T.C.

ALTINBASUNIVERSITY
Institute of Graduate Studies
Information Technologies

DATA MANAGEMENT SYSTEM IN IOTS USING DEEP LEARNING TECHNIQUE

Amer BENAROUS

Master of Science

Supervisor

Prof. Dr. Osman Nuri UÇAN

Istanbul, 2021

DATA MANAGEMENT SYSTEM IN IOTS USING DEEP LEARNING TECHNIQUE

by

Amer MOHAMED ALI BENAROUS

Information Technologies

Submitted to the Institute of Graduate Studies

In partial fulfillment of the requirements for the degree of

Master of Science

ALTINBAŞ UNIVERSITY

2021

The thesis titled “DATA MANAGEMENT SYSTEM IN IOTS USING DEEP LEARNING TECHNIQUE” prepared and presented by “Amer BENAROUS” was accepted as a Master of Science Thesis in Information Technologies.

_____

Prof. Dr. Osman Nuri UÇAN

Supervisor

Thesis Defense Jury Members:

Prof.Dr. Osman Nuri UÇAN	School of Engineering and Architecture, Altinbas University	_____
Asst. Prof.Dr. Sefer KURNAZ	School of Engineering and Architecture, Altinbas University	_____
Prof.Dr. Mesut RAZBONYAL.	School of Engineering and Natural Sciences, Maltepe University	_____

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Approval Date of Institute of Graduate Studies:
____/____/____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Amer Mohamed Ali Benarous

ACKNOWLEDGEMENTS

The first and most important thing I would say is that I am extremely grateful to Allah (the Almighty) for providing me with the mental power, health, and stamina, as well as the guiding knowledge and abilities that have enabled me to complete this research. Thanks again to my thesis supervisor Professor. Doctor Osman Nuri Uçan for his guidance and assistance during the writing process. I wrote this thesis in remembrance of my father, who was a great inspiration to me and once said, "Whatever you accomplish in your life will be the best present I could ever get." Many thanks go out to my mother, who supported me with gentle and immaculate love, as well as to my other half, "my wife," who supported and stood by me throughout the crisis, and to the best smiles in my life, "my children." I'd also like to thank all of my friends who shared their experiences with me throughout the crisis, during the days of the COVID19 virus, with bitterness and sweetness, and I'd like to thank my mother for her support.

My brother "FATHE ALTRHUNEE" deserves a special thank you.

ABSTRACT

DATA MANAGEMENT SYSTEM IN IOTS USING DEEP LEARNING TECHNIQUE

Benarous, Amer

M.Sc., Computer Engineering, Altınbaş University,

Supervisor: Professor Dr. Osman Nuri UCAN

Date: December/2021

Pages: 60

Nowadays, cyberattacks occurring and the variety, size and intensity of cyberattacks are increasing. In this thesis the deep learning neural network based on the LSTM method is used to prediction of Intrusion detection system in Internet of Things. For evaluation of the results we used the results with Sensitivity, Specificity, Accuracy, Precision, Recall and F1 Score. Using the selected features, a classification methodology is proposed. As a result, an average accuracy of 92.94% was achieved by using the proposed model with 41 features. Also the best result for the Sensitivity and Recall obtained from CNN method and it is 94.30% and 94.30% respectively. For Specificity, Accuracy, Precision and F1 Score the Ensemble method has highest percentage and it has 99.47%, 93.43%, 99.86% and 95.75% respectively.

Keywords: Deep Learning, Classification, Intrusion Detection System

TABLE OF CONTENTS

Pages

ABSTRACT ..... **vi**

LIST OF FIGURES ..... **v**

LIST OF TABLES ..... **vi**

LIST OF ABBREVIATIONS ..... **vii**

1. INTRODUCTION **1**

1.1 BACKGROUND 1

1.2 INTERNET OF THINGS (IOTS) 3

1.3 HISTORY OF IOT 4

1.4 APPLICATION AREAS 4

1.4.1 Smart City Applications 4

1.4.2 Smart Environmental Applications 4

1.4.3 Smart Home Applications 4

1.4.4 Procurement Applications 5

1.4.5 Smart Livestock Applications 5

1.5 IMPORTANT CONCEPTS AND SUBTITLES 5

1.5.1 Data Analysis Architecture 5

1.5.2 Data Collection Architecture 5

1.5.3 Sensor Technologies 6

1.5.4 IoT Data Lifecycle 6

1.6 PROS AND CONS OF INTERNET OF THINGS 11

1.7 IOT AND ITS APPLICATION IN INDUSTRY 11

1.7.1 Remote Access 12

1.7.2 Real-Time Monitoring 13

1.7.3 Predictive Maintenance 14

1.8 IOT SECURITY CHALLENGE 15
1.9 ORGANIZATION OF THE THESIS 15
2. LITERATURE REVIEW AND MATHEMATICAL REPRESENTATION 17
2.1 LITERATURE REVIEW 17
2.2 SELECTED DATA SETS FOR THE RESEARCH 22
2.2.1 Institutional Data Set 23
3. METHODOLOGY 25
3.1 MACHINE LEARNING TECHNIQUES FOR ANOMALY DETECTION 25
4. EXPERIMENTAL RESULT 32
4.1 DATASET 33
4.2 PERFORMANCE METRICS 36
4.3 RESULTS 39
4.4 DECISION TREE 43
4.5 ENSEMBLE 45
4.6 CNN RESULT 47
4.7 COMPARISION BETWEEN THE RESULTS ……..……47
5. CONCLUSION AND FUTURE WORKS 48
5.1 CONCLUSION 48
5.2 FUTURE WORK 48
REFERENCES 49

LIST OF FIGURES

Pages

Figure 1.1: IoT Architecture [7]……..3

Figure 1.2: Data Lifetime And Data Management In Internet Of Things [15]..... 7

Figure 2.1: Logical Topology Of An Enterprise Network [8] 17

Figure 2.2: Methods And Attack Mechanisms Used To Classify[9]…….18

Figure 3.1: Artificial Intelligence And Its Subdomains [31] 25

Figure 3.2: Machine Learning Methods For Anomaly-Based Intrusion Detection. 26

Figure 3.3: Flowchart Of Proposed Method 31

Figure 4.1: Standard Feed-Forward Neural Net. 35

Figure 4.2: Confusion Matrix Representation 38

Figure 4.3: Training Step 42

Figure 4.4: Network Analyzer 43

Figure 4.5: Layers Information 43

Figure 4.6: Confusion Matrix For Decision Tree 44

Figure 4.7: Confusion Matrix For Ensemble 45

Figure 4.8: CNN Result 46

Figure 4.9: Comparison Between The Results 47

LIST OF TABLES

Pages

Table 4.1: Train Data That Used In The Model..... 39

Table 4.2: Basic Features Of Individual Tcp Connections..... 40

Table 4.3: Content Features Within A Connection Suggested By Domain Knowledge 41

Table 4.4: Traffic Features Computed Using A Two-Second Time Window..... 42

LIST OF ABBREVIATIONS

IoTs	:	Internet of Things
ACCS	:	Australian Cyber Security Center
LSTM	:	Long Short-Term Memory
ROC	:	Receiver Operating Characteristic
HIDS	:	Host-based IDS
NIDS	:	Network-based IDS
LAN	:	Local Area Network
WAN	:	Wide Area Network
ML	:	Machine Learning
KNN		k-Nearest Neighbors
TP	:	True Positive
TN	:	True Negative
FP	:	False Positive
FN	:	False Negative

1. INTRODUCTION

1.1 BACKGROUND

The Internet of Things (IoT) has exploded in popularity in the twenty-first century in the last few years, becoming one of the most extensively utilized and prominent technologies. By using embedded methods, anybody may link common objects such as kitchenette apps, automobiles, regulators, and baby screens to the Internet. As a result, people, processes, and things may communicate with one another in a cohesive manner.

Physical objects can collect and store data with the least amount of human involvement thanks to low-cost computers, the cloud, big data, analytics, and smart phone knowledge. In this highly linked world, digital systems have the ability to maximize, filter, and switch any communication between things that are connected. The physical world comes into contact with and collaborates with the digital world.

Investigation of the transmitted data is subject to a number of restrictions. Regular home-grown internet service providers do not earn time, do not have the practical expertise to monitor their network movement, and smooth so, if the data is encrypted, it will not be clever to recite the data that has been given to their attention. It is possible to utilize more than ten home smart devices that communicate with the Internet at the same time in a typical house, and this number is increasing [1].

Internet of Things is a revolutionary revolt against Internet. It resolves to make them known and to become intelligent. Make or enable others to make choices on their behalf due to their ability

to link facts about themselves. They may modify the data that has been compiled by others. Things, or mechanisms, can be sophisticated facilities. This is the shift that heralds the advent of cloud computing and Internet broadcast capabilities based on IPv6 and its near-limitless processing capability [2][3]. IoT's goal is to make it possible for anything and everything to connect with anyone and everything via any network, route, or facility. [4].

Companies should begin learning and capitalizing on the opportunities created by the IoT. Capturing and analyzing data from scattered connected devices enables the potential to improve operations, create new revenue streams, and reclaim customer service [5]. The Internet of Things also exposes governments to new security holes created by increased network connections and faulty project planning. Progressive attackers have developed the capacity to pivot on other systems [6].

The purpose of this thesis is to create a novel framework and optimal data mining methods for the detection of IDS in IoTs. The proposed framework will be verified using commonly used datasets in this area, and the results will be compared to the best techniques already published. Until now, deep learning applications in other areas of study have shown acceptable results, including image identification, illness detection, and data categorization. Deep learning method optimized for detecting IDS attacks in gas and oil IoTs. In complicated situations, data mining methods have a poor degree of accuracy. However, improved data mining results in an improvement in method accuracy, which is the study's primary innovation.

1.2 INTERNET OF THINGS (IOTS)

Internet of Things (IoT), smartphone-connected systems, and schemes all have an effect on data calm via embedded devices and actuators on machines and extra physical sensors. IoT is expected to grow rapidly in the next years, and this conference will unveil a new facility that enhances consumers' quality of life and business output, GSMA "Related Life." The use of IoT is complicated by a number of technologies. The IoT's basic structure is shown in Figure 1.1, along with many of these technologies:

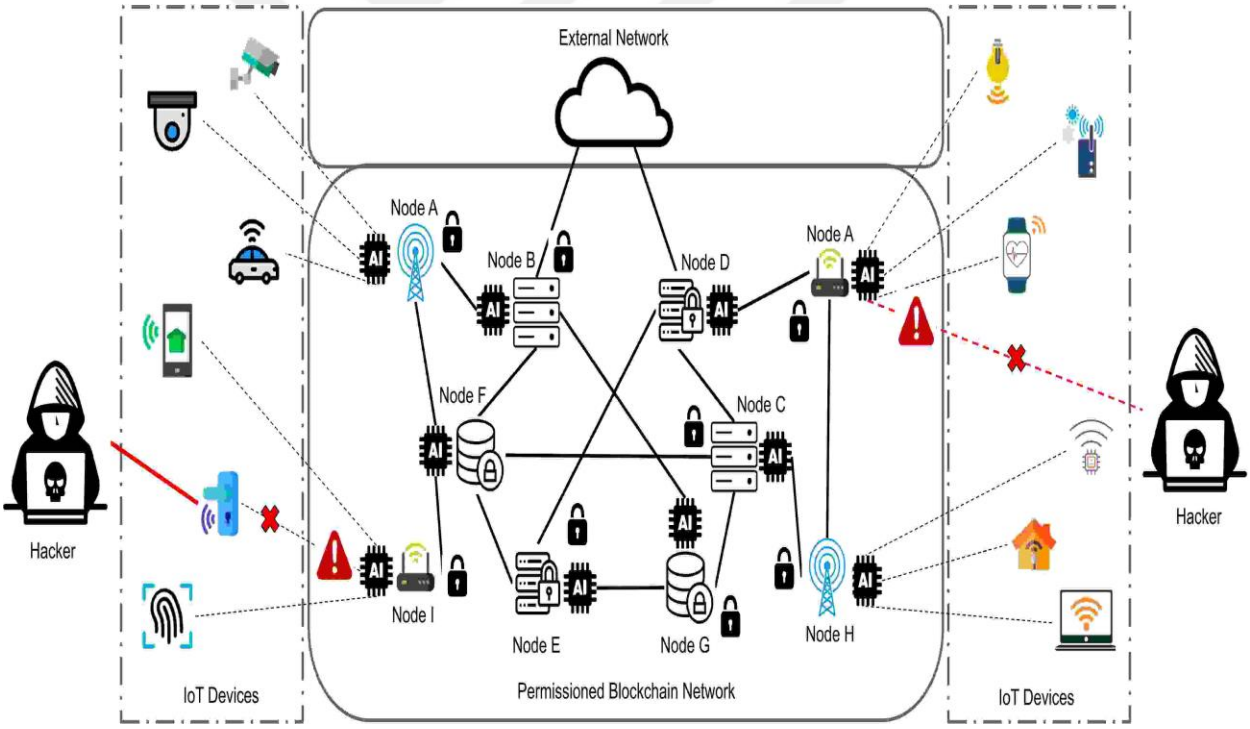


Figure 1.1: IoT Architecture [7]

1.3 HISTORY OF IOT

In 1982, the fundamental concept of an intelligent device network was investigated., when a customized Coca-Cola chocolate appliance at Carnegie Mellon University became the first ARPANET-connected apparatus [8], capable of describing its stock and detecting the temperature of freshly piled beverages [9]. The 1991 essay "The Computer of the Twenty-First Century" by Mark Weiser, as well as educational environments such as UbiComp and PerCom, influenced the present idea of the Internet of Things [10].

