Advancing Anomaly Detection in IoT: A Comparative Study of Machine Learning and Deep Learning Approaches on the IoT-23 Dataset

A MAJOR PROJECT-II REPORT
SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF
MASTER OF TECHNOLOGY
IN
INFORMATION SYSTEMS

Submitted by:
RAJESH KUMAR SAHU
2K22/ISY/13

Under the supervision of

DR. VIRENDER RANGA

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

May, 2024

ACKNOWLEDGEMENT

I would like to express my sincere gratitude towards my supervisor Dr. Virender Ranga for providing her invaluable guidance, comments, and suggestions throughout the course of the project.

The results of this thesis would not have been possible without support from all who directly or indirectly, have lent their hand throughout the course of the project. I would like to thank my parents and faculties of the Department of Information Technology, Delhi Technological University, for their kind cooperation and encouragement which helped me complete this thesis. I hope that this project will serve its purpose to the fullest extent possible.

RAJESH KUMAR SAHU
2K22/ISY/13

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-42

CANDIDATE’S DECLARATION

I RAJESH KUMAR SAHU hereby certify that the work which is being presented in the thesis entitled “Advancing Anomaly Detection in IoT: A comparative Study Of Machine Learning And Deep Learning Approaches On The IOT-23 Dataset” in partial fulfillment of the requirements for the award of the Degree of Master of Technology, submitted in the Department of Information Technology, Delhi Technological University is an authentic record of my own work carried out during the period from January to June under the supervision of Dr. Virender Ranga.

The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other Institute.

Candidate's Signature

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

CERTIFICATE BY THE SUPERVISOR(s)

Certified that Rajesh Kumar Sahu **(**2K22/ISY/13) has carried out their search work presented in this thesis entitled “Advancing Anomaly Detection in IoT: A comparative Study Of Machine Learning And Deep Learning Approaches On The IOT-23 Dataset” for the award of Master of Technology from Department of Information Technology, Delhi Technological University, Delhi, under my supervision. The thesis embodies results of original work, and studies are carried out by the student himself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Dr. Virender Ranga
Associate Professor
Department of Information Technology
DELHI TECHNOLOGICAL UNIVERSITY

Date:

ABSTRACT

With the rise of technology, the IoT got easily available to people in various form and various field. Thus, a rapid amount of data is processed in a limited amount of time. So, reliability and scalability become prime issues. But we cannot forget about the security of this data as it can lead to many problems. We have firewall, antivirus etc., for security but to detect it earlier and solve the problem before any problem can occur is the challenging task. Intrusion detection is one of the solutions for it. Anomaly based IDS is better than signature-based IDS as it finds any irregularity then it detects and take proper action. Here we reviewed about the IoT, its architecture, its applications and its challenges. We find the motivation when we find the importance of security challenges of IoT. It can be solved using Machine Learning and Deep Learning as finding the anomaly in crucial to solve this problem. We tried to implement some machine learning and Deep Learning algorithm on a most recent dataset called IoT-23 which was developed by created by Avast AIC laboratory and Stratosphere IPS. We implemented random forest, decision tree, convolution neural network CNN, stacked long short-term memory gated recurrent unit and extreme gradient boosting. In which we find that decision tree and random forest are most suitable for anomaly detection in IoT-23 dataset.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

CHAPTER 3 METHODOLOGY

CHAPTER 4 RESULTS

LIST OF TABLES

CHAPTER 4 RESULTS

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
CNN	Convolution Neural Network
CSD	Clustering-based Semi-supervised Defence
DRL	Deep Reinforcement Learning
DDoS	Distributed Denial Of Service
FL	Federated Learning
GRU	Gated Recurrent Unit
IDS	Intrusion Detection System
IOT	Internet Of Things
KNN	K Nearest Neighbor
LFA	Label Flipping Attack
LSD	Label-based Semi-supervised Defence
LSTM	Long Short Term Memory
PCA	Principal Component Analysis
ROC	Receiver Operating Characteristic
RL	Reinforcement Learning
SEE	unsupervised feature engineering
SMOTE	Synthetic Minority Oversampling Technique
SLSTM	Stacked Long Short Term Memory
SVM	Support Vector Machine
XGB	Xtreme Gradient Boosting

