected by using Wireshark from IoT networks within the connected IoT devices. Since the collected dataset is a raw dataset that is prepared into the correct format by data cleaning and data transformation which is suitable for learning and testing to generate an intrusion detection model. In the end the generated model after learns and tests it classify whether normal or Dos attack.

4.2 System Architecture

This new architecture showed the design of deep neural intrusion detection which is an anomaly-based intrusion detection system. This architecture explained the phases of deep neural network learning from the dataset. The dataset is preprocessing and ready to learn by the deep neural network. After the dataset is preprocessed it fits into the deep neural network to learn. After that, the new intrusion detection is generated, and it tests by the new dataset to evaluate the model's efficiency.

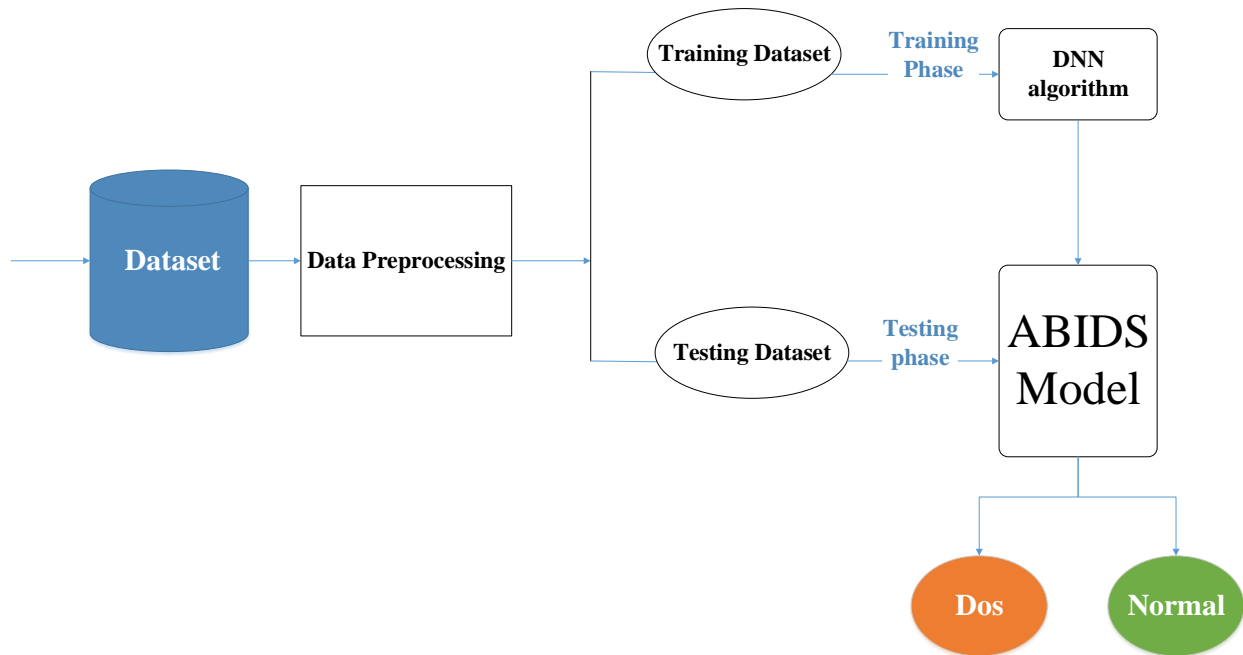


Fig 18: proposed Architecture

The anomaly-based intrusion detection system deep neural network (ABIDS) module captures the network packet from the network interface it analysis and checks it whether it is DoS or normal traffic. If it is DoS traffics it generates an alert immediately the responsible agent take an action before crashes the IoT network. If the packet is normal it passes to IoT networks.

4.2.1 Components of the Proposed IDS

4.2.1.1 DNN Training

A deep neural network is a system that uses numerous layers of nodes to extract high-level functions from input data. It entails repurposing data more abstractly and creatively. At its most basic level, a deep neural network (DNN) is a neural network with some level of complexity, usually at least has two layers. Deep neural networks process information in complicated ways using advanced mathematical modeling[80], [81].

The deep neural network training phase involves teaching the model with a sufficient learning dataset to generate a good and accurate model. For learning and testing the dataset the used algorithm is a deep neural network algorithm to generate a model. The purpose of this module is to generate an accurate and efficient deep neural model.

4.2.1.2 Centralized based Deployment Anomaly-based DNN Model

Anomaly-based IDSs compare a system's behaviors at any given time to a normal behavior profile and emit an alarm when a divergence from normal behavior reaches a certain threshold. Anomaly detection module accepts network packet that has been received from network interface card. Once it received the packet this module load the packet into the model, and the model detects whether it is a DoS attack or normal network traffic. If the packet is classified as normal then it will be allowed to pass into IoT networks. But if the packet is classified as DoS it will generate an alert alarm.

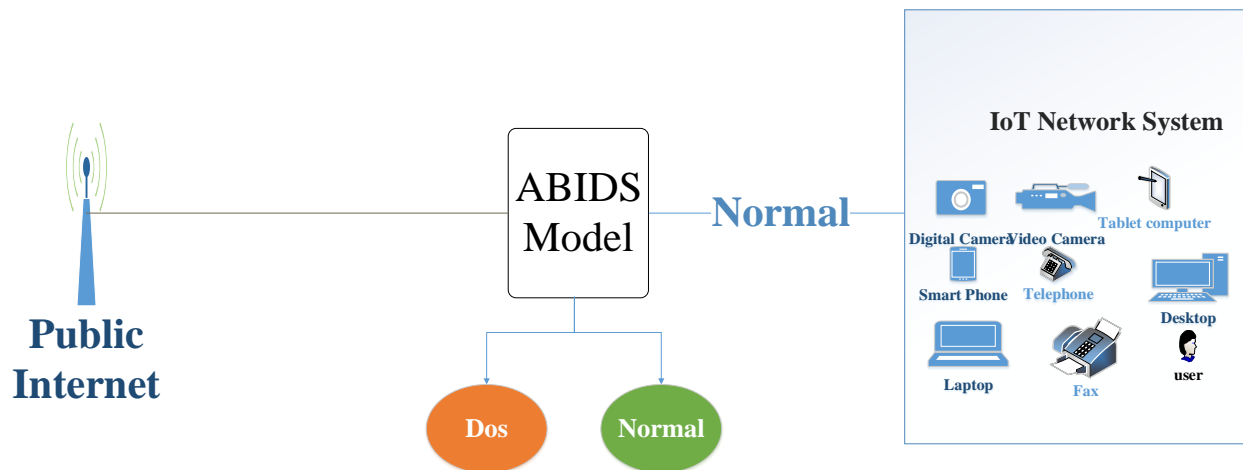


Fig 19: Deployed Anomaly-based IDS Model

4.2.1.3 Dataset collection

The dataset is collected by creating a local area wireless connection of the Internet of Things network. The devices used to collect are D-Link DWA-125 Wireless N 150 USB Adapter (rev.A2), NodeMCU(ESP_C62591), Huawei laptop, Toshiba Laptop, HUAWEI smartphone, and Techno Smartphone. It is explained in detail in chapter one methodology part.