1.4 APPLICATION AREAS

1.4.1 Smart City Applications

The usage of these systems allows local governments and city inhabitants to utilize their time and other resources more effectively while working together on problems like traffic, parking and lighting. They also allow for better trash management.

There are applications for generating and disseminating projected traffic densities in the present or future. Car parking apps can inform you of available parking spots in the region you're visiting, as well as their occupancy rates and parking fees. They may offer data on waste management techniques, such as monitoring trash container occupancy rates, optimizing garbage truck collection operations, relocating container points, and resizing container sizes [11].

1.4.2 Smart Environmental Applications

Smart environmental applications are being developed to monitor environmental variables like air pollution, precipitation, dam fullness, and forest fires, and to take immediate action if and when this is required.

1.4.3 Smart Home Applications

Applications for smart homes are being created to allow users to remotely control and monitor the status of services supplied by their houses in areas such as lighting, heating, ventilation, entertainment, and security, among others.

1.4.4 Procurement Applications

It is being created in order to contribute to procurement procedures such as the follow-up of goods whose stock is depleting and whose expiry date is approaching, as well as the restocking of products if this is required.

1.4.5 Smart Livestock Applications

It is being developed in order to monitor environmental factors such as temperature and the presence of hazardous gases in settings where animals are present, among other things.

1.5 IMPORTANT CONCEPTS AND SUBTITLES

1.5.1 Data Analysis Architecture

In The output of IoT initiatives is intended to provide capabilities or value to the business layer of the organization. Analysis systems are created that include the required analysis architecture and methods for transmitting the large volume of sensor data (data) coming from the physical surroundings to the business layer as information after analysis.

1.5.2 Data Collection Architecture

Under this topic, activities such as determining which sensors to collect data from, which door to transmit data to focus employees, information communication norms, and the arrangement of decision systems, are performed in IOT [12].

1.5.3 Sensor Technologies

The sensors to be utilized in the systems may be basic mechanisms such as RFID tags, simple sensors such as a thermometer circuit that displays the ambient temperature at certain times, or decision-making and operating systems such as a camera that focuses on motion.

1.5.4 IOT Data Lifecycle

Figure 1 depicts the data retention period for an Internet of Things (IoT) system. Starting with creation of data and continuing through aggregation, transmission and filtering, preprocessing is optional as well as storage and archiving, but all of these steps are essential. Even though data is produced (requested) and consumed (analyzed), querying and analysis are the times when it is produced (requested) and consumed (analyzed). Online, communication-intensive include data gathering, consolidation, filtering, and early processing. [15]. Offline storage is required for many tasks, including thorough preparation, long-term storage and archiving, and in-depth processing and analysis. Figure 1.2 shows how long Internet of Things data lasts and how it is managed.

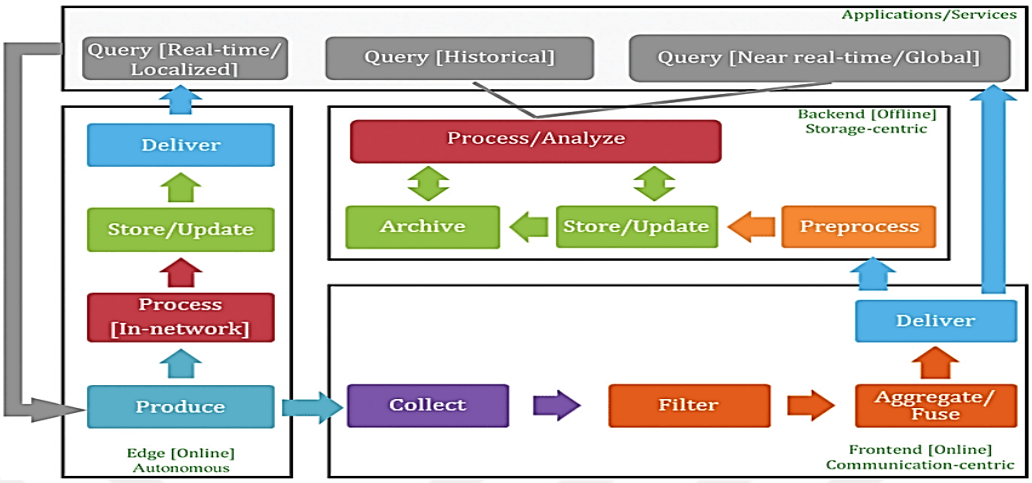


Figure 1.2: Data Lifetime And Data Management In Internet Of Things [15].

When it comes to storage operations, long-term data and availability for updates are concerned, while archiving is concerned with read-only data. Some Internet of Things devices create, process, and store data in the network in order to provide localized and real-time services, and the edges that merge processing and storage components may exist as separate entities in the data cycle. The components of IoT data, as well as their lifespan, are detailed in further detail below.

Querying: For the purpose of gaining access to and retrieving data from data-intensive systems, querying is a critical activity. To collect real-time data for temporal monitoring or to get a specific perspective of information stored inside the system, a query can be done in the context of IoT. First scenario is common when a real-time (mainly localized) data request is required. The second example shows how data may be seen from a broader geographic viewpoint and how trends and patterns can be examined in greater depth.

Production: In order to produce something, data must be sensed and transferred within an IoT framework, and interested parties are notified on a regular basis (as in a subscription/notify model), pushed up network to aggregation points, and then to database servers, or responding to queries requesting data gathered from sensors and intelligent objects. Data often timestamped, geotagged, and it may take the structure of fundamental key-value pairs or sophisticated audio/image/video material of various degrees of sophistication.

Collection: Within the Internet of Things, sensors and smart devices may retain data for an indefinite period of time or report it to controlling components. Data collection may occur at network nodes or gateways, and then filtered and processed further, perhaps fusing it into compact forms for efficient transfer. Wireless communication systems such as Zigbee, Wi-Fi, and cellular enable objects to interact with collection points.

Aggregation/Fusion: It's becoming increasingly difficult to send all of the network's raw data out in real time because of increasing data streaming speeds and a limited amount of available bandwidth. Aggregation and fusion methods allow for real-time data summarization and merging, reducing the amount of data that must be kept and sent. [15].

Delivery: Results of these activities may need to be conveyed further up the system, either as final responses or for storage and in-depth analysis, depending on where Internet of Things concentration points or autonomous virtual units filter, collect and maybe process data. Data can be delivered to long-term storage via wired or wireless internet connections.

Preprocessing: Internet of Things will generate data from a variety of sources and in a variety of forms and formats. For some data, preprocessing is required before putting it into storage. This is done to make sure that there aren't any errors, duplicates, or missing data. Data cleaning is a term used to describe this type of preprocessing in data mining. Although schema integration implies all data must be put into a relational (tables) schema, it is actually a more abstract concept of a consistent mechanism for accessing data that does not necessitate a change for each source's data type (s). Data ambiguity or a lack of trust in data sources can be accounted for by applying probabilities at various levels of the schema to IoT data items at this step [16].

Storage/Update-Archiving: As a result of this phase's responsibilities, data must be stored and administered, as well as continuously updated in reaction to new information. To archive data, you store it in a non-critical state for as long as possible without affecting the system's ability to function. Centralized storage is defined by use of storage structures that can adapt to different types of data and frequency at which they are collected. Relational database management systems, which are extensively used, organize data in tables with specified linkages and metadata for future retrieval [17]. Due to its capacity to store huge quantities of data without needing a relational schema or adhering to the stringent consistency requirements associated with relational database systems, Key-value stores based on NoSQL databases are becoming increasingly popular as data storage solutions.[18].

Data can be stored decentralized for autonomous IoT systems, with the items that generate it instead of transmitting it up the system. However, the storage capacity of such things is restricted in comparison to the centralized storage technique because of the limited capabilities of these objects.

Processing/Analysis: There will be a constant flow of information to acquire new insights into previous data, forecast future trends, or identify data anomalies that need more research. Before performing significant operations to filter and clean data, it's probable that task-specific preparation will be needed. When an IoT subsystem is self-contained and doesn't require long-term storage of its data, in-network processing can be used to respond to real-time or localized requests.

The diagram shows that data can flow in three different directions: from autonomous systems within IoT that start with production and continue through data collection and filtering/aggregation/fusion until data delivery to initiate (possibly global or near real-time) queries, or from autonomous systems that start with production and continue through data collection and filtering/aggregation/fusion until data delivery is initiated.. It's important to note that, in light of this life cycle, typical data management systems can only go so far.

1.6 PROS AND CONS OF INTERNET OF THINGS

Any existing technology has not reached its full potential. It always has a void. So, we can say that the Internet of Things is a major technology that can assist other technologies to achieve their full potential.

a) Automation results in task consistency, service quality, and control of day-to-day activities without human involvement. Additionally, machine-to-machine communication aids in the maintenance of process transparency.
b) Machine-to-machine contact improves efficiency by allowing humans to concentrate on other tasks.
c) Along with optimizing energy and resource efficiency, the IoT assists in resolving issues connected with bottlenecks, malfunctions, and system damage.
d) The Internet of Things enables physical objects to remain linked and interact more effectively, resulting in increased quality control.
e) Increased information availability aids in simplifying the decision-making process, thus making life simpler to manage.
f) As more of our daily gadgets, devices, and services become internet-connected, access to far more information becomes possible. It makes it more difficult to keep sensitive data safe from hackers and other unauthorized users.
g) Currently, there is no worldwide standard for IoT interoperability, which makes it difficult for devices from disparate manufacturers to interact.
h) Because the IoT is such a large and varied network, a single software or hardware failure may have catastrophic effects.

i) As the Internet of Things introduces more constant automation, we may witness a decrease in the need for low-skilled workers in the workplace.

j) As our lives grow more reliant on technology, society's fundamental human connection abilities will deteriorate.

Without many of us recognizing it, the Internet of Things has already become a significant part of daily life. As technology advances and develops, the usage of IoT for many of our fundamental interactions will increase. It is entirely up to us to choose how much of our everyday life we are prepared to cede to technology. However, if done properly, it will adapt to our requirements and serve society as a whole.

1.7 IOT AND ITS APPLICATION IN INDUSTRY

Intelligent asset inspection enables you to comprehend the current condition of operation of a piece of contemporary O&C equipment. To do this, businesses must use data from a variety of sources, including aerial surveillance, subterranean surveillance, remote surveillance, remote sensing, SCADA, and manual or visual inspections. This data is then incorporated into the back of the desk, where it is used to store the intelligent data for photogrammetry and thermography. This enables businesses to develop machine learning algorithms and obtain insights to help them make better decisions. For instance, businesses may effectively use digital models such as digital twins to design and manage existing facilities with the assistance of intelligent plant inspection. As an intelligent resource, the inspection software may assist businesses in maintaining an asset's digital twin in the case of model data changes. This makes it helpful for simulating what-if situations and monitoring in real time. Additionally, businesses may perform inspections using the right sensors at the appropriate frequency to maintain the model current. A manufacturing

plant or refinery may also use satellites and drones to map assets and generate a digital depiction of them. Pop-up memory may be used to store data from the internal pipeline data collection system and other sources in order to produce information and feedback for the purpose of detecting abnormalities. Additionally, such a smart asset inspection program may use IT data for supply chain and financial information to improve enterprise-wide operational effectiveness. Then, various approaches based on IoT and AI techniques were used to create a variety of strategies for gas and oil applications:

1.7.1 Remote Access

Regardless of the location or relocation of gasoline tanks, IoT technology and its sophisticated capabilities enable remote access to them. Because IoT is a sensor technology, it links to your smart device through communication protocols that also serve as a platform for remotely monitoring tanks. The Internet of Things is completely built on a sensor-based approach. This data is then transmitted to the cloud platform, where it is stored and processed prior to being made available to users. As a result, it enables remote access for users with defined credentials and customized settings.