CHAPTER 1

INTRODUCTION

1.1 Internet of Things

IoT is a complex mix of networked things that have sensors, software, and control systems incorporated in them. This allows data to be collected and transferred without the need for human interaction. Using unique identities (UIDs), it links people, machines, devices, and other items so they may exchange data and interact. IoT has a broad range of applications, from cars that provide the best routes to home equipment like refrigerators and TVs that can be controlled by smartphones and smartwatches. Through embedded sensors, devices collect and exchange operational data, creating a network that improves automation and efficiency in routine processes.[16]

A three-, four-, or five-layer structure of sensors, actuators, protocols, cloud services, and communication layers are just a few of the components that make up an Internet of Things architecture. The perceptual, network, and application layers make up the three-layer model. Sensors and actuators are used by the perception layer to gather environmental data. The network layer, which manages data transmission to other hardware and services, prioritizes energy economy, security, and dependability. Based on the information gathered, the application layer offers services tailored to the user. A data processing layer is added to the four-layer architecture, protecting and processing information from the layer of perception before sending for analytics to the layer of application. A business layer is an additional component of the five-layer concept that oversees user privacy, business models, and the IoT system as a whole. In this approach, the processing layer (middleware) manages data storage and analysis, while the transport layer enables data movement between the perception and processing layers. In order to link devices and guarantee smooth data transfer, IoT networks can make use of cellular, mesh, LAN/PAN, and LPWAN networks. [30]

The six primary components of security services in distributed systems are accountability, non-repudiation, authentication, authorization, secrecy, and data integrity. Sensitive information is kept hidden from unauthorized parties thanks to confidentiality. Data integrity guards against unwanted modifications by preserving the accuracy and consistency of data. Through techniques like digital signatures, authentication verifies identities to build confidence between parties. Only genuine person can access the resources. Non-repudiation ensures responsibility by preventing the validity of utterances from being disputed. Accountability facilitates traceability by identifying the entities accountable for actions. Attacks that aim to disrupt service availability and cause disruptions include DDoS attacks, attempts to stop users from getting any work; Man-in-the-Middle attacks, which intercept and potentially modify communications; Phishing attacks, which steal personal information; and DoS attacks, which try to stop users from getting any work. These attacks have drawn media attention and frequently target well-known websites. [2]

IoT applications are proliferating in a number of industries.

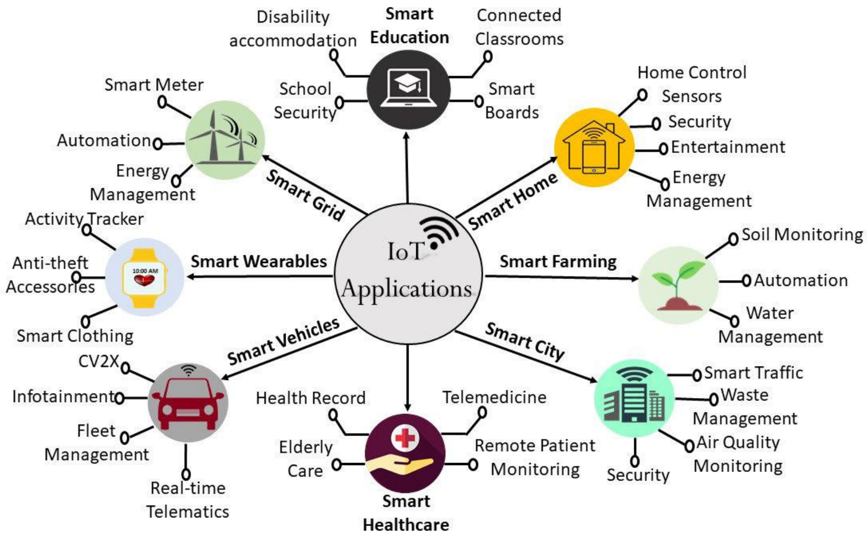


Fig 1.1.1 IoT applications (taken from Mishra et al. [12])

IoT in education makes it possible for classrooms to be connected and helps students with disabilities by providing them with devices like tablets and connected gloves. IoT is used in smart cities and households for entertainment, waste management, air quality, and security. Wearable technology, telemedicine, and remote patient monitoring are beneficial to the healthcare industry. IoT enhances conventional farming in agriculture by facilitating improved soil monitoring and water management. IoT also improves energy management in electric grids and transforms mobility with connected cars. [12]

There are various research problems arising from the swift expansion of IoT devices. Because of the low computing power of IoT devices, standard security methods are unfeasible and security policies are not uniform among vendors.