4.2.1.4 Data Preprocessing

I have prepared the dataset in the correct format through data cleaning and data transformation. The parts of the dataset are presented in different Comma-separated values (CSV).

4.2.1.5 Deep Learning Model Training

Deep Learning is a subset of machine learning or a specific type of machine learning. It works in the same manner that machine learning does in terms of technology, but with different capabilities

and techniques. It is based on the functionality of human brain cells known as neurons, and it gives rise to the concept of artificial neural networks. A deep neural network, or deep neural learning, is another name for it. Deep learning models learn and discover insights from data using multiple layers. The preprocessed data is fed into the deep neural network algorithm after feature extraction is done. Finally, an IoT intrusion detection model is generated using the dataset with extracted features. The generated model is a sequential Keras layer consist of the input layer, hidden layer, and output layer. The input layer and output layers have 20 and 1 neurons respectively. There are five hidden layers, from hidden layer one up to hidden layer five. The number of neurons from the hidden layer one to hidden layer five are 26,22,21,25, and 10 respectively. The used activation functions are rectified linear unit(ReLu) and sigmoid function. Rectified linear unit is used for the input layer and the hidden layer whereas sigmoid function is used for the output layer.

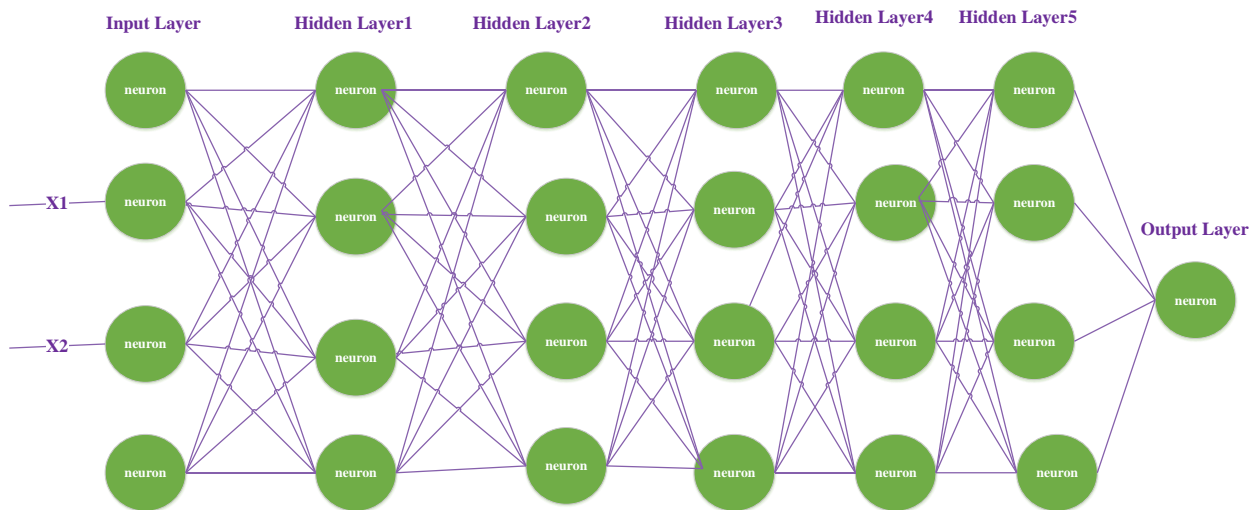


Fig 20: Structure of Deep neural Network

Chapter 5: Implementation and Performance Evaluation

5.1 Overview

The implementation of the proposed intrusion detection system for IoT networks was covered in this chapter. The suggested anomaly-based intrusion detection system is built on a deep neural network that was implemented using Keras, an open-source neural network library written in Python, and implemented in the cloud using Google Collab. The system's implementation was broken down into the following stages. They are input data collection and preprocessing, training the dataset by using a deep neural network and generating the model, testing the model, and evaluate the model.

5.2 Tools used

Different tools have been used for implementing the proposed architecture. The tools are:- Laptop computer, Ubuntu operating system, Kali Linux operating system, Microsoft Excel to edit and examine CSV files, Google Collaboratory for implementing the model, and Python programming language.

5.3 Dataset

The required materials to collect the dataset are:-Local area wireless IoT connection, NodeMCU (Node MicroController Unit), Wireshark, laptop, VMware, Kali Linux operating system, Ubuntu operating system, and smartphone. The dataset is collected by creating a local area wireless Internet of Things connection. The local connection is creating by using D-Link DWA-125 Wireless N 150 USB Adapter (rev.A2). The devices are connected to the local area wireless connection through the Connectify hotspot. Those devices connected to local area connections are:- NodeMCU(ESP_C62591), Huawei laptop, Toshiba Laptop, HUAWEI smartphone, and Techno Smartphone. The dataset is collected by tracking using the Wireshark. The total number of the collected dataset is 83,225 records. From this 67.06% is Dos and 32.94% is normal records.

5.3.1 Dataset Description

From the total collected dataset of eighty-three thousand two hundred twenty-five(83,225) records the eighty (80%) percent is used for training purposes which is sixty-six thousand five hundred

eighty records (66,580). The remaining twenty percent (20 %) is used for testing purposes which is sixteen thousand six hundred forty-five records (16,645). 11250 records are Dos and 5395 are normal records.

| Both Normal and Attack | Records | Percentage |
|------------------------|---------|------------|
| Normal | 27,414 | 32.94% |
| DoS | 55,811 | 67.06% |
| Total | 83,225 | |

Table 1: Dataset Distribution

5.3.2 Dataset Features

The dataset has the following features.

| Feature name | Feature description |
|---------------------|-----------------------------|
| ip.src | Source IP address |
| IP.dst | Destination IP address |
| IP. proto | IP protocol number |
| IP.length | IP length |
| tcp.srcport | TCP source port number |
| tcp.dstport | TCP Destination Port number |
| IP. TTL | IP Time-to-live |
| tcp.seq | TCP sequence number |
| IP.checksum | IP checksum |
| tcp.ack | TCP acknowledgment |
| udp.srcport | UDP Source port number |
| UDP.dstport | UDP Destination Port Number |
| frame.length | Frame length |

Table 2: Features of Dataset

5.4 Data preprocessing

Data preprocessing is the procedure for preparing raw data for use in a deep learning model. It's the first and most important stage in building a machine learning model. It is not always the case that we come across clean and prepared data when working on a machine learning project. And, before doing any data-related activity, it is necessary to clean the data and format it. As a result, we use a data pretreatment activity for this. Real-world data typically contains noise, missing values, and is sometimes in an unsuitable format that cannot be used directly for deep learning models. Data preprocessing is a necessary step for cleaning data and preparing it for use by a deep learning model, which improves the model's accuracy and efficiency. Therefore the steps for preprocessing are:-converting the raw dataset into comma-separated values(CSV), importing

libraries, Importing the Datasets, Handling Missing data, Encoding Categorical data (have used OneHotEncoder and LabelEncoder), and Splitting the Dataset into the Training set and Test set.