1.7.2 Real-time Monitoring

The Internet of Things is a cutting-edge technology that makes use of touchscreen devices to monitor assets in real time. This enables businesses to improve their performance and establish a competitive advantage in the market. As a result, the oil and diesel tank monitoring system is built on IoT technology, which enables a more accurate assessment of the fuel level. Install sensors on the tanks to monitor and communicate the precise level of the tanks to managers through their linked smart devices. Due to its real-time monitoring method, which delivers SMS

alerts and changes to managers immediately, the technology is useful to a broad range of businesses. As a result, the technology allows real-time monitoring of industrial facilities, which enables managers to take appropriate action.

1.7.3 Predictive Maintenance

Business people are progressing in sectors, particularly in the petroleum and natural gas industries, as a result of IoT technology and its newest ideas. The oil and gas sector can now overcome and cover all areas, including supply chain management, tank levels, and more. Everything is feasible because of sophisticated data management and analysis. This enables businesses to monitor predictive maintenance in order to identify and address asset breakdowns and accurate situations. Authorities may use this data to forecast planned maintenance and minimize operating and maintenance expenses. This improves the company's total profitability and enables exponential growth. Intrusion detection is a well-recognized issue in the area of cyber security, and many methods have been attempted to address it. AI's machine learning and deep learning subfields can both aid with this problem's resolution. This chapter provides an overview of intrusion detection systems and the associated machine learning algorithms.

1.8 IOT SECURITY CHALLENGE

The Internet of Things (IoT) has recently been studied. With IoT's huge potential come a slew of issues. The security of Internet of Things (IoT) is a major concern. This article analyzes IoT research development and highlights several security issues that must be addressed. Secure data privacy, confidentiality, integrity, authentication and access control are essential for ensuring data identity and ensuring heterogeneity, scalability and availability. Taking these facts into account, new IoT solutions from the technical, academic, and industrial viewpoints are presented and discussed. It is essential to design and deploy IoT solutions that offer anonymity, confidentiality, and integrity in various contexts. Focusing only on IoT devices does not give a whole understanding of why and what security entails. This vital IoT security consists of several components. Growing attack surface makes IoT security critical. Users and businesses that do not have the means or knowledge to protect their IoT ecosystems exacerbate the dangers. Apart from the dangers themselves, coping with the fallout from IoT hazards is difficult. The IoT is unusual in that it may affect both virtual and physical systems. Due to the Internet of Things Due to the ease with which Cyberattacks may be converted into physical consequences, their results are considerably more unpredictable. This is especially true in the Industrial Internet of Things, where previous hacks had cascading effects. Securing the network and maximizing its security While IoT devices may cause havoc on networks, they also provide a level playing field for users to adopt security measures that protect all connected devices.

1.9 ORGANIZATION OF THE THESIS

In Chapter 1, The fundamentals of Internet of Things (IoT) are covered, as well as the distinctions between the technologies. Chapter 2 contains an overview of the literature as well as a mathematical depiction. A discussion of methodology is provided in Chapter 3, as well as a comparison of the benefits and drawbacks of various methods. The deep learning method that will be utilized in this research is also highlighted in this chapter. It is simulated in MATLAB 2021a package program in Chapter 4, and the experimental findings are presented in this chapter. The last section of Chapter 5 is devoted to the conclusion.

2. LITERATURE REVIEW AND MATHEMATICAL REPRESENTATION

2.1 LITERATURE REVIEW

Intrusion attempts are one of the most serious risks to network security that may occur. The ultimate aim of hackers is to gain complete or at least partial control of the targeted system, and they may use a variety of methods and tricks to achieve this goal. Information security specialists, on the other hand, are charged with the duty of safeguarding the digital assets of the organization. Additionally, the defending side has a slew of network/host security programs, to name a few: IDS/IPS, firewalls, data loss detection and prevention systems, and anti-malware software, among others. Traditionally, intrusion detection systems (IDS) and firewalls, whether independent or integrated, served as the outside perimeter, or border wall, of the protected region. Figure 2.1, a simplified representation of an enterprise network [20] shows an example of this type of network.

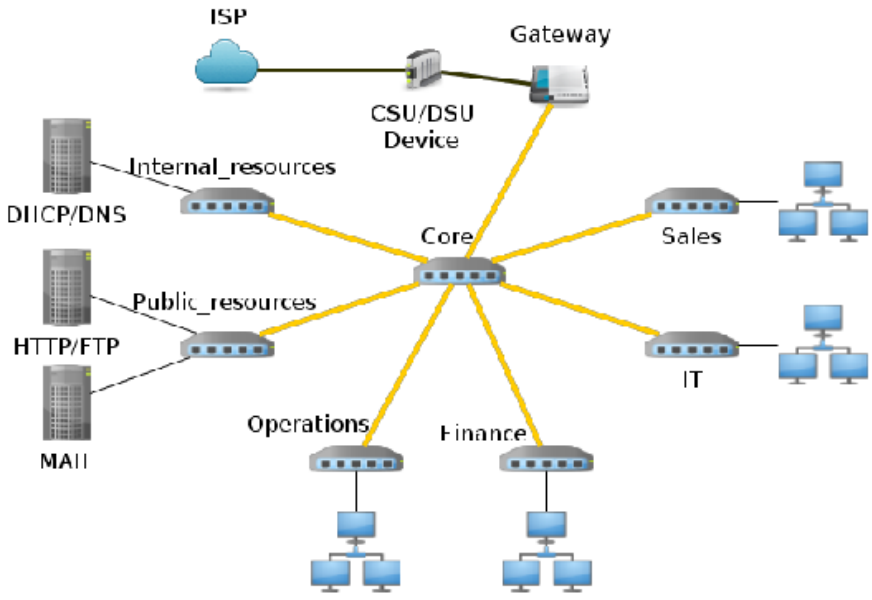


Figure 2.1: Logical topology of an enterprise network [21]

Recent defensive strategy has shifted toward detecting intrusions rather than preventing them, since network assaults have increased in diversity and sophistication, making them more difficult to block at the enterprise's outer perimeter. Additionally, direct blocking of suspicious activity may result in a denial of service attack on genuine traffic as a result of false positives. To further illustrate the idea of IDS, Figure 2.2 presents a taxonomy that categorizes IDS according to data gathering methods and attack detection methodology [22].

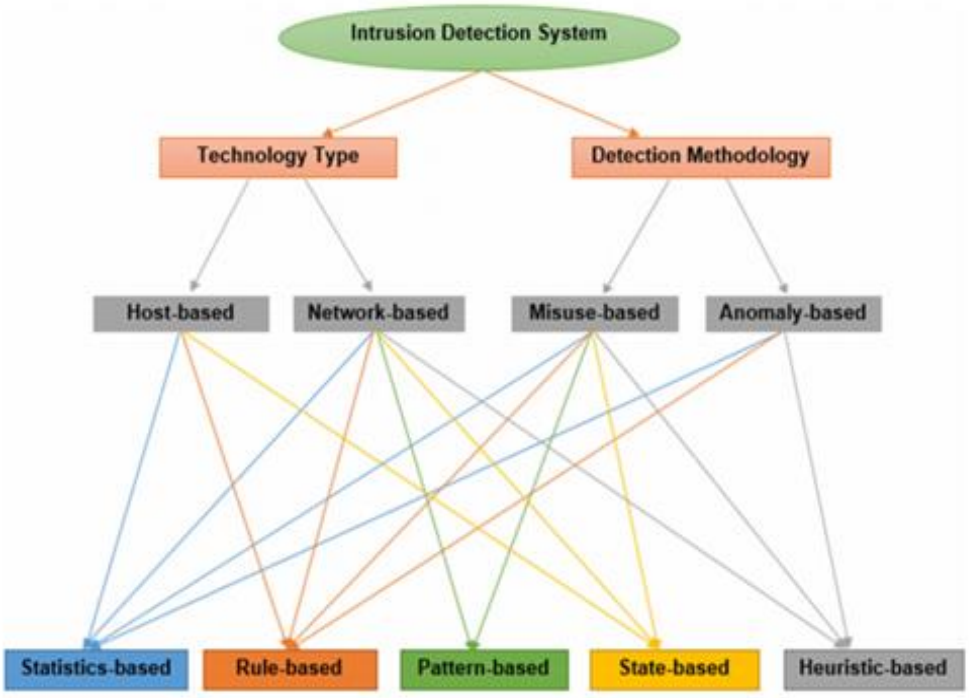


Figure 2.2: Methods And Attack Mechanisms Used To Classify Intrusion-Detection Systems[23]

Host-based IDS (HIDS) examines activity on endpoints, namely hosts, depending on the data collecting method. In other words, System resources including memory, CPU, and disk I/O are monitored by HIDS, as well as programs to look for unusual behavior. To that degree, their jurisdiction extends to the host on which they function. The majority of HIDS search for indications of a danger in locations such as operating system log files or the audit logs of another

piece of software [24]. Alarms are generated based on specified criteria. Due to the fact that they function according to rules, their performance in detecting known intrusion attempts is very good. They are, however, ineffective in detecting new assaults.

Although sensors are used to capture data moving across network in network-based IDS (NIDS), they are analyzed to look for abnormal activity [25]. Such systems may operate independently at the protected network's entry point, capturing and inspecting each incoming and outgoing packet. They may also be relocated to another network section. Local area network (LAN) and wide area network (WAN) traffic may be routed to key nodes through tap devices placed in them for examination. As will be described, NIDS may operate on a rule-based or anomaly-based basis. Both kinds have their advantages and disadvantages.

Hybrid IDS are designed to combine the advantages of HIDS and NIDS. This strategy argues for the deployment of IDS on both hosts and networks to examine local occurrences and analyze network data. This concept may seem reasonable at first. Management of both systems, on the other hand, imposes a greater administrative burden on information security experts and system administrators.

Another distinction is made based on the techniques employed to detect assaults. IDS that are misuse-based detect known threats based on their distinct behavior, or signatures [26]. They protect against well-documented and acknowledged intrusion attempts. The IDS must be trained to perform suitable countermeasures whenever a rule is violated for each attack type. Misuse-based IDS, on the other hand, has a number of drawbacks. They simply lack the capacity to detect unidentified or zero-day threats. Worse still, each attack type must be converted to a rule in order to be detected, implying that effective protection needs frequent updates. Snort,

Suricata, and Zeek (previously known as Bro) are the most commonly used and open-source intrusion detection systems that are based on abuse. This rule produces an alert anytime an attempt is made to connect to SQL Server's tcp port 1433 from any external IP address or port and the payload includes the string "sa" and the intrusion attempt happens more than five times within two seconds. As previously stated, if the same attempt happens once per second (or fewer than five times per two seconds), the misuse-based IDS will miss it and the brute force login attempt will succeed. Due to the ease with which network scanning and packet construction tools are presently available, a competent intruder may easily bypass such limitations [24].

This distinction is established by two methods based on a broad categorization of anomaly-based intrusion detection systems. To begin, models are constructed using statistical methods by conducting statistical tests on past events. Then, for each occurrence, the classification model is compared [25]**.**Second, machine learning techniques, will be described in detail in the next paragraphs.

Machine learning models require training data regardless of the method utilized. There is no exception to this rule in machine learning models used in anomaly-based intrusion detection systems. Public data sets are distinguished from private data sets by the availability to the general public. In the lack of private sector inquiry, this study will only look at well-known public data sets. These are frequently employed by academics and researchers alike for benchmarking purposes.

As part of Defense Advanced Research Projects Agency (DARPA) network security investigation, a lab-generated 1998-1999 data set is used. These protocols generate a lot of network traffic, thus there's a lot of it. DARPA's research from 1998 to 1999 included attack

methods like as probing, rootkits, buffer overflows, and denial of service. In addition to being outdated and lacking contemporary attacks, it has real-world oddities such as a lack of false positives and an outdated network configuration in the lab (MIT Lincoln Laboratory, 1998).

It contains attacks that are distinct from those of its parents, such as smurf and neptune. There are also four different types of attacks classified as DoS, user to root, remote to local, and probing. It is, however, criticized for being very repetitive and for having damaged data [27]. It was suggested that NSLKDD be used to solve these problems [28]. Although it mitigates statistical weaknesses in the data set, NSL-KDD still suffers from the lack of modern attacks and real-world samples.