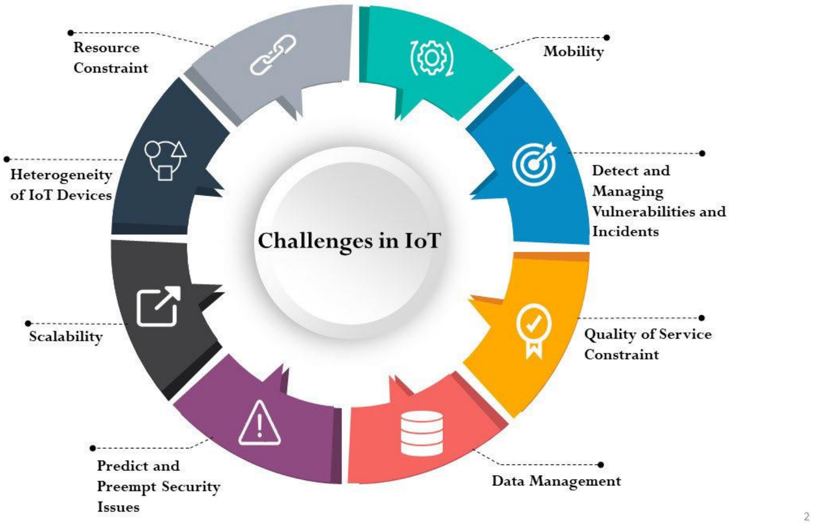

Challenges in IoT
Resource Constraint
Mobility
Detect and Managing Vulnerabilities and Incidents
Quality of Service Constraint
Data Management
Predict and Preempt Security Issues
Scalability
Heterogeneity of IoT Devices
2

Fig 1.1.2 challenges in IoT (taken from Mishra et al. [12])

Device location and power resources significantly complicate security implementation. As mobility is a big factor, trust is more difficult to establish in linked automobiles than it is in stationary applications. Resource limitations for Internet of Things devices include price, power, and size. IoT applications run in distributed environments with a variety of sensors and devices, creating heterogeneity and interoperability challenges that call for shared platforms. Because of its sensitive nature there is difficulty in reliable data transfer. [12]

The sophistication of cyberattacks especially DDoS attacks, has increased with growth of Internet of Things. IoT is vulnerable because of its extensive use low cost and constrained processing power. Common errors such as unprotected channels. Default passwords. And computing constraints are readily exploited.

Perception Layer. This layer is vulnerable to eavesdropping. Malicious code injection and node capturing, since it contains sensors and actuators. They are also susceptible to episodes of sleep deprivation due to power limitations.

Network Layer. This layer is responsible for data transmission. It is vulnerable to DDoS phishing, data transit and routing attacks (such as sinkholes and wormholes). DDoS assaults can use IoT devices as botnets.

Security of databases and cloud storage is crucial. The support layer sitting between network and application layers, manages resource allocation computation and storage. It is susceptible to DDoS. Man-in-the-Middle. And SQL injection attacks.

Application Layer: This layer deals directly with end users and hosts smart applications such as smart homes. Smart cities and smart healthcare systems. It vulnerable to malicious code injection. Sniffing attacks. Intrusions and privacy breaches. Securing this layer is essential to safeguarding user data. Upholding service integrity.[12]

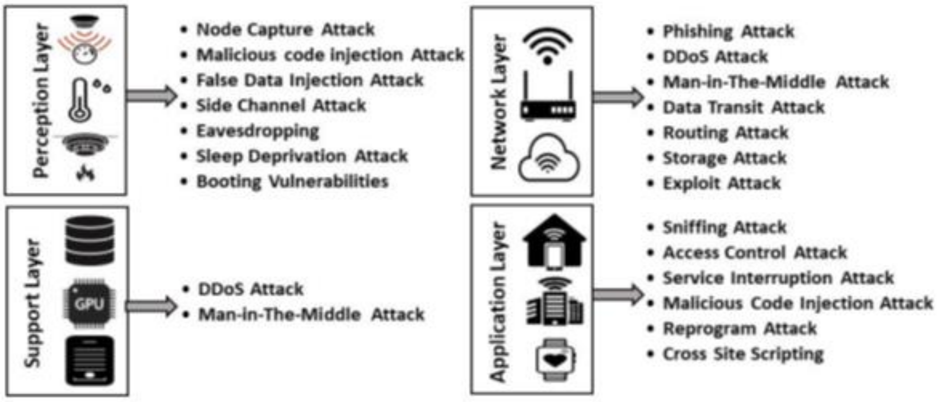


Fig 1.1.3 various attack on IoT layer (taken from Mishra et al. [12])