5.5 Training

Here are the used algorithms for training deep neural network algorithms. The preprocessed dataset is fed into the deep neural network with one input layer, five hidden layers, and one output layer. The generated model is a sequential Keras layer consist of the input layer, hidden layer, and output layer. The input layer and output layers have 20 and 1 neurons respectively. There are five hidden layers from hidden layer one up to hidden layer five. The number of neurons from the hidden layer one to hidden layer five are 26,22,21,25, and 10 respectively. The used activation functions are rectified linear unit(ReLu) and sigmoid function. Rectified linear unit is used for the input layer and the hidden layer whereas sigmoid function is used for the output layer. Finally, the model is compiled using binary category entropy loss type and adam optimizer. The basic steps used for training are listed below and it is described in detail in chapter one in the methodology part of 1.5.5.

- Get the Dataset
- Importing Libraries
- Importing the Dataset
- Encoding Categorical data
- Splitting the Dataset into the Training set and Test set
- Generate anomaly-based IDS Model:
- Compile Keras Model
- Fit Keras Model
- Evaluating the Model
- Make Predictions

| Parameter | Value |
|---------------------|-----------------------------------|
| Used algorithms | Deep neural network |
| Activation function | Sigmoid and rectified linear unit |
| Loss | binary category entropy |
| Optimizer | Adam |
| Epochs | 50 |
| Batch size | 32 |

Table 3: parameter setting

5.6 Implementation of the Components

An anomaly-based intrusion detection system is a type of intrusion detection system that monitors system activity and categorizes it as normal or DoS to detect network and computer intrusions and misuse. Heuristics or rules are used to classify the items.

5.6.1 DNN Training Module

Deep neural network training took place in a cloud platform known as Google Collaborator. It uses a python programming language to feed data in deep learning algorithms, to preprocess and train the model.

5.6.2 Anomaly-based IDS Model

The anomaly-based intrusion detection system monitors the IoT network, comparing typical activity against a threshold. The model recognizes any deviation from normal behavior as an anomaly, which is an intruder. The following are the components of the built anomaly-based IDS using a deep neural network model.

5.6.2.1 Anomaly Monitoring

The anomaly detection module is implemented using a deep neural network model. Once abnormal behavior is detected from incoming packets it will be forwarded to the anomaly classification module.

5.6.2. 2 Anomaly Classification

The detected anomalous packet is classified into two classification groups. Namely normal and Dos.

5.6.2.3 Anomaly Alarm

When the classifier model detects an attack it generates an alarm to the responsible agent.

5.7 Experiments and Result

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

The following are the result of the DNN algorithms.

```
2081/2081 - 9s - loss: 1.4484e-04 - accuracy: 1.0000 - val_loss: 1.9995e-04 - val_accuracy: 0.9999
```

Fig 21: Accuracy of the DNN

As it is shown in above fig 21 the validation accuracy of the generated deep neural network is 99.99%. This indicates that using a deep neural network algorithm for intrusion detection systems is prominent for identifying the intruder with high accuracy. The highest accuracy indicates that the performance of the model is very effective to identify the attacks that are deviations from the normal behavior of the IoT networks.

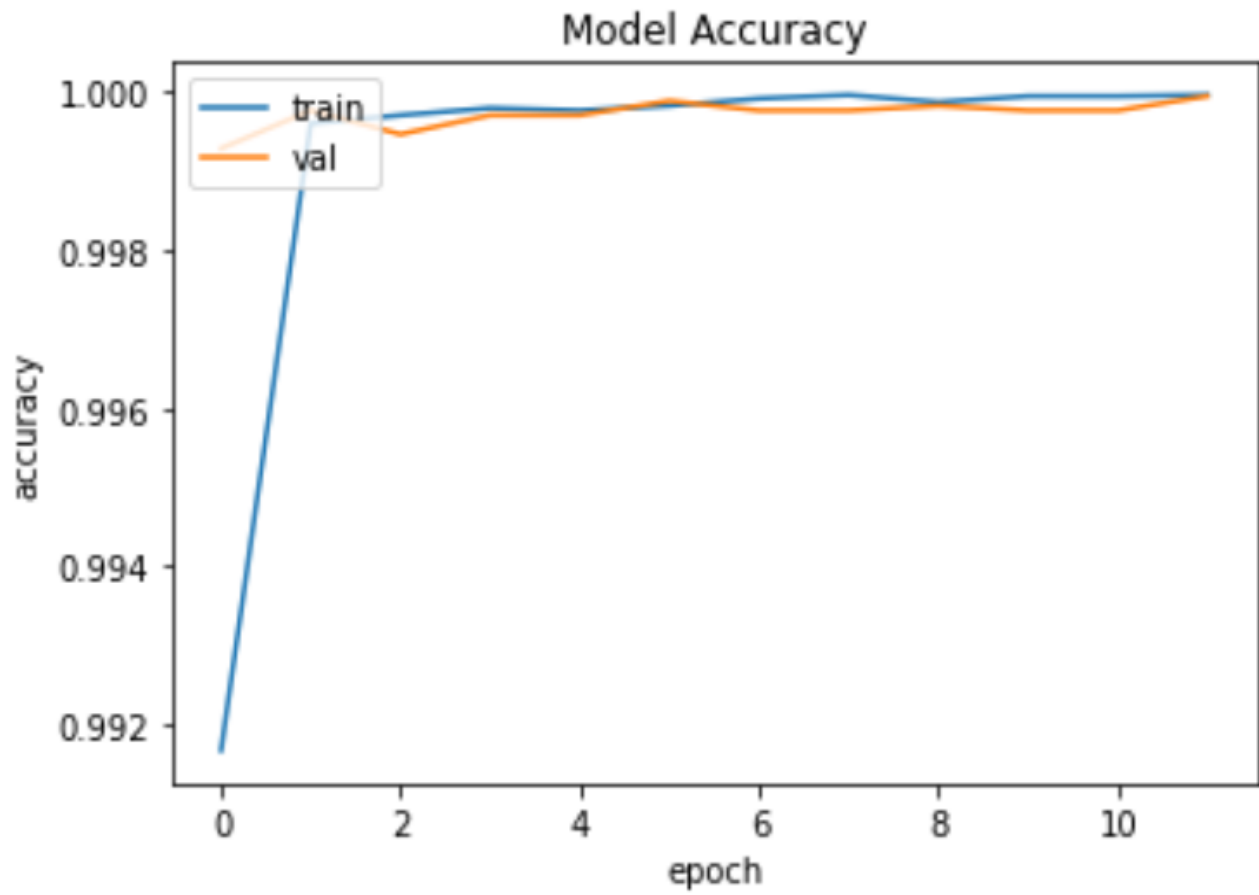


Fig 22: The Accuracy DNN Model Graph

— Validation Accuracy

— Training Accuracy

The above graph shows the validation and training accuracy of the generated model.