On the other hand, there are modern public data sets for IDS research created by universities and/or research groups. Recognizing the need for a modern IDS data set, Moustafa and Slay proposed UNSW-NB15 data set [29]. As with earlier examples, it was produced in a lab setting using fake malicious and benign instances generated using a commercial network simulation device. To be more precise, such tools allow the simulation of current attacks by automatically gathering data from Common Vulnerabilities and Exposures (CVE) web sites and then developing and delivering network assaults in response.

Additionally, the data collection includes packet capture (pcap) files of flowing traffic and characteristics derived from raw data. These characteristics are retrieved by the authors using the freely available argues tool and some custom scripts. The Canadian Institute for Cybersecurity has also compiled a collection of available data sets for IDS research (CIC). These are designated using the CICIDS format, for example, CICIDS2017 [30].

CICIDS data sets are generated in a laboratory setting. The researchers generated twenty-five users with mimicked human behavior using an abstraction technique called B-Profile. Eleven criteria are specified to ensure the data set's reliability and completeness, including a fully configured network, attack variety, heterogeneity, and metadata availability [31].

Thirdly, Stratosphere Lab is another organization that creates data sets. We generate and collect fictitious traffic with harmful, background, and normal profiles. Although their primary emphasis is on malware and distributed denial-of-service assaults, current attack cases may be found among these data sets.

The list of publicly accessible IDS data sets is not exhaustive. Although not as well-known as the previously mentioned examples, the DEFCON data set by the Shmoo group (2000–2002), the Lawrence Berkeley National Laboratory (LBNL) data set (2004–2005), the CDX data set (2009) by the United States Military Academy, the Kyoto data set (2009) by Kyoto University, and the Twente data set (2009) by the University of Twente are additional publicly available examples of IDS data sets..

2.2 SELECTED DATA SETS FOR THE RESEARCH

As previously stated, few studies using real-world data exist in IDS research, with a few notable exceptions [24]. Majority of prior work has been conducted on public or private data sets generated in lab settings using network traffic simulators. However, drawbacks associated with the use of artificially generated data sets may be described as follows: (1) a lack of real-world behavior profiling for a particular business network; (2) a model's lack of representation of real-world risks.

The first argument might be stated as follows: each company has its own unique approach to service provision and network architecture design. Organization A may offer service X, but not

B or C. However, since they are produced in a controlled setting, artificially manufactured data sets attempt to cover all possible attack types. Thus, by training a model on a simulated data set and deploying it on one's own corporate IDS, one may train the model against attack types that are difficult to disclose. In summary, corporate IDS models must be trained on real-world traffic in order to increase their dependability. Not only harmful simulated traffic, but even regular traffic does not accurately represent the behavior of particular business users, since it is generated by simulators. Second, attack simulators make every effort to generate attacks that exploit known vulnerabilities. However, since there will always be other methods to exploit a vulnerability, in a real-world situation, other payloads may be developed. For example, Hulk is a well-known and recent example of a distributed denial of service (DoS) attack at application layer. And most often used implementation is Shteiman's Python script, which is also freely accessible online. However, even within GitHub's public repositories, many variations of the original script are available. While the wildness of the actual world may be addressed with all of these variations, artificial settings cannot.

2.2.1 Institutional Data Set

Due to the reasons mentioned in the preceding chapter, this research will examine network data gathered from institutional networks. The collection was performed using the tcpdump program from the organization's pop-up network interface. Sessions started from the outside were prioritized to reduce complexity on the premise that an attack from the outside is more likely. The simplified environment in which the data is gathered is shown in Figure 6. Session/flow data was retrieved at the packet level. Following that, the data sets were jumbled and randomly sampled to create a representative and understandable collection. There are 100,000 flows/sessions in the final data collection.

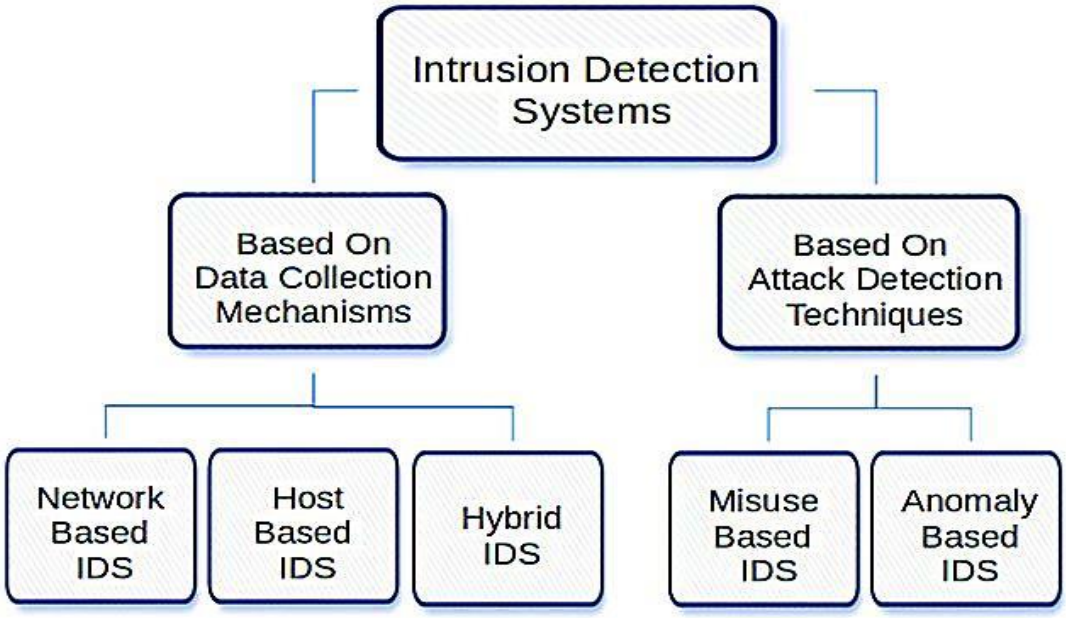


Figure 2.3: Intrusion Detection System [32]

3. METHODOLOGY

3.1 MACHINE LEARNING TECHNIQUES FOR ANOMALY DETECTION

Machine learning (ML) exists at the nexus of many scientific disciplines, including computer sciences, statistics, and natural sciences. It is one of the most popular areas of study, with many cutting-edge applications and potential career possibilities. Information security, and particularly network security, is no exception to this trend in which machine learning offers answers to many long-standing issues. It was first described as a "branch of research that enables computers to learn without being explicitly taught" [33]. The central premise of machine learning is that computers are capable of doing human-level activities. Due to their popularity, the words artificial intelligence, machine learning, and deep learning may be used synonymously. As previously acknowledged by earlier research [34][35][36][37].

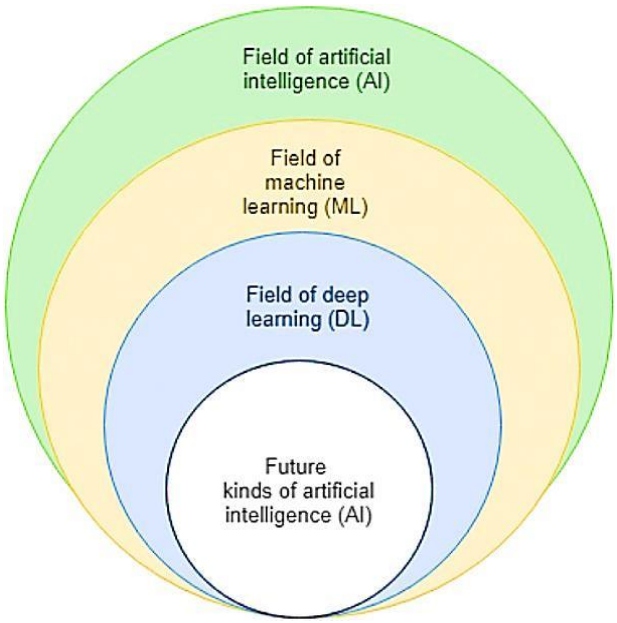


Figure 3.1: Artificial Intelligence And Its Subdomains [38]

Because the emphasis of this work is on anomaly detection using machine learning, prior research will be assessed in that light. There is no agreement on how machine learning techniques for IDS should be classified. Anomaly-based intrusion detection uses a variety of machine learning algorithms, as seen in Figure 3.2 [24][39][40].

Providing samples from each and every method is far beyond the scope of this study. However, prominent sub-methods and related work will be evaluated.

Providing samples from each technique would significantly exceed the scope of this research. Significant sub-methods and associated work, on the other hand, will be assessed.

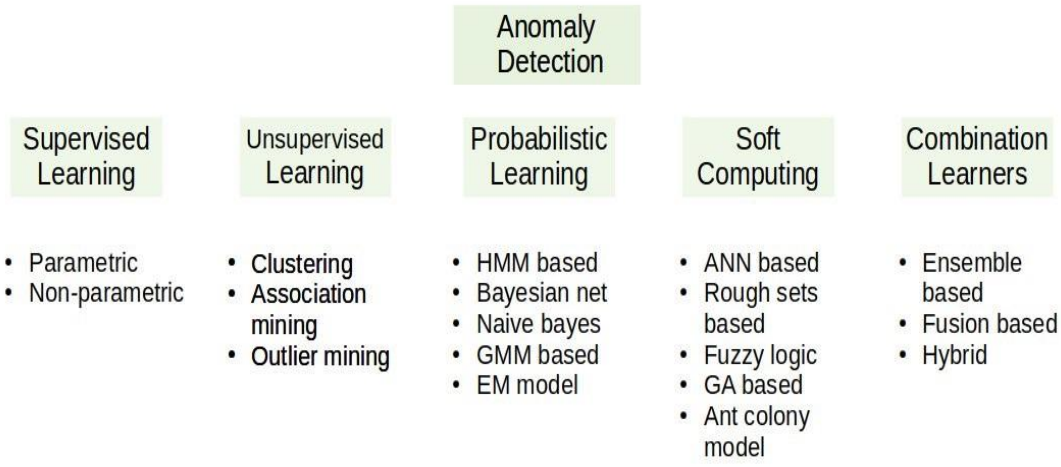


Figure 3.2: Machine Learning Methods For Anomaly-Based Intrusion Detection

k) **Supervised learning** presupposes the data set has a distinction indicating whether a flow/packet is under attack or not. Or, more frequently, the type of assault that occurred. Then, using the labeled data set, a classifier is trained. Following that, new instances are checked against the classifier to see if they belong to a class. The model's success is determined by the proportion of correct predictions to false positives and negatives. Depending on whether or not the parameters are fixed in size, there are two types of supervised learning procedures. In each case, parametric models summarize data by utilizing a preset set of parameters, regardless of sample size. Non-parametric models, on the other hand, have parameters, but the sizes of the data sets might vary them [35]. K-nearest neighbors (KNN), decision trees, and support vector machines (SVMs) are all machine learning approaches in the second category. Logistic regression and naive bayes fall into the first category.

Using supervised learning for anomaly-based IDS, ubiquitous work is possible. To conclude, Chikrakar and Chuanhe developed a model in which comparable data samples are clustered using k-nearest neighbor methods and then categorized using SVM classifiers as output clusters. This model employs KNN and SVM techniques and achieves a high level of accuracy. However, the authors recognized the complexity problem that may arise as the data collection becomes bigger [41]. For detection and false positive rate, Singh et al. advocated combining use of naive bayes and decision tree approaches [42]. It showed a satisfactory false positive and detection rate, however the prediction of distant to local attack classes need refinement. Peddabachigari et al. developed a model that incorporates decision tree and support vector machine techniques. They began by running the data through a decision tree model to generate first features.