1.2 Intrusion Detection In IoT

IDS and IPS are two important defenses against DDoS attacks. IDS acts as a preventive measure by sounding an alert during intrusions without taking any further action, hence reducing the possibility of false positives. Punitive measures like IPS act when there is an incursion, but they may inadvertently restrict authorized users. IDS is frequently chosen because of worries about false alarms and the punitive measures of IPS. IDS is essential to security since it stops a wide range of threats and assaults, but it has trouble identifying unknown, unbalanced, and quickly changing attack samples. [4]

Due to the widespread connectivity of IoT devices, real-time remote data collection using a variety of sensors is made possible by extended processing and communication across disparate devices. These systems are safeguarded by intrusion detection systems (IDS), which sound an alert in the case of a security breach and take corrective action. IoT security defensive techniques such as machine learning and deep neural network-driven IDS are quickly becoming indispensable.[8]

IDS products and firewalls work together to create essential security elements that can successfully fend off a variety of security threats. By using machine learning approaches, they can be divided into two categories: schemes for detecting misuse and schemes for detecting anomalies. Attack signatures are the foundation of misuse detection, which provides high accuracy in identifying known malicious activities but is unable to identify new attacks in the absence of signatures. Based on users' typical activity patterns, anomaly detection can identify new assaults but can only classify attacks as binary and may result in false positives, necessitating frequent profile updates. Utilizing machine learning approaches, recent research focuses on both misuse and anomaly detection; however, traditional methods are limited in their ability to be deployed on broad platforms by the absence of labeled training data and their reliance on human-extracted features.[9]

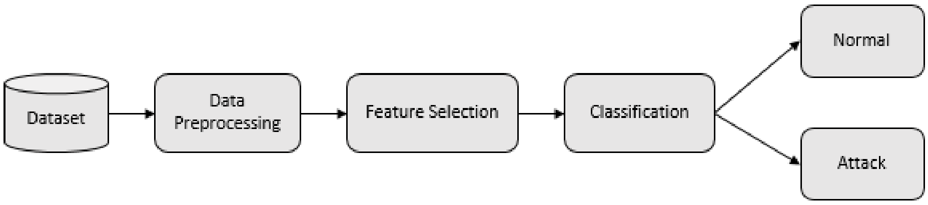


Fig 1.1.4 Block Diagram of Intrusion Detection System

(taken from Mishra et al. [12])

Several IDS kinds consist of:

Network-based intrusion detection systems, or NIDSs, analyze packets as they travel over the network to look for patterns that could indicate an attack.

Host-based intrusion detection systems (HIDS): Track activity on certain hosts or servers and look for indicators of malicious activity that could point to a compromised system.

Malicious activity is detected by anomaly-based intrusion detection systems (IDSs), which learn about and notify users of unusual network or system activity.

Using a database of attack signatures, signature-based intrusion detection systems (IDS) can identify patterns of known malicious activity, but they may not be able to stop zero-day attacks.

Hybrid IDS: Provides thorough security coverage by combining several detection methods, such as anomaly- and signature-based detection. [36][6]