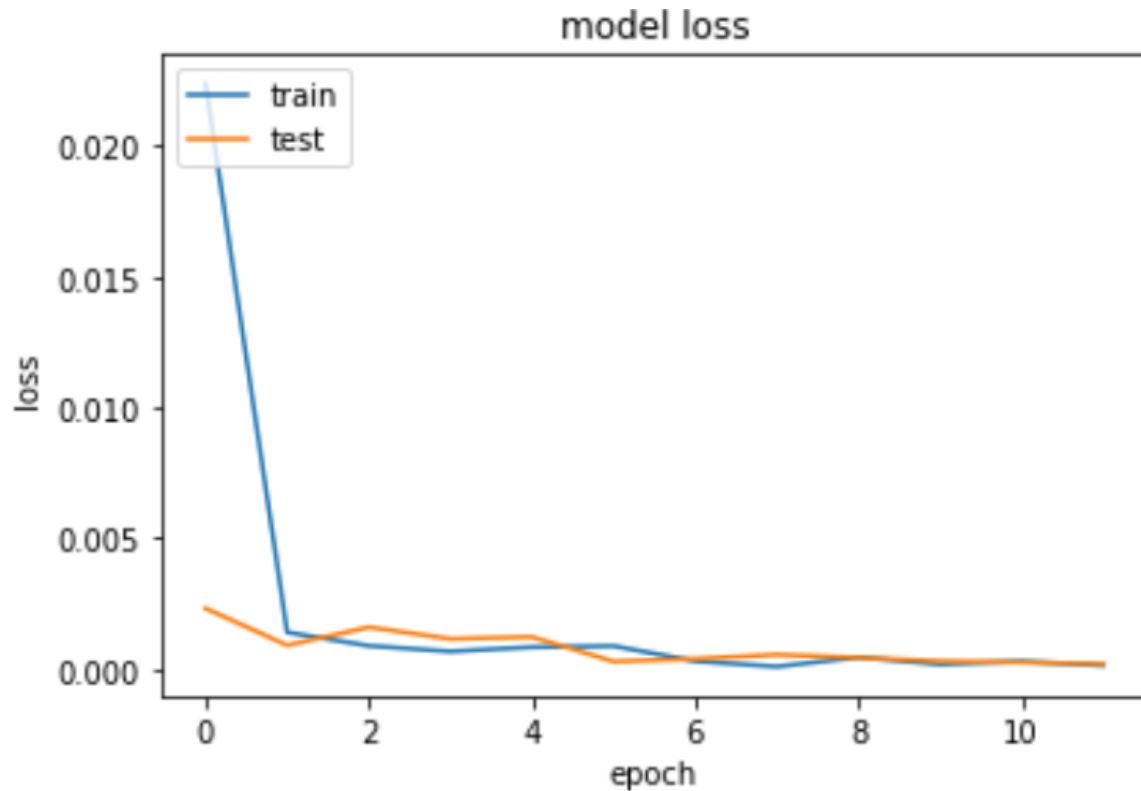


Fig 23: DNN Loss Graph

— Validation lost

— Training lost

5.8 Performance Evaluation

Many assessment metrics are created using the values in the confusion matrix to evaluate the performance of the IDS. The following is a description of these values: The confusion matrix is a matrix that is used in deep learning and machine learning algorithms to assess the performance of classification models by using the testing datasets and using various parameters. The parameters that were evaluated using the confusion matrix are accuracy, F1 score, precision, and recall.

(True Positive (TP): The number of records accurately categorized to the Normal class.

True Negative(TN): The number of data correctly labeled as Attack

False Positive (FP): the number of records categorized as Attacks when they should have been classified as Normal.

False Negative (FN): the number of Attack reports classed mistakenly as Normal.

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

Recall The 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.

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

| | | | |
|---------------|--------|------------------|-------|
| | | Predicted values | |
| | | Normal | Dos |
| Actual Values | Normal | 5395 | 0 |
| | Dos | 1 | 11249 |

Table 4: Confusion Matrix of The DNN Model

From the above Confusion Matrix, the values of TP, TN, FP, and FN are 5395, 11249,0, and 1 respectively.

The following formulas give the most often used evaluation measures based on the above values:

$$TP = \frac{TP}{TP+FN} \quad (5.8.1)$$

$$TN = \frac{TN}{TN+FP} \quad (5.8.2)$$

$$FP = \frac{FP}{FP+TN} \quad (5.8.3)$$

$$FN = \frac{FN}{FN+TP} \quad (5.8.4)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (5.8.5)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (5.8.6)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (5.8.7)$$

$$F1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5.8.8)$$

5.8.1 Performance Evaluation: Accuracy

As discussed previously the effectiveness of an anomaly 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 DoS attack. The accuracy is calculated by the following formula that is stated in equations 5.8.5. from the confusion matrix, the true positive value is 5395, the value of true negative is 11,249, the value of false positive is 0 and the value of false negative is 1. The fore when those values were substituted in the given formula the validation accuracy of the model is 99.99%.

$$\text{Accuracy} = \frac{TP+TN}{TN+TP+FP+FN} = 99.99$$

5.8.2 Performance Evaluation: Precision

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

$$\text{Precision} = \frac{TP}{TP+FP} = 1$$

5.8.3 Performance Evaluation: Recalls

The recall is the measure of our model correctly identifying True Positives.

$$\text{Recall} = \frac{TP}{TP+FN} = 0.9998$$

5.8.4 Performance Evaluation: F1

F1 score: The F1 score is the harmonic mean of precision and recalls it takes into consideration

$$\text{both measurements. } F1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} = 0.9999$$

5.8.5 Performance Evaluation: Completeness

This completeness aims to minimize the false alarm rate. When the false alarm rate decreases we can assume that the IDS will start examining both known and unknown attacks in a network. Therefore the proposed anomaly detection system is less in alerting the false alarm since it is prominent in every aspect of the requirement of the performance evaluations and it is complete in identifying whether it is a DoS attack or normal traffic classifications.

5.8.5 Performance Evaluation: False Alarm Rate(FAR)

False alarm rate is an alarm falsely predicted the normal traffic behavior of the network as attack traffic. This issue is one of the main problems of the anomaly-based intrusion detection system. From the confusion matrix described before $FP = 0$ and $TN = 11,249$ when it calculates using the formula the result of FAR is zero. The false alarm rate is calculated as $FAR = \frac{FP}{FP+TN} = 0$. The false alarm rate is zero means the finding model has not predicted the actual value normal as a DoS attack.