After that, created features were submitted to an SVM model together with the original ones to obtain the final classification [43]. This model performed well on the KDD cup dataset. On the other hand, no significant improvement was seen when compared to SVM-only model results.

l) **Unsupervised learning** the benefit of models is that they may be constructed without labeled data sets. Due to the fact that labeling processes often need human knowledge and frequently include manual procedures, this advantage becomes even more critical. Unsupervised learning models are classified into three categories. To begin, techniques based on clustering do not need classes other than labels. The model distinguishes classes based on the training data set. Second, association mining aims to discover often occurring occurrences. This approach effectively employs statistical techniques in that it estimates two factors while creating an association rule: support and confidence. Following that, inferences about future occurrences may be drawn utilizing the associations. Thirdly, outlier mining identifies patterns that do not fit the bulk of the data. In contrast to the first sub-method of clustering, outliers are examples that do not fit into any of the clusters. The critical step in outlier mining is defining what defines an outlier. This is accomplished via the use of statistical and other machine learning techniques. When it comes to previous research, a model built on five different clustering algorithms and tested against the efficacy of misuse-based models was presented by Syarif et al. Four of their models performed better than the competition's equivalent models. [44]. Zhang and Zulkernine proposed an outlier mining approach combined with a random forest method Because the model is not reliant on labeled data or the training environment of a specific network, it is said to be more flexible and perform better than

supervised models. However, the authors highlighted a drawback of unsupervised models: performance degradation in the presence of a large rate of assault occurrences, such as a DOS attack. To address this issue, it is possible to combine the suggested model with misuse-based models.

a) **Probabilistic learning** Models offer estimates for new occurrences that are subject to randomness and other forms of probabilistic uncertainty [24]. The most distinguishing characteristic of probabilistic models is their ability to adjust prior estimates in response to new evidence acquired from training data. Tapiador et al. developed a model based on HTTP protocol traffic to specify its states and transitions between them[45]. During the training phase, it calculated the probability of symbols originating from a Markovian source using simulated attack-free traffic. The findings indicated that the detection and false positive rates were satisfactory. However, the authors highlighted the model's disadvantage of regular retraining. Aung and Oo proposed a model based on the algorithm for frequent pattern development [46]. This method is a variant of the association algorithm and is often used to extract item sets from huge data sets. After then, the authors employed item sets to create frequent pattern trees for spotting traffic anomalies. According to the findings, the model was able to effectively control traffic and spot anomalies in real time.

b) **Soft computing** In contrast to traditional computing, models stress the use of unpredictability in communication. This may be accomplished via the use of probabilistic models and fuzzy sets. Soft computing techniques include things like fuzzy sets, rough sets, ant colony algorithms, artificial immune systems, and genetic algorithms. In a study by Hamamoto et al., an evolutionary algorithm was combined with fuzzy logic to create a

new model [47]. Amin et al. proposed and compared a model based on rough sets theory to four existing machine learning methods [48]. The authors highlighted the findings demonstrating that a model based on rough sets beats the other four when evaluated on the KDD Cup data set.

m) **Combination learners** as the name suggests, this method integrates many previously mentioned learning strategies. The primary objective is to mitigate the shortcomings of each model in an effort to raise the general ability of the mutual prototype [49]. The bagging and boosting ensemble techniques were compared in this model .As the basis classifier; the J48 decision tree classifier is employed. The NSL-KDD data set was used to assess the results, and bagging method's classifier was shown to be more accurate in terms of detection and false positive rates.

One conclusion that could be drawn from this argument is that anomaly-based intrusion detection machine learning approaches have no inherent advantages or disadvantages. Researchers may be influenced by a variety of factors when choose which technique to use. The amount of computer power available, the availability of field experts to manually intervene in the data, and the amount of data to be processed in a second are some of the issues a researcher should consider to choose the method of choice. To summarize, the effectiveness of a particular strategy should be assessed on a case-by-case basis.

The flow chart of proposed method is shown in figure 3.3.

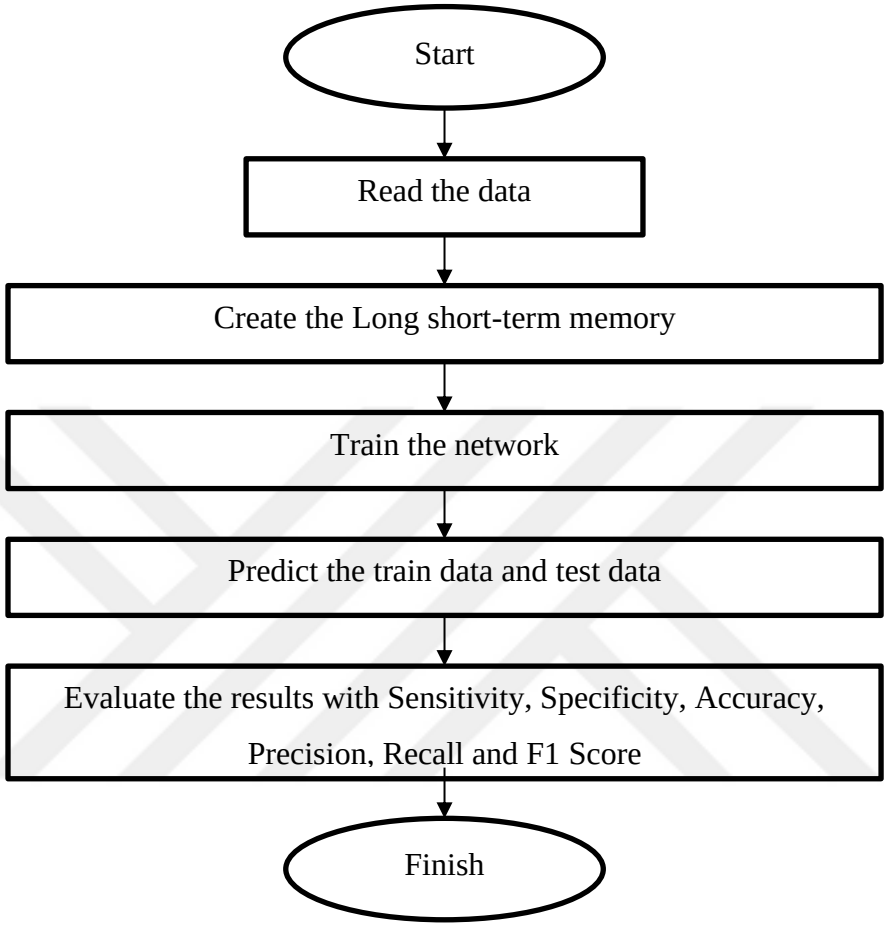


Figure 3.3: Flowchart Of Proposed Method

First, we read the data and save it in the given variable, which will be used in subsequent stages. After that, we construct a lengthy short term memory that will be used throughout the network's training process. LSTM networks are trained for about 100 epochs after they are created in order to get the lowest loss value that is acceptable for estimation of the weights in the network. As soon as the training procedure is over, we utilize the evaluation results technique to forecast the train data and the test data for the next day. When evaluating the results and drawing conclusions, we specifically used Sensitivity and Specificity findings in conjunction with Accuracy, Precision, Recall, and F1 Score.

In order to evaluate the method, Deep Learning Toolbox is utilized; an example of deep learning code that has been created by Matlab2021a is given below in the following line:

```
layers = [ ...
    sequenceInputLayer(featureDimension)
    lstmLayer(numHiddenUnits,'OutputMode','sequence') %
(LSTM) Long short-term memory
    fullyConnectedLayer(50)
    dropoutLayer(0.5)
    fullyConnectedLayer(numResponses)
    regressionLayer];
    maxEpochs = 100;
    miniBatchSize = 5;

options = trainingOptions('adam', ...
    'MaxEpochs',maxEpochs, ...
    'MiniBatchSize',miniBatchSize, ...
    'InitialLearnRate',0.01, ...
    'GradientThreshold',1, ...
    'Shuffle','never', ...
    'Plots','training-progress',...
    'Verbose',0);
```

For normalization of the data following equation is used:

$$NormalizedData = \frac{X_{test} - \mu}{\delta} \tag{3.1}$$

In this equation $NormalizedData$ is the normalized test or normalized train data that use in training of the deep learning method. X_{test} is the test or train data. As seen below, μ is the average value of each column of data.

$$\mu = \frac{Sum\ of\ term}{Number\ of\ terms} \tag{3.2}$$

δ the data standard deviation is given as::

$$\delta = \frac{\sum_{i=1}^{N}(X_i - \bar{X})^2}{N-1} \tag{3.3}$$

4. EXPERIMENTAL RESULT

4.1 DATASET

Using the IXIA PerfectStorm tool from Australian Cyber Security Center (ACCS) Cyber Range Lab, raw network packets in UNSW-NB 15 dataset were created in order to give a combination of current actual and modern routine business synthetic attack behavior.

The dataset's URL is https://kdd.ics.uci.edu/databases/kddcup99/task.html, and we utilized the KDDCup-'99' dataset for our project. It is necessary to capture 100GB of raw traffic using the tcpdump program (like a pcap file).

Fuzzier, Analysis, Backdoor, DoS, Exploit, Generic, Recon, Shellcode, and Worm are all included in this dataset. Attacks using Fuzzier, Analysis, and Backdoor are the most common. For a total of 49 categorized functions, Argus, Bro-IDS, and other software tools are employed. Twelve algorithmic frameworks are developed. In a wide range of systems, including network systems, the Internet of Things (IoT), SCADA, the Industrial Internet of Things, and Industry 4.0, they have published numerous articles on the development of intrusion detection, network forensics, and data security techniques.

In this research, a novel framework for detecting IDS in IoTs was created. The proposed system utilizes autoencoders for feature selection and long short-term memory (LSTM) method for autoencoder training. First auto-encoder trained on input features was used to extract features. The LSTM was used to train the first auto-encoder, which was responsible for extracting sensitive features from the input dataset. The retrieved characteristics were then connected to a second auto-encoder and trained using LSTM. The second auto-encoder then connected the retrieved characteristics to the softmax, which classified them as normal or abnormal. Conduct

and analyze experiments using the MATLAB R2021a program. Versions of Neural Network Toolbox, Statistics Toolbox, and Machine Learning Toolbox Suggestions for classification. The time series data collected in the previous phase is processed in the first step using the window processing technique. The suggested system offers a system along with an observation window. The window size is set to 60 sample periods, with a 30-second slide sampling interval [50].

Then, using a new method called RBM, the size of the window gathered is reduced. This method summarizes 60 feed samples from the current window by applying the symlets4 or sym4 wavelet to extract the potential energy at level 10. This implies that rather of supplying the full window, just the summary observations are processed. This intended phase enhances the proposed system's performance by decreasing the processing time required to finalize the model [51][52].

Backpropagation may be shown using the network seen in Figure 4.1, which has been replicated with notation from Duda, Hart, and Stork [52]. The inputs x1, x2,..., xd are representations of the data from which we want to learn, and the outputs (z 1), (z 2),..., (z c) are representations of the labels z1, z2,.., zc associated with the input data. For instance, the input might be a vector of gray-scale pixel values from an image, whereas the output could be a binary vector labeling a specific item in the picture. The layer y1,y2,..,yn is referred to as the hidden layer, and the connections between it and the input layer are what will be used to identify characteristics in the input data. Backpropagation begins with a feed-forward pass in which all input values xi are multiplied by their corresponding weights wij, and then the weighted inputs for each neuron yj in the hidden layer are summed and a non-linear activation function is applied to total (equation yj in 4.1). Each hidden neuron's output, or state, is determined by its activation function. The

hidden layer's states are then used as inputs to the subsequent layer, and the process is repeated to determine the state of each output neuron (see 4.1 and 4.2 for the equation for (z k).

$$y_j = \sum_{i=1}^{d} x_i w_{ji} + x_{j0} \qquad \widehat{z_k} = \sum_{i=1}^{d} y_j w_{kj} + y_{k0} \tag{4.1}$$

$$\widehat{z_k} = \sum_{j=1}^{n_h} w_{kj}\left(\sum_{i=1}^{d} w_{ji} x_i + x_{j0}\right) + w_{k0} \tag{4.2}$$

The goal of Backpropagation is to reduce the error J (w) between the output states z and the target values z. The Euclidean distance is a commonly used unit of error, often referred to as a loss function:

$$J(w) = \frac{1}{2}\sum_{k=1}^{c}(z_k - \widehat{z_k})^2 = \frac{1}{2}\|z - \hat{z}\|^2 \tag{4.3}$$

Minimizing error is accomplished through gradient-descent, in which the connections between neurons are modified in accordance with the derivative of the error with respect to

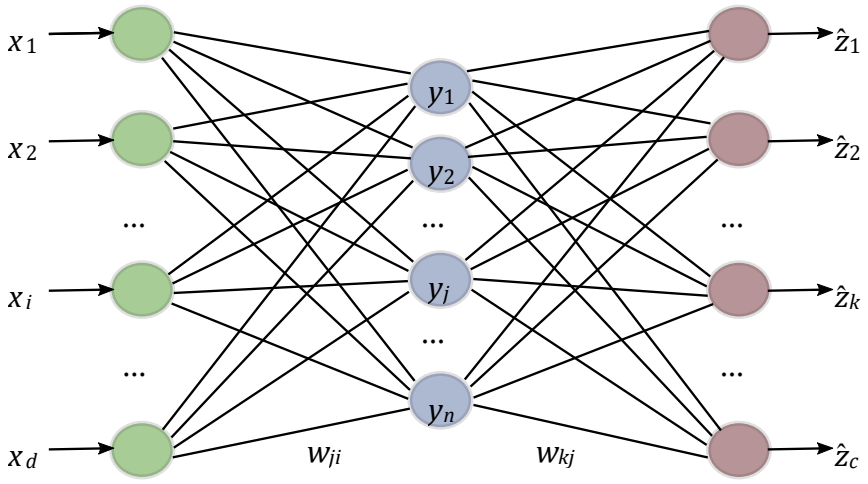


Figure 4.1: Standard Feed-Forward Neural Net.

Gradient-descent error minimization is achieved by modifying the connections between neurons in accordance with the derivative of the error with regard to the weights w=- J/w. The learning rate is a parameter that determines the magnitude of each update. Numerous variables may affect the activation function used, but it is advantageous to have smooth derivatives around zero, for example, with f = tanh(x), since this helps stabilize network behavior when weight values get very small:

$$f(x) = \text{atanh}(\beta.x) = a\frac{1-\exp(-2\beta.x)}{1+\exp(-2\beta.x)} \quad (4.4)$$

$$f'(x) = \frac{\beta}{a}[a + f(x)][a - f(x)] \quad (4.5)$$

The error gradient is determined first with respect to the weights for each neuron in the output layer, and then with respect to the weights for each neuron in the hidden layer (equations 4.6 and 4.7). The direction of steepest descent in the gradient is used to update the weights:

$$\Delta w_{kj} = -\eta(z_k - \widehat{z_k})f'(\widehat{z_k}).y_i \quad (4.6)$$

$$\Delta w_{ji} = -\eta\left[f'(y_j)\sum_{k=1}^{c}(w_{kj})(z_k - \widehat{z_k})f'(\widehat{z_k})\right]x_i \quad (4.7)$$

4.2 PERFORMANCE METRICS

Model performance is quantified using the terms Accuracy, Precision, Recall, and F1-Score. Overall accuracy, class detection rate, and class FP rate are utilized for multiclass classification. Our basic terms are True Positive (TP), which refers to positive instances classified as positive; True Negative (TN), which refers to negative instances classified as negative; False Positive (FP), which refers to negative instances classified as positive; and False Negative (FN), which refers to positive instances classified as negative.

Accuracy: percentage of correctly classified instances and is measured as

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{4.8}$$

Precision: percentage of positive instances that are correctly labeled and is measured as

$$\text{Precision} = \frac{TP}{TP+FP} \tag{4.9}$$

Recall (Detection Rate): percentage of actually positive instances and is measured as

$$\text{Recall} = \frac{TP}{TP+FN} \tag{4.10}$$

F1-Score: weighted average of the precision and recall:

$$F_1 - Score = 2 \times \frac{(precision \times recall)}{(precision+recall)} \tag{4.11}$$

Overall accuracy: exemplars classified correctly from all exemplars.

Class detection rate: ratio of exemplars classified correctly to all exemplars from the given class.

Class FP rate: ratio of exemplars classified incorrectly from given class to all exemplars not from the given class.

Receiver Operating Characteristic curve The receiver operating characteristic curve (ROC curve) used to compare the real positive rate with the false positive rate at different threshold settings.

The Confusion matrix is shown in Figure 4.2.

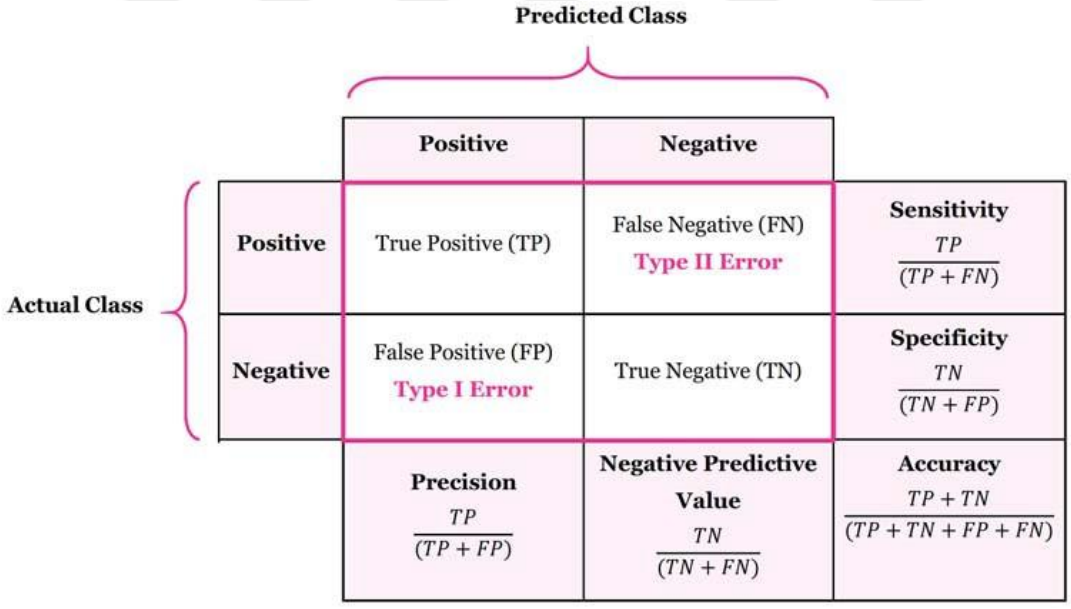


Figure 4.2: Confusion Matrix Representation

4.3 RESULTS

MATLAB 2021a was used to implement the model in this thesis, and the values in Table 4.1 for sensitivity, specificity, and accuracy were computed for training and testing the data.

Table 4.1: Train Data That Used In The Model

0	0	1	1	1	181	5450	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	8	8	0	0	0	0	1	0	0	9	9	1	0	0.11	0	0	0	0	0
0	0	1	1	1	239	486	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	8	8	0	0	0	0	1	0	0	19	19	1	0	0.05	0	0	0	0	0
0	0	1	1	1	235	1337	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	8	8	0	0	0	0	1	0	0	29	29	1	0	0.03	0	0	0	0	0
0	0	1	1	1	219	1337	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	6	6	0	0	0	0	1	0	0	39	39	1	0	0.03	0	0	0	0	0
0	0	1	1	1	217	2032	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	6	6	0	0	0	0	1	0	0	49	49	1	0	0.02	0	0	0	0	0
0	0	1	1	1	217	2032	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	6	6	0	0	0	0	1	0	0	59	59	1	0	0.02	0	0	0	0	0
0	0	1	1	1	212	1940	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	1	0	1	1	69	1	0	1	0.04	0	0	0	0
0	0	1	1	1	159	4087	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	5	5	0	0	0	0	1	0	0	11	79	1	0	0.09	0.04	0	0	0	0
0	0	1	1	1	210	151	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	8	8	0	0	0	0	1	0	0	8	89	1	0	0.12	0.04	0	0	0	0
0	0	1	1	1	212	786	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	8	8	0	0	0	0	1	0	0	8	99	1	0	0.12	0.05	0	0	0	0
0	0	1	1	1	210	624	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	18	18	0	0	0	0	1	0	0	18	109	1	0	0.06	0.05	0	0	0	0
0	0	1	1	1	177	1985	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	28	119	1	0	0.04	0.04	0	0	0	0
0	0	1	1	1	222	773	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	11	11	0	0	0	0	1	0	0	38	129	1	0	0.03	0.04	0	0	0	0
0	0	1	1	1	256	1169	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	4	4	0	0	0	0	1	0	0	4	139	1	0	0.25	0.04	0	0	0	0

0	0	1	1	1	241	259	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	14	149	1	0	0.07	0.04	0	0	0	0
0	0	1	1	1	260	1837	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	11	11	0	0	0	0	1	0	0	24	159	1	0	0.04	0.04	0	0	0	0
0	0	1	1	1	241	261	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	1	0	0	34	169	1	0	0.03	0.04	0	0	0	0
0	0	1	1	1	257	818	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	12	12	0	0	0	0	1	0	0	44	179	1	0	0.02	0.03	0	0	0	0
0	0	1	1	1	233	255	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2	8	0	0	0	0	1	0	0.25	54	189	1	0	0.02	0.03	0	0	0	0
0	0	1	1	1	233	504	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	7	7	0	0	0	0	1	0	0	64	199	1	0	0.02	0.03	0	0	0	0

The three tables below provide a full listing of the characteristics specified for connection records. The contest dataset's data structure is provided in machine-readable format.

Table 4.2 shows the fundamental characteristics of a single TCP connection.

Table 4.2: Basic Features Of Individual Tcp Connections

feature name	***description***	***type***
duration	length (number of seconds) of the connection	continuous
protocol_type	type of the protocol, e.g. tcp, udp, etc.	discrete
service	network service on the destination, e.g., http, telnet, etc.	discrete
src_bytes	number of data bytes from source to destination	continuous
dst_bytes	number of data bytes from destination to source	continuous
flag	normal or error status of the connection	discrete
land	1 if connection is from/to the same host/port; 0 otherwise	discrete
wrong_fragment	number of ``wrong'' fragments	continuous
urgent	number of urgent packets	continuous

Table 4.3 displays the features of a connection as suggested by domain knowledge.

Table 4.3: Content Features Within A Connection Suggested By Domain Knowledge

feature name	*description*	*type*
hot	number of ``hot'' indicators	continuous
num_failed_logins	number of failed login attempts	continuous
logged_in	1 if successfully logged in; 0 otherwise	discrete
num_compromised	number of ``compromised'' conditions	continuous
root_shell	1 if root shell is obtained; 0 otherwise	discrete
su_attempted	1 if ``su root'' command attempted; 0 otherwise	discrete
num_root	number of ``root'' accesses	continuous
num_file_creations	number of file creation operations	continuous
num_shells	number of shell prompts	continuous
num_access_files	number of operations on access control files	continuous
num_outbound_cmds	number of outbound commands in an ftp session	continuous
is_hot_login	1 if the login belongs to the ``hot'' list; 0 otherwise	discrete
is_guest_login	1 if the login is a ``guest''login; 0 otherwise	discrete

Table 4.4 displays the traffic characteristics calculated with a two-second time window.

Table 4.4: Traffic Features Computed Using A Two-Second Time Window

feature name	*description*	*type*
count	number of connections to the same host as the current connection in the past two seconds	continuous
	Note: The following features refer to these same-host connections.	
serror_rate	% of connections that have ``SYN" errors	continuous
rerror_rate	% of connections that have ``REJ" errors	continuous
same_srv_rate	% of connections to the same service	continuous
diff_srv_rate	% of connections to different services	continuous
srv_count	number of connections to the same service as the current connection in the past two seconds	continuous
	Note: The following features refer to these same-service connections.	
srv_serror_rate	% of connections that have ``SYN" errors	continuous
srv_rerror_rate	% of connections that have ``REJ" errors	continuous
srv_diff_host_rate	% of connections to different hosts	continuous

The training steps for the network are shown in figure 4.3.

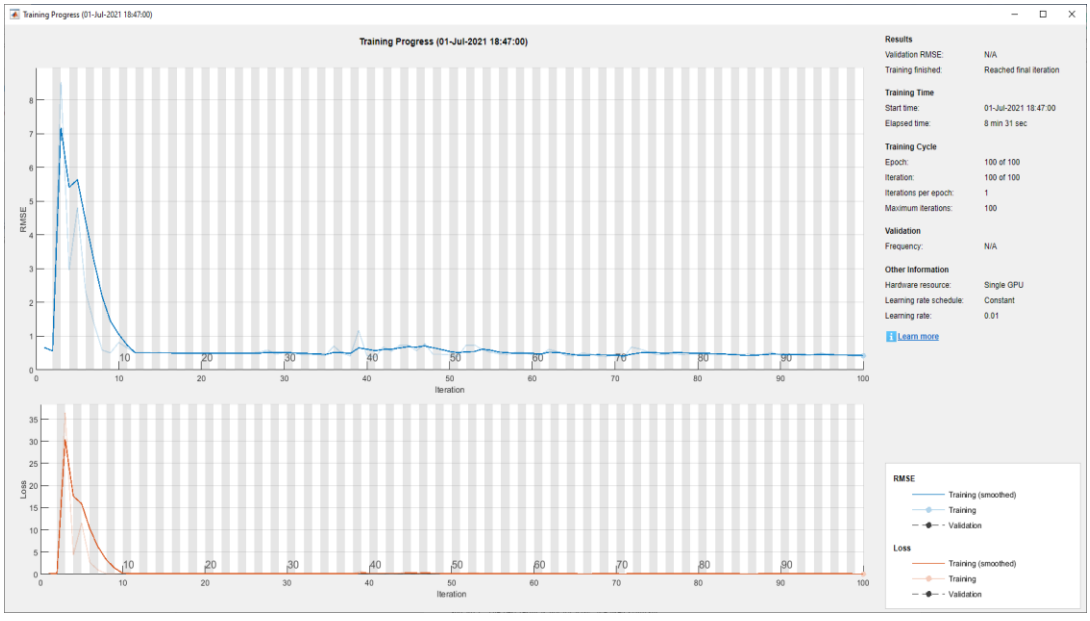


Figure 4.3: Training Step

This figure shows the RMSE and Loss function against of the iteration. the elapsed time is 8 min 31 sec. Also the 100 epochs is selected.