5.8.5 Performance Evaluation: False Negative Rate(FNR)

It indicates the misclassification of the generated model. The model predicted the DoS attack negatively to normal traffic. Shortly the actual value was DoS but the model predicted it as normal. From the confusion matrix described before the value of $FN = 1$ and the value of $TP = 5395$. The probability of the false-negative rate is calculated in the following formula. There fore all values are given to calculate the false-negative rate.

$$FNR = \frac{FN}{FN+TP} = 0.0001853568 = 0.01853568 \%$$

5.9 Discussion Results

One of the critical issues that need to be addressed for the Internet of Things is security problems. A variety of obstacles prevent IoT devices from being secured and end-to-end protection in an IoT environment from being achieved. Since networking equipment and other artifacts are still a relatively new concept, protection hasn't always been a top priority during the design process of a product. Another problem with IoT devices is that they are frequently resource-constrained and lack the compute resources required to implement strong security. IDS is one of the main methods used for information systems and network system security. IDS tracks the activities of a host or network, alerting the system administrator when a security breach is identified. Despite the

sophistication of IDS technology for conventional networks, existing implementations are insufficient for IoT systems due to complex IoT features that affect the creation of IDS. The discussion result is explained in detail below.

From the confusion matrix in table 4, the output of TP = 5395 , TN = 11249 , FP = 0 ,and FN = 1. By using the following formula the result of the study is summarized in the following table.

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

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

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

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

$$\text{FAR} = \frac{FP}{FP+TN}$$

$$\text{FNR} = \frac{FN}{FN+TP}$$

| Algorithm | Accuracy(%) | Precision(%) | Recall(%) | FAR(%) | F1 Score(%) | FNR (%) |
|-----------|-------------|--------------|-----------|--------|-------------|------------|
| DNN | 99.99 | 1 | 99.98 | 0 | 99.99 | 0.01853568 |

Table 5: The summary result of the proposed system

The model is evaluated using a confusion matrix output and it was described in table 4. The model was tested by a new dataset. Which is twenty percent (20%) from the whole dataset. The total dataset used for testing is sixteen thousand six hundred forty-five records (16,645). 11250 records are Dos and 5395 records are Normal traffic. The metrics used to evaluate the generated model which is anomaly-based IDS using deep neural networks are:- Accuracy, precision, recall, F1 Score, False alarm rate, and False-negative rate.

- True Positive (TP): means the actual values are normals and the predicted values by the model are normals
- True Negative (TN): means the actual values are Attacks and the predicted values by the model are an attacks
- False Positive (FP): means the actual values are normal but the predicted values are attacks.

- False-negative (FN): means the actual values are attacks but the predicted values are normal.

The accuracy of the finding is 99.99 % which indicates the capacity of the generated model in predicting and identifying is very prominent. Because accuracy is used to evaluate the generate anomaly base IDS how is efficient and accurate to classify the normal traffic and DoS attack. From the confusion matrix TP = 5395 which means when the model predicted the actual values were 5395 records of normal traffic and it predicted 5395 Normal traffics. That means the model predicted the whole normal traffic as normal traffic without missing the whole record. TN = 11249 from the confusion matrix which means the actual value was 11250 Dos records and the model predicted 11249 DoS records. Here it missed one record that was predicted as normal traffic. FP = 0 means the actual value of normal traffic is not predicted as a Dos attack. FN = 1 which means from the total records of 11250 Dos attack records the model predicted 1 Dos Attack records as Normal traffic. FAR = 0 % which means there is no misclassified actual value of normal is not predicted as DoS attack. The false-negative rate value is $0.0001853568 = 0.01853568 \%$.

| Existing papers | Accuracy(%) | Precision(%) | Recall(%) | FAR(%) | F1 Score (%) | FNR (%) |
|-----------------|-------------|--------------|-----------|--------|--------------|---------|
| [74] | 99.01 | 99.03 | 98.020 | 3.915 | 98.642 | 0.044 |
| [78] | 98 | 97 | 97 | - | 97 | - |

Table 6: The results of the existing papers

The result of the new find is described in table 5 and the results of the previous existing study are described in table 5. Evaluating the contribution of the new study is necessary to compare with related previous existing research papers. The performance of the deep learning model is evaluated using accuracy result, precision result, recall result, false alarm rate result, F1 score result, and false-negative rate result.

| Metrics % | Existing Study [74] | New Study | Outperformed % |
|-----------|---------------------|------------|----------------|
| Accuracy | 99.01 | 99.99 | 0.98 |
| Precision | 99.03 | 100 | 0.97 |
| Recall | 98.020 | 99.98 | 1.96 |
| F1 Score | 98.642 | 99.99 | 1.348 |
| FAR | 3.915 | 0 | |
| FNR | 0.044 | 0.01853568 | |

Table 7: Comparing the new study with existing paper [74]

The new study outperforms the existing study in all metrics. The accuracy of the new study outperforms the existing study by 0.98. The precision of the new study outperforms the existing study by 0.97. The recall of the new study outperforms the existing study by 1.96. the F1 score of the new study outperforms the existing study by 1.348. the false-negative rate of the new study is 0 whereas the false negative of the existing study is 3.915. the false-negative rate of the new study is 0.01853568 whereas tt the false-negative rate of the existing study is 0.044. The result of the new finding outperformed the existing study in all metrics.

| Metrics % | Existing Study [78] | New Study | Outperformed % |
|-----------|---------------------|-----------|----------------|
| Accuracy | 98 | 99.99 | 1.99 |
| Precision | 97 | 100 | 3 |
| Recall | 97 | 99.98 | 2.98 |
| F1 Score | 97 | 99.99 | 2.99 |

Table 8: Comparing the new study with existing paper [78]

The new study outperformed the existing study in [78]. The accuracy of the new study outperformed by 1.99 %. The precision of the new study outperformed the existing study by 3. The recall of the new study outperformed the existing study by 2.98 and the F1 score of the new study outperformed the existing study by 2.99.

The result of the finding is prominent in all the required metrics. This indicates the proposed model achieved its main objective. The anomaly-based intrusions detection system is mostly evaluated by accuracy and false alarm rate. Therefore the accuracy and false alarm rate of the new finding

were achieved. The accuracy of the new model of anomaly-based IDS is 99.99 %. The false alarm rate of the new model of anomaly-based IDS is 0%. That means there is no misclassification of false positives. Therefore the new model answers one of the main problems was in the traditional anomaly-based detection system was high false alarm rate even if this was high in a recent existing study. when it is summarized the contribution and the significance of the new study is concluded that using deep neural network-based anomaly-based intrusion detection is robust for security purposes for the Internet of Things is intelligent with efficient and effective accuracy that is proved from the result.

Chapter 6: Conclusion and Future Work

6.1 Conclusion

Internet of Things is the network of tiny objects. Its main goal is to connect the objects at any time, anywhere, and ubiquitous for sharing information and providing different smart applications for the user. IoT systems are faced with cybersecurity challenges. IoT network is the recent technology, diverse and heterogeneous, and the designers, as well as the manufactures of the IoT devices, are concerned about distributing them throughout the world rather than securing the IoT network device.

The intrusion detection system is software or hardware used to monitors the IoT networks. The intrusion detection system that is applied in the traditional network is not sufficient to prevent the Internet of Things. Because the traditional IDS requires high computing power and energy to run it on the IoT system. IoT network systems are resource-constrained like computing processing and battery power.