The deep learning network analyzer Figure 4.4 illustrates that.

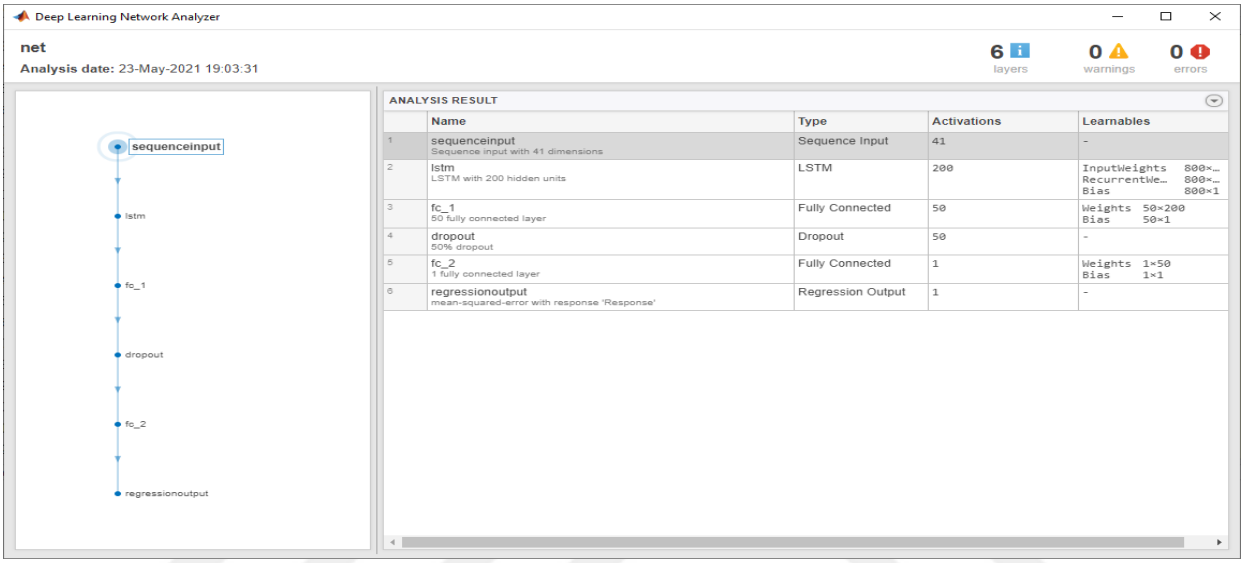


Figure 4.4: Network Analyzer

As seen in this figure 4.5 layers are selected that contain the sequential input, lstm fully connected, dropout, fully connected and regression output.

Information pertaining to all layers is shown in Figure 4.5.

```
6×1 Layer array with layers:

   1   'sequenceinput'      Sequence Input      Sequence input with 6 dimensions
   2   'lstm'               LSTM                LSTM with 10 hidden units
   3   'fc_1'               Fully Connected     50 fully connected layer
   4   'dropout'            Dropout             50% dropout
   5   'fc_2'               Fully Connected     1 fully connected layer
   6   'regressionoutput'   Regression Output   mean-squared-error with response 'Response'
```

Figure 4.5: Layers Information

We evaluated the proposed methods in three scenarios: “Decision tree, Ensemble and CNN”.

4.4 DECISION TREE

A decision tree is a type of technique that helps an institution or organization understand preferences, risks, benefits, and goals. It is also a decision support tool that can be applied in many important investment areas and is used to examine various decision points related to interrelated chance events [53]. It is a way to view an algorithm that contains only conditional control statements.

A decision tree is a method for determining the best approach for achieving a certain objective. It is widely used in the investigation of complex problems, especially in decision analysis. It is also a common tool used in machine learning.

The confusion matrix from decision tree for the IDS system is shown in figure 4.6.

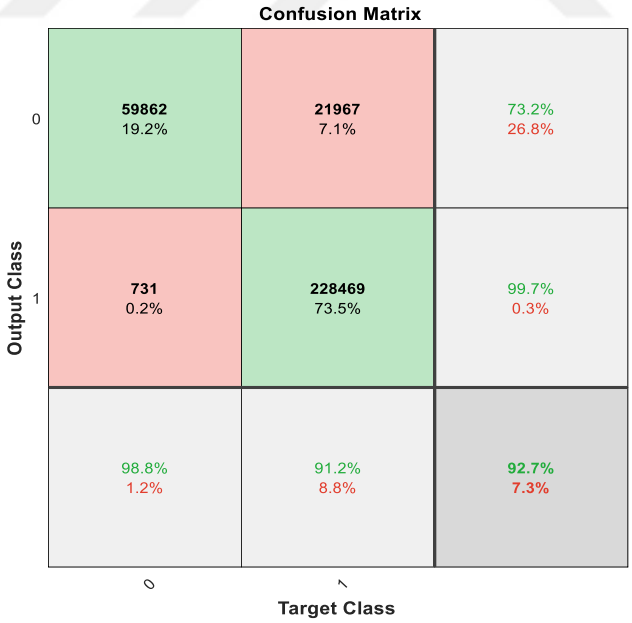


Figure 4.6: Confusion Matrix For Decision Tree

4.5 ENSEMBLE

In statistics and machine learning, ensemble methods combine the results of multiple training algorithms to improve forecasting accuracy over the use of a single training algorithm alone. [54][55][56]. In contrast to the statistical ensemble in statistical mechanics, which is often infinite, the ensemble of techniques in machine learning is a limited collection of different models that typically allows for substantially more flexible architectures.

The Confusion matrix for Ensemble is shown in figure 4.7.

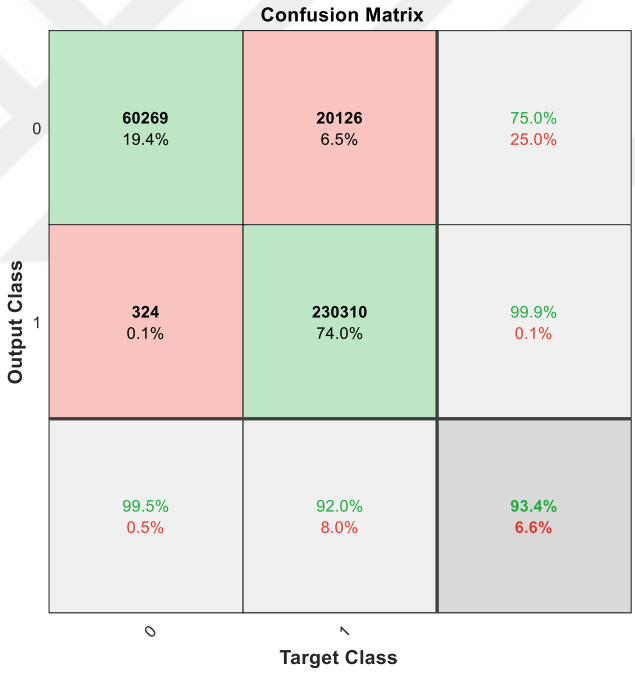


Figure 4.7: Confusion Matrix For Ensemble

4.6 CNN RESULT

Convolutional neural networks are a subfield of deep learning that is used to evaluate visual input. Recognition of images and videos, recommendation systems, and other comparable applications are widespread [57], classification of images, analysis of medical images, and natural language processing [58].

The confusion matrix for CNN result is shown in figure 4.8.

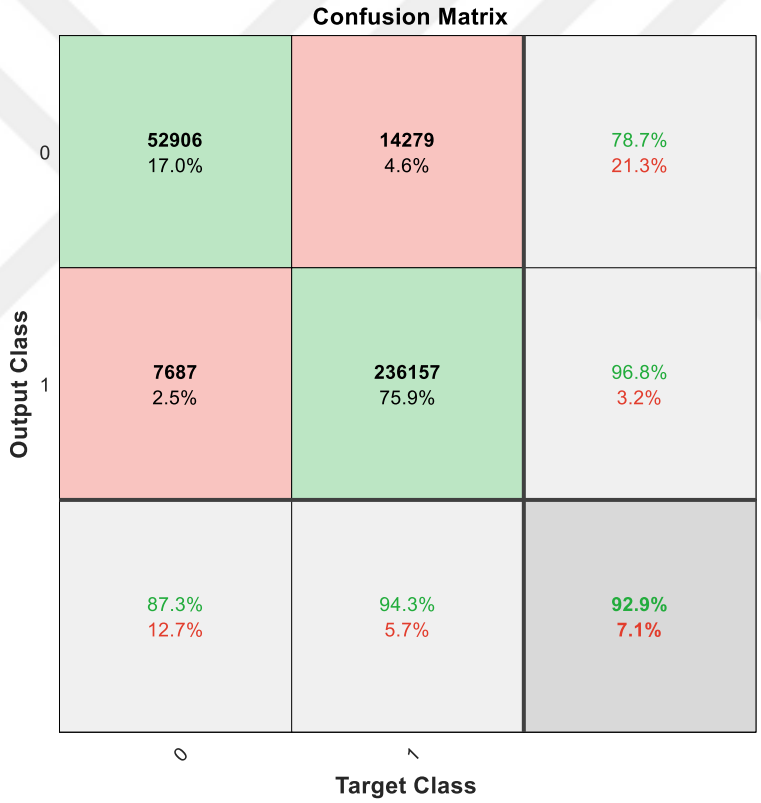
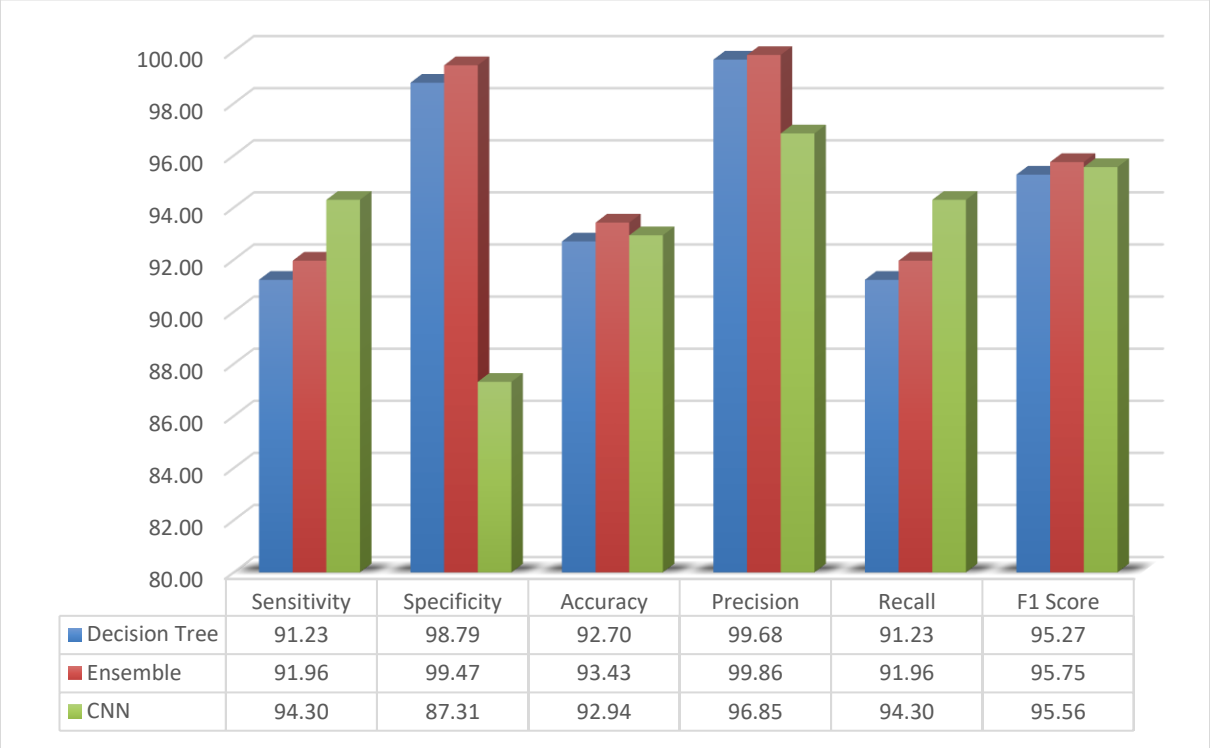


Figure 4.8: CNN Result

4.7 COMPARISON BETWEEN THE RESULTS

The comparison between the results is shown in figure 4.9.



	Sensitivity	Specificity	Accuracy	Precision	Recall	F1 Score
Decision Tree	91.23	98.79	92.70	99.68	91.23	95.27
Ensemble	91.96	99.47	93.43	99.86	91.96	95.75
CNN	94.30	87.31	92.94	96.85	94.30	95.56

Figure 4.9: Comparison Between The Results

For comparison of the results we used the Decision tree and Ensemble method. As seen from the figure the best result for the Sensitivity and Recall obtained from CNN method and it is 94.30% and 94.30% respectively. For Specificity, Accuracy, Precision and F1 Score the Ensemble method has highest percentage and it has 99.47%, 93.43%, 99.86% and 95.75% respectively.

5. CONCLUSION AND FUTURE WORKS

5.1 CONCLUSION

In this study, we have proposed a new IDS detection system using a rare and deep automatic encoder based on the LSTM. There are many automatic detection systems available that automatically detect IDS attacks using deep learning and machine learning techniques. Deep learning method with LSTM method has shown satisfactory results in various areas such as image classification, video recognition, and face recognition. Results compared to other studies. The aim of study applied optimization algorithm LSTM with deep sparse autoencoder to detect IDS in IoTs. The optimization algorithm and deep learning combinations presented remarkable results in previous studies. The deep sparse autoencoder based LSTM algorithm is new and not used in previous studies.

5.2 FUTURE WORK

For future work we can use the meta heuristic methods like, particle swarm optimization, method, or Ant colony optimization method for feature selection and the best feature can be select and implement on the model. With selecting the best features, we can get more accurate results and efficient performance.

REFERENCES

[1] S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan, "Cost-based modeling for fraud and intrusion detection: Results from the JAM project," in *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00*, 2000, vol. 2, pp. 130–144.

[2] M. Alkasassbeh, G. Al-Naymat, A. Hassanat, and M. Almseidin, "Detecting distributed denial of service attacks using data mining techniques," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 1, pp. 436–445, 2016.

[3] A. Abraham, C. Grosan, and C. Martin-Vide, "Evolutionary design of intrusion detection programs.," *Int. J. Netw. Secur.*, vol. 4, no. 3, pp. 328–339, 2007.

[4] W. J. Schull, M. Otake, and J. V Neel, "Genetic effects of the atomic bombs: a reappraisal," *Science (80-.).*, vol. 213, no. 4513, pp. 1220–1227, 1981.

[5] H. Liu, Y. Sun, and M. S. Kim, "Fine-grained DDoS detection scheme based on bidirectional count sketch," in *2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*, 2011, pp. 1–6.

[6] C. Khammassi and S. Krichen, "A GA-LR wrapper approach for feature selection in network intrusion detection," *Comput. Secur.*, vol. 70, pp. 255–277, 2017.

[7] N. Dhieb, H. Ghazzai, H. Besbes, and Y. Massoud, "Scalable and Secure Architecture for Distributed IoT Systems," in *2020 IEEE Technology & Engineering Management Conference (TEMSCON)*, 2020, pp. 1–6.

[8] C. Machine, "The Only Coke Machine on the Internet." The Carnegie Mellon University Computer Science Department, Jan, 2014.

[9] F. Palermo, "Internet of Things done wrong stifles innovation," *Inf. Week*, vol. 7, 2014.

[10] M. Friedemann and C. Floerkemeier, "From the Internet of Computers to the Internet of Things," *Informatik-Spektrum*, vol. 33, no. 2, pp. 107–121, 2010.

[11] I. Aktemur, K. Erensoy, and E. Kocyigit, “Optimization of Waste Collection in Smart Cities with the use of Evolutionary Algorithms,” in *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2020, pp. 1–8.

[12] B. Bhushan and G. Sahoo, “Requirements, protocols, and security challenges in wireless sensor networks: An industrial perspective,” in *Handbook of computer networks and cyber security*, Springer, 2020, pp. 683–713.

[13] G. Pujolle, “An autonomic-oriented architecture for the internet of things,” in *IEEE John Vincent Atanasoff 2006 International Symposium on Modern Computing (JVA’06)*, 2006, pp. 163–168.

[14] J. Cooper and A. James, “Challenges for database management in the internet of things,” *IETE Tech. Rev.*, vol. 26, no. 5, pp. 320–329, 2009.

[15] F. Carrez, “Converged architectural reference model for the IoT.” 2013.

[16] C. Luo, F. Wu, J. Sun, and C. W. Chen, “Compressive data gathering for large-scale wireless sensor networks,” in *Proceedings of the 15th annual international conference on Mobile computing and networking*, 2009, pp. 145–156.

[17] L. Chen, M. Tseng, and X. Lian, “Development of foundation models for Internet of Things,” *Front. Comput. Sci. China*, vol. 4, no. 3, pp. 376–385, 2010.

[18] R. Ramakrishnan, J. Gehrke, and J. Gehrke, *Database management systems*, vol. 3. McGraw-Hill New York, 2003.

[19] R. Cattell, “Scalable SQL and NoSQL data stores,” *Acm Sigmod Rec.*, vol. 39, no. 4, pp. 12–27, 2011.

[20] J. Crichigno, E. Bou-Harb, and N. Ghani, “A comprehensive tutorial on science DMZ,” *IEEE Commun. Surv. Tutorials*, vol. 21, no. 2, pp. 2041–2078, 2018.

[21] A. Sedigh, K. Radhakrishnan, C. E. A. Campbell, and D. Singh, “Trust evaluation of the current security measures against key network attacks,” *MAGNT Res. Rep.*, vol. 2, no. 4, pp. 161–171, 2014.

[22] D. Singh and V. P. Singh, “Comparative study of various distributed intrusion detection systems for WLAN,” *Glob. J. Res. Eng. Electr. Electron. Eng.*, vol. 12, no. 6, pp. 49–56, 2012.

[23] A. S. Eesa, “OPTIMIZATION ALGORITHMS FOR INTRUSION DETECTION SYSTEM: A REVIEW,” *Int. J. Res.*, vol. 8, no. 8, pp. 217–225, 2020.

[24] D. K. Bhattacharyya and J. K. Kalita, *Network anomaly detection: A machine learning perspective*. Chapman and Hall/CRC, 2019.

[25] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, “Intrusion detection system: A comprehensive review,” *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 16–24, 2013.

[26] A. L. Buczak and E. Guven, “A survey of data mining and machine learning methods for cyber security intrusion detection,” *IEEE Commun. Surv. tutorials*, vol. 18, no. 2, pp. 1153–1176, 2015.

[27] K. D. D. Cup, “http://kdd. ics. uci. edu/databases/kddcup99/kddcup99. html,” *UCI KDD Arch.*, 1999.

[28] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” in *2009 IEEE symposium on computational intelligence for security and defense applications*, 2009, pp. 1–6.

[29] N. Moustafa and J. Slay, “UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set),” in *2015 military communications and information systems conference (MilCIS)*, 2015, pp. 1–6.

[30] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization.,” *ICISSp*, vol. 1, pp. 108–116, 2018.

[31] N. Serketzis, V. Katos, C. Ilioudis, D. Baltatzis, and G. Pangalos, “Improving forensic triage efficiency through cyber threat intelligence,” *Futur. Internet*, vol. 11, no. 7, p. 162, 2019.

[32] S. Hajj, R. El Sibai, J. Bou Abdo, J. Demerjian, A. Makhoul, and C. Guyeux, “Anomaly-based intrusion detection systems: The requirements, methods, measurements, and datasets,” *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 4, p. e4240, 2021.

[33] A. L. Samuel, “Some studies in machine learning using the game of checkers,” *IBM J. Res. Dev.*, vol. 3, no. 3, pp. 210–229, 1959.

[34] T. M. Mitchell, “Machine learning,” 1997.

[35] P. Norving and S. Russell, *Artificial intelligence: a modern approach, Global Edition*. Pearson Education Limited, 2021.

[36] J. J. P. Tsai and S. Y. Philip, *Machine learning in cyber trust: security, privacy, and reliability*. Springer Science & Business Media, 2009.

[37] I. H. Witten and E. Frank, “Data mining: practical machine learning tools and techniques with Java implementations,” *Acm Sigmod Rec.*, vol. 31, no. 1, pp. 76–77, 2002.

[38] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, “State-of-the-art in artificial neural network applications: A survey,” *Heliyon*, vol. 4, no. 11, p. e00938, 2018.

[39] I. H. Sarker, “Deep cybersecurity: a comprehensive overview from neural network and deep learning perspective,” *SN Comput. Sci.*, vol. 2, no. 3, pp. 1–16, 2021.

[40] S. Naseer, R. Faizan Ali, P. D. D. Dominic, and Y. Saleem, “Learning representations of network traffic using deep neural networks for network anomaly detection: A perspective towards oil and gas IT infrastructures,” *Symmetry (Basel).*, vol. 12, no. 11, p. 1882, 2020.

[41] R. Chitrakar and H. Chuanhe, “Anomaly detection using support vector machine classification with k-medoids clustering,” in *2012 Third Asian himalayas international conference on internet*, 2012, pp. 1–5.

[42] D. M. Farid, N. Harbi, and M. Z. Rahman, “Combining naive bayes and decision tree for adaptive intrusion detection,” *arXiv Prepr. arXiv1005.4496*, 2010.

[43] S. Peddabachigari, A. Abraham, C. Grosan, and J. Thomas, “Modeling intrusion detection system using hybrid intelligent systems,” *J. Netw. Comput. Appl.*, vol. 30, no. 1, pp. 114–132, 2007.

[44] I. Syarif, A. Prugel-Bennett, and G. Wills, “Unsupervised clustering approach for network anomaly detection,” in *International conference on networked digital technologies*, 2012, pp. 135–145.

[45] J. M. Estevez-Tapiador, P. García-Teodoro, and J. E. Díaz-Verdejo, “Detection of web-based attacks through Markovian protocol parsing,” in *10th IEEE Symposium on Computers and Communications (ISCC’05)*, 2005, pp. 457–462.

[46] K. M. M. Aung and N. N. Oo, “Association rule pattern mining approaches network anomaly detection,” in *Proceedings of 2015 International Conference on Future Computational Technologies (ICFCT’2015) Singapore*, 2015, pp. 164–170.

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

[48] A. Amin, S. Anwar, A. Adnan, M. A. Khan, and Z. Iqbal, “Classification of cyber attacks based on rough set theory,” in *2015 First International Conference on Anti-Cybercrime (ICACC)*, 2015, pp. 1–6.

[49] N. T. Pham, E. Foo, S. Suriadi, H. Jeffrey, and H. F. M. Lahza, “Improving performance of intrusion detection system using ensemble methods and feature selection,” in *Proceedings of the Australasian Computer Science Week Multiconference*, 2018, pp. 1–6.

[50] A. Abraham and J. Thomas, “Distributed intrusion detection systems: a computational intelligence approach,” in *Applications of information systems to homeland security and defense*, IGI Global, 2006, pp. 107–137.

[51] N. Sharma and S. Mukherjee, “A novel multi-classifier layered approach to improve minority attack detection in IDS,” *Procedia Technol.*, vol. 6, pp. 913–921, 2012.

[52] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2012.

[53] B. Charbuty and A. Abdulazeez, “Classification based on decision tree algorithm for machine learning,” *J. Appl. Sci. Technol. Trends*, vol. 2, no. 01, pp. 20–28, 2021.

[54] J. Zhang, K. Xia, Z. He, Z. Yin, and S. Wang, “Semi-Supervised Ensemble Classifier with Improved Sparrow Search Algorithm and Its Application in Pulmonary Nodule Detection,” *Math. Probl. Eng.*, vol. 2021, 2021.

[55] M. M. Jaber, S. K. Abd, P. M. Shakeel, M. A. Burhanuddin, M. A. Mohammed, and S. Yussof, “A telemedicine tool framework for lung sounds classification using ensemble classifier algorithms,” *Measurement*, vol. 162, p. 107883, 2020.

[56] K. Mehmood *et al.*, “Short term power dispatch using neural network based ensemble classifier,” *J. Energy Storage*, vol. 33, p. 102101, 2021.

[57] L. T. A. Sherly and T. Jaya, “An efficient indoor scene character recognition using Bayesian interactive search algorithm-based adaboost-CNN classifier,” *Neural Comput. Appl.*, pp. 1–12, 2021.

[58] B. P. Prathaban and R. Balasubramanian, “Dynamic learning framework for epileptic seizure prediction using sparsity based EEG Reconstruction with Optimized CNN classifier,” *Expert Syst. Appl.*, vol. 170, p. 114533, 2021.