The new study proposes an anomaly-based Intrusion detection system model based on a deep neural network. The deep neural network creates multiple hidden layers. The creating of multiple hidden layers can increase the learning capacity and the efficiency of the deep neural network model. The generated model is training and testing using the coomined_iot3 dataset which is collected from the IoT networks. The anomaly-based IDS is evaluated by accuracy, precision, recall, F1 score, false alarm rate, and the false-negative rate. The results of those metrics are outperformed when compared to the related work. The accuracy of then the anomaly-based model is 99.99 %. The false alarm rate of the model is 0 %. The new study result indicates the model is novel in identifying the DoS attack with the prominent accuracy, and with the zero false alarm rate. According to the result the new study outperformed in all metrics from the existing related work. Therefore the combination of intrusion detection with deep neural techniques for the classification of cyberattack challenges of the Internet of things is robust and powerful.

6.2 Future work

For future work, I recommended investigating with other deep neural networks such as convolutional neural networks, recurrent and other machine learning algorithms with other datasets and comparing the result with this research, and the next future work is implementing the model in the real IoT networks.

Bibliography

- [1] Q. F. Hassan, “Part I Concepts and Perspectives,” *Internet Things A to Z Technol. Appl.*, pp. 1–6, 2018.
- [2] S. Chawla, “Deep Learning based Intrusion Detection System for Internet of Things,” 2017, [Online]. Available: <https://digital.lib.washington.edu/researchworks/handle/1773/39829>.
- [3] I. Butun, P. Osterberg, and H. Song, “Security of the Internet of Things: Vulnerabilities, Attacks, and Countermeasures,” *IEEE Commun. Surv. Tutorials*, vol. 22, no. 1, pp. 616–644, 2020, DOI: 10.1109/COMST.2019.2953364.
- [4] M. A. Jabraeil Jamali, B. Bahrami, A. Heidari, P. Allahverdizadeh, and F. Norouzi, *IoT Architecture BT - Towards the Internet of Things: Architectures, Security, and Applications*. 2020.
- [5] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey,” *Comput. Networks*, vol. 54, no. 15, pp. 2787–2805, 2010, DOI: 10.1016/j.comnet.2010.05.010.
- [6] M. A. Chaqfeh and N. Mohamed, “Challenges in middleware solutions for the internet of things,” *Proc. 2012 Int. Conf. Collab. Technol. Syst. CTS 2012*, pp. 21–26, 2012, DOI: 10.1109/CTS.2012.6261022.
- [7] A. Torkaman and M. A. Seyyedi, “Analyzing IoT Reference Architecture Models,” *Int. J. Comput. Sci. Softw. Eng. ISSN*, vol. 5, no. 8, pp. 2409–4285, 2016, [Online]. Available: www.IJCSSE.org.
- [8] M. Kufner, W. v. Roth, and H. Schmidt, “Zur Festigkeit von Holzmasten mit Xyloterus-Befall,” *Holz als Roh- und Werkst.*, vol. 31, no. 9, pp. 337–341, 1973, doi: 10.1007/BF02606980.
- [9] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, “A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security,” *IEEE Commun. Surv. Tutorials*, vol. 22, no. 3, pp. 1646–1685, 2020, DOI:

10.1109/COMST.2020.2988293.

- [10] S. Fenanir, F. Semchedine, and A. Baadache, “A machine learning-based lightweight intrusion detection system for the internet of things,” *Rev. d’Intelligence Artif.*, vol. 33, no. 3, pp. 203–211, 2019, DOI: 10.18280/ria.330306.
- [11] J. Asharf, N. Moustafa, H. Khurshid, E. Debie, W. Haider, and A. Wahab, “A review of intrusion detection systems using machine and deep learning in the internet of things: Challenges, solutions, and future directions,” *Electron.*, vol. 9, no. 7, 2020, DOI: 10.3390/electronics9071177.
- [12] S. Malliga, S. Darsniya, and P. S. Nandhini, “A network intrusion detection system for IoT using machine learning and deep learning approaches,” *Int. J. Adv. Sci. Technol.*, vol. 29, no. 3 Special Issue, pp. 1017–1023, 2020.
- [13] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, “A survey of intrusion detection in Internet of Things,” *J. Netw. Comput. Appl.*, vol. 84, no. September 2016, pp. 25–37, 2017, DOI: 10.1016/j.jnca.2017.02.009.
- [14] A. Khraisat and A. Alazab, “A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges,” *Cybersecurity*, vol. 4, no. 1, 2021, DOI: 10.1186/s42400-021-00077-7.
- [15] ElectronicWings, “Introduction to NodeMCU,” vol. 2, pp. 1–5, 2020, [Online]. Available: <https://www.electronicwings.com/>.
- [16] Make-it.ca, “NodeMCU ESP8266 Detailed Review Specifications, Overview and Setting Up Your NodeMCU,” Make-it.ca, pp. 1–7, 2021, [Online]. Available: <https://www.make-it.ca/nodemcu-arduino/nodemcu-details-specifications/>.
- [17] A. Ide, “Getting Started w / NodeMCU What Is NodeMCU ? How to program NodeMCU using Arduino IDE,” pp. 1–8, 2021.
- [18] A. Project and S. In, “ESP8266 WiFi Module Introduction ESP8266-01 Module Pin Description,” pp. 1–10, 2021.
- [19] G. Y. Orhan, “4 Reasons Why You Should Use Google Colab for Your Next Project,”

- Towards Data Science. 2020, [Online]. Available: <https://towardsdatascience.com/4-reasons-why-you-should-use-google-colab-for-your-next-project-b0c4aad39ed>.
- [20] J. Notebooks, “8 Tips For Google Colab Notebooks To Take Advantage Of Their Free-of-charge 12GB-RAM 1 . Map your Google Drive 2 . Work with your files transparently in your computer,” pp. 1–10, 2021.
- [21] H. Sgd, “The algorithms are described in the official documentation here (<https://keras.io/optimizers/>). In this notebook I go into slightly more detail about how each of the various algorithms is constructed.,” pp. 1–10, 2021.
- [22] V. Bushaev, “Adam — latest trends in deep learning optimization,” <https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c>, pp. 1–18, 2018.
- [23] “Internet of Things-IoT : Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges,” vol. 6, no. 5, 2016, DOI: 10.4010/2016.1482.
- [24] A. S. Gillis and T. Writer, “Our expert picked these 6 IoT security best practices for you to know,” pp. 1–12, 2021.
- [25] X. Xu, J. Zhou, and H. Wang, “Research on the basic characteristics, the key technologies, the network architecture and security problems of the Internet of things,” Proc. 2013 3rd Int. Conf. Comput. Sci. Netw. Technol. ICCSNT 2013, pp. 825–828, 2014, DOI: 10.1109/ICCSNT.2013.6967233.
- [26] J. Y. Khan, “Introduction to IoT Systems,” Internet of Things (IoT), pp. 1–24, 2019, DOI: 10.1201/9780429399084-1.
- [27] K. Kour, J. Kour, and P. Singh, “Smart Applications of Internet of Things,” ICSCCC 2018 - 1st Int. Conf. Security. Cyber Comput. Commun., pp. 143–146, 2018, DOI: 10.1109/ICSCCC.2018.8703278.
- [28] H. Sedjelmaci, S. M. Senouci, and M. Al-Bahri, “A lightweight anomaly detection technique for low-resource IoT devices: A game-theoretic methodology,” 2016 IEEE Int. Conf. Commun. ICC 2016, 2016, DOI: 10.1109/ICC.2016.7510811.

- [29] V. Arti, “Volu y 2018 REV VIEW ARTI available online at WW national journalof Ad advanced research in Computer Science REVIEW W ON SE Y IN THEE INTERN NET OF T THINGS,” pp. 645–649, 2018.
- [30] Micro Trend, “Smart Yet Flawed: IoT Device Vulnerabilities Explained,” pp. 1–6, 2020, [Online]. Available: <https://www.trendmicro.com/vinfo/us/security/news/internet-of-things/smart-yet-flawed-iot-device-vulnerabilities-explained#newnavmenu-mobile>.
- [31] M. Litoussi, N. Kannouf, K. El Makkaoui, A. Ezzati, and M. Fartitchou, “IoT security: Challenges and countermeasures,” *Procedia Comput. Sci.*, vol. 177, pp. 503–508, 2020, DOI: 10.1016/j.procs.2020.10.069.
- [32] M. Abomhara and G. M. Køien, “Cyber Security and the Internet of Things : Vulnerabilities, Threats, Intruders,” vol. 4, pp. 65–88, 2015, DOI: 10.13052/jcsm2245-1439.414.
- [33] S. Choudhary and N. Kesswani, “A survey: Intrusion detection techniques for internet of things,” *Int. J. Inf. Security. Priv.*, vol. 13, no. 1, pp. 86–105, 2019, doi: 10.4018/IJISP.2019010107.
- [34] M. A. Razzaq, M. A. Qureshi, and S. Ullah, “Security Issues in the Internet of Things (IoT): A Comprehensive Study,” vol. 8, no. 6, 2017.
- [35] H. U. & H. B. P. TARIQAHMAD SHERASIYA, “a Survey: Intrusion Detection System for the Internet of Things,” *Int. J. Comput. Sci. Eng.* , vol. 5, no. 2, pp. 91–98, 2016, [Online]. Available: http://www.iaset.us/view_archives.php?year=2016&id=14&jtype=2&page=2.
- [36] A. B. Mohamed, N. B. Idris, and B. Shanmugum, “A brief introduction to intrusion detection system,” *Commun. Comput. Inf. Sci.*, vol. 330 CCIS, pp. 263–271, 2012, DOI: 10.1007/978-3-642-35197-6_29.
- [37] N. A. Alrajeh, S. Khan, and B. Shams, “Intrusion Detection Systems in Wireless Sensor Networks : A Review,” vol. 2013, 2013, DOI: 10.1155/2013/167575.
- [38] C. Liang et al., “Intrusion detection system for the internet of things based on blockchain and multi-agent systems,” *Electron.*, vol. 9, no. 7, pp. 1–27, 2020, DOI:

10.3390/electronics9071120.

- [39] K. Rajasekaran, “Classification and Importance of Intrusion Detection System,” no. June, pp. 6–10, 2020.
- [40] R. R. Chaudhari and S. P. Patil, “Intrusion Detection System : Classification, Techniques, and Datasets To Implement,” *Int. Res. J. Eng. Technol.*, vol. 4, no. 2, pp. 1860–1866, 2017, [Online]. Available: <https://irjet.net/archives/V4/i2/IRJET-V4I2366.pdf>.
- [41] M. F. Elrawy, A. I. Awad, and H. F. A. Hamed, “Intrusion detection systems for IoT-based smart environments: a survey,” *J. Cloud Comput.*, vol. 7, no. 1, pp. 1–20, 2018, DOI: 10.1186/s13677-018-0123-6.
- [42] S. K. Gautam, H. Om, and K. Dixit, “Intrusion detection system in Internet of things,” *Lect. Notes Networks Syst.*, vol. 82, pp. 65–93, 2020, DOI: 10.1007/978-981-13-9574-1_4.
- [43] K. Yang, J. Ren, Y. Zhu, and W. Zhang, “Active Learning for Wireless IoT Intrusion Detection,” *IEEE Wirel. Commun.*, vol. 25, no. 6, pp. 19–25, 2018, doi: 10.1109/MWC.2017.1800079.
- [44] H. Li, K. Ota, and M. Dong, “Learning IoT in Edge : Deep Learning for the Internet of Things with Edge Computing,” no. January 2018.
- [45] Z. M. Fadlullah et al., “State-of-the-Art Deep Learning: Evolving Machine Intelligence Toward Tomorrow’s Intelligent Network Traffic Control Systems,” *IEEE Commun. Surv. Tutorials*, vol. 19, no. 4, pp. 2432–2455, 2017, DOI: 10.1109/COMST.2017.2707140.
- [46] MANOJ KUMAR PUTCALA B.E., “Deep Learning Approach for Intrusion Detection System (Ids) in the Internet of Things (IoT) Network Using Gated Recurrent Neural Networks (Gru),” Thesis, vol. 1, no. 1, pp. 1188–1197, 2017, [Online]. Available: <https://osf.io/nf5me%0Ahttp://dx.doi.org/10.1016/j.tree.2015.01.012%0Ahttps://www.tandfonline.com/doi/full/10.1080/1047840X.2017.1373546%0Ahttp://dx.doi.org/10.1016/j.lindif.2016.07.011%0Ahttp://dx.doi.org/10.1016/j.paid.2017.06.011%0Ahttp://programme.exo>.
- [47] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, “Convolutional neural networks:

- an overview and application in radiology,” *Insights Imaging*, vol. 9, no. 4, pp. 611–629, 2018, DOI: 10.1007/s13244-018-0639-9.
- [48] X. W. Chen and X. Lin, “Big data deep learning: Challenges and perspectives,” *IEEE Access*, vol. 2, pp. 514–525, 2014, DOI: 10.1109/ACCESS.2014.2325029.
- [49] J. Urbánek, K. Brabec, L. Dušek, I. Holoubek, J. Hřebíček, and M. Kubásek, “Artificial Neural Networks – ICANN 2010,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6354, no. PART 3, pp. 483–488, 2010, doi: 10.1007/978-3-642-15825-4.
- [50] D. C. Cireşan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, “Flexible, high performance convolutional neural networks for image classification,” *IJCAI Int. Jt. Conf. Artif. Intell.*, no. July, pp. 1237–1242, 2011, DOI: 10.5591/978-1-57735-516-8/IJCAI11-210.
- [51] Y. Chen, Y. Zhang, and S. Maharjan, “Deep Learning for Secure Mobile Edge Computing,” *arXiv*, pp. 1–7, 2017.
- [52] E. De Coninck, T. Verbelen, B. Vankeirsbilck, S. Bohez, P. Demeester, and B. Dhoedt, “Distributed neural networks for Internet of Things : the Big-Little approach,” pp. 1–9.
- [53] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” *Adv. Neural Inf. Process. Syst.*, vol. 4, no. January, pp. 3104–3112, 2014.
- [54] A. M. and G. H. Alex Graves, “Speech Recognition with Deep Recurrent Neural Networks, Department of Computer Science, University of Toronto,” *Dep. Comput. Sci. Univ. Toronto*, vol. 3, no. 3, pp. 45–49, 2013, [Online]. Available: <https://ieeexplore.ieee.org/stampPDF/getPDF.jsp?tp=&arnumber=6638947&ref=aHR0cHM6Ly9pZWVleHBsb3JlLmlZlZWUub3JnL2Fic3RyYWN0L2RvY3VtZW50LzY2Mzg5NDc/Y2FzYV90b2tlbj1OQUo1VFJxWk5JRUFBUFB0mtPZmdDbS00NGhqaGI2N3dMd2JrU3lSaEdJREhBWnpMSkxoT201Um5YMXR0S0poUDAzM2hkbt>.
- [55] A. Graves, “Towards End-to-End Speech Recognition with Recurrent Neural Networks,” vol. 32, 2014.
- [56] M. Hermans and B. Schrauwen, “Training and analyzing deep recurrent neural networks,”

- Adv. Neural Inf. Process. Syst., pp. 1–9, 2013.
- [57] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, “How to construct deep recurrent neural networks,” 2nd Int. Conf. Learn. Represent. ICLR 2014 - Conf. Track Proc., no. December 2014.
- [58] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, DOI: 10.1038/nature14539.
- [59] J. Heaton, “Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning,” *Genet. Program. Evolvable Mach.*, vol. 19, no. 1–2, pp. 305–307, 2018, DOI: 10.1007/s10710-017-9314-z.
- [60] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, “Deep learning for IoT big data and streaming analytics: A survey,” *IEEE Commun. Surv. Tutorials*, vol. 20, no. 4, pp. 2923–2960, 2018, DOI: 10.1109/COMST.2018.2844341.
- [61] G. E. Hinton, “A practical guide to training restricted Boltzmann machines,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7700 LECTU, pp. 599–619, 2012, DOI: 10.1007/978-3-642-35289-8_32.
- [62] U. Fiore, F. Palmieri, A. Castiglione, and A. De Santis, “Network anomaly detection with the restricted Boltzmann machine,” *Neurocomputing*, vol. 122, pp. 13–23, 2013, DOI: 10.1016/j.neucom.2012.11.050.
- [63] M. Mayuranathan, M. Murugan, and V. Dhanakoti, “Best features based intrusion detection system by RBM model for detecting DDoS in cloud environment,” *J. Ambient Intell. Humanize. Comput.*, no. 0123456789, 2019, DOI: 10.1007/s12652-019-01611-9.
- [64] G. E. Hinton, S. Osindero, and Y.-W. Teh, “2006 Dbn,” *Neural Comput.*, vol. 18, pp. 1527–1554, 2006.
- [65] H. F. Nweke, Y. W. Teh, M. A. Al-garadi, and U. R. Alo, “Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges,” *Expert Syst. Appl.*, vol. 105, pp. 233–261, 2018, DOI: 10.1016/j.eswa.2018.03.056.

- [66] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "A survey on deep learning for big data," *Inf. Fusion*, vol. 42, no. October 2017, pp. 146–157, 2018, DOI: 10.1016/j.inffus.2017.10.006.
- [67] A. Khan, "Intro to Deep Neural Networks," no. August 2016, DOI: 10.13140/RG.2.2.17217.15200.
- [68] I. Goodfellow et al., "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020, DOI: 10.1145/3422622.
- [69] G. A. Networks, "A Gentle Introduction to Generative Adversarial Networks (GANs) What Are Generative Models ?" pp. 1–27, 2021.
- [70] L. I. Kuncheva, *Ensemble Feature Selection*. 2014.
- [71] B. Marr, "Deep Learning Vs Neural Networks - What ' s The Difference ?," pp. 1–7, 2019, [Online]. Available: <https://bernardmarr.com/default.asp?contentID=1789>.
- [72] L. C. Nguyen and H. Nguyen-Xuan, "Deep learning for computational structural optimization," *ISA Trans.*, vol. 103, no. XXXX, pp. 177–191, 2020, DOI: 10.1016/j.isatra.2020.03.033.
- [73] J. Schmidhuber, "Deep Learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015, DOI: 10.1016/j.neunet.2014.09.003.
- [74] Z. Ahmad et al., "Anomaly detection using deep neural network for IoT architecture," *Appl. Sci.*, vol. 11, no. 15, 2021, DOI: 10.3390/app11157050.
- [75] Y. Fu, Z. Yan, J. Cao, O. Koné, and X. Cao, "An Automata Based Intrusion Detection Method for Internet of Things," *Mob. Inf. Syst.*, vol. 2017, pp. 6–10, 2017, DOI: 10.1155/2017/1750637.
- [76] B. Susilo and R. F. Sari, "Intrusion detection in IoT networks using deep learning algorithm," *Inf.*, vol. 11, no. 5, 2020, DOI: 10.3390/INFO11050279.
- [77] A. Verma and V. Ranga, "Machine Learning-Based Intrusion Detection Systems for IoT Applications," *Wirel. Pers. Commun.*, vol. 111, no. 4, pp. 2287–2310, 2020, DOI: 10.1007/s11277-019-06986-8.
- [78] K. V. V. N. L. Sai Kiran, R. N. K. Devisetty, N. P. Kalyan, K. Mukundini, and R. Karthi,

- “Building an Intrusion Detection System for IoT Environment using Machine Learning Techniques,” *Procedia Comput. Sci.*, vol. 171, no. 2019, pp. 2372–2379, 2020, DOI: 10.1016/j.procs.2020.04.257.
- [79] C. Liang, B. Shanmugam, S. Azam, M. Jonkman, F. De Boer, and G. Narayansamy, “Intrusion Detection System for Internet of Things based on a Machine Learning approach,” *Proc. - Int. Conf. Vis. Toward. Emerg. Trends Commun. Networking, ViTECoN 2019*, pp. 1–6, 2019, DOI: 10.1109/ViTECoN.2019.8899448.
- [80] J. Johnson, “What ’ s a Deep Neural Network ? Deep Nets Explained What is a deep neural network ? Improving accuracy : The black box problem About the author,” *Bmc Blogs*, pp. 1–3, 2020, [Online]. Available: <https://www.bmc.com/blogs/deep-neural-network/>.
- [81] W. Does, D. Neural, N. Mean, T. Explains, and D. Neural, “What Does Deep Neural Network Mean ?” pp. 1–7, 2021.