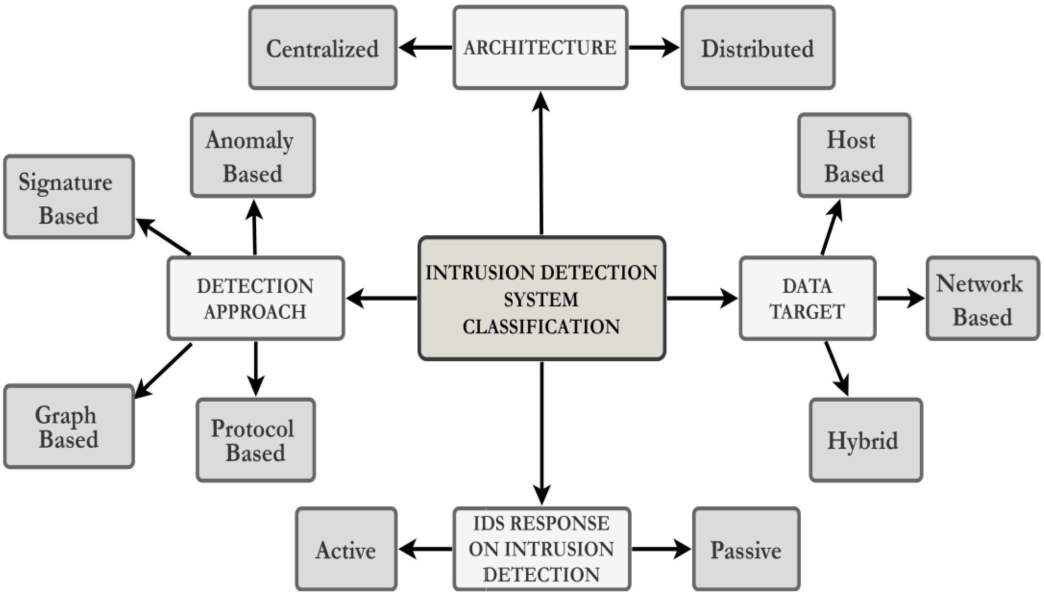


Fig 1.1.5 A graphical representation of classification of various IDS techniques (taken from Mishra et al. [12])

Patterns that depart from predicted behavior are called anomalies, and they fall into three primary categories: Point Anomalies: Individual data points that are deemed unusual in relation to the overall data. Contextual anomalies: Data examples that are abnormal in one context but not in another. Collective Anomalies: Groups of related data points that are out of the ordinary for the whole collection. [15]

Based on training data, anomaly detection may be divided into three primary categories to find patterns in data that differ from expected behavior: Supervised: Instances that are typical and abnormal are tagged for training. For each groups, models are constructed and contrasted. The rarity of anomalous cases and the difficulty of appropriately classifying them present challenges. Semi-supervised: Training only uses labeled examples that are normal. Anything that fits the definition of anomalous is not deemed typical. Since these methods don't need anomalous labeling, they are more widely used. Unsupervised: There's no need for training data. In test data, these strategies presume that typical examples are more common than anomalies. On the other hand, there may be a large false alarm rate if this assumption proves incorrect. [5]

Six categories are used to categorize challenges in anomaly-based intrusion detection systems (IDS): potential attacks, datasets, data processing, security issues, newly discovered malware, and increasing network traffic. [14]

1.3 Motivation

Many industries, including smart cities, healthcare, transportation, agriculture, manufacturing, and education, have profited tremendously from the widespread use of IoT devices. However, because of insufficient resources and the massive amount of data generated, the exponential rise in linked IoT devices has highlighted security and privacy problems, making them appealing targets for hackers. Unsecured IoT devices can potentially be used as attack points, opening up IoT networks to creative zero-day attacks. IoT security requires the development of efficient security solutions to identify intrusions and zero-day threats. The focus of IoT security research has shifted to effective detection methods, with a particular emphasis on adopting Deep Learning (DL) for anomaly-based detection because of its adaptability to the IoT environment, including its capacity to handle massive datasets and low necessity for human interaction. [1]

Cybersecurity Ventures estimates that by 2021, the yearly cost of cybersecurity-related damages will approach $6 trillion, while Gartner projects that worldwide cybersecurity costs will reach $133.7 billion by 2022. IDS is one of the many security techniques that have been created to combat these threats. On the other hand, the drawbacks of current IDS include their high false positive rates, slow detection rates, and dependence on dated datasets such NSL KDD and KDD Cup '99. We used the most recent IOT-23 dataset for more precise and current attack detection in order to overcome these problems.[3]

Research in cybersecurity spans various domains such as healthcare, education, and e-commerce, all requiring protection of sensitive client and patient information from cyber attacks. Intrusion Detection Systems (IDS) play a crucial role in detecting and preventing cyber attacks like malware and DDoS, often using deep learning approaches for accurate classification. The increasing need for high-performance IDS motivates significant research efforts in this area. Projections indicate a significant rise in cybercrime costs, with malware threats potentially costing $23.82 trillion annually by 2027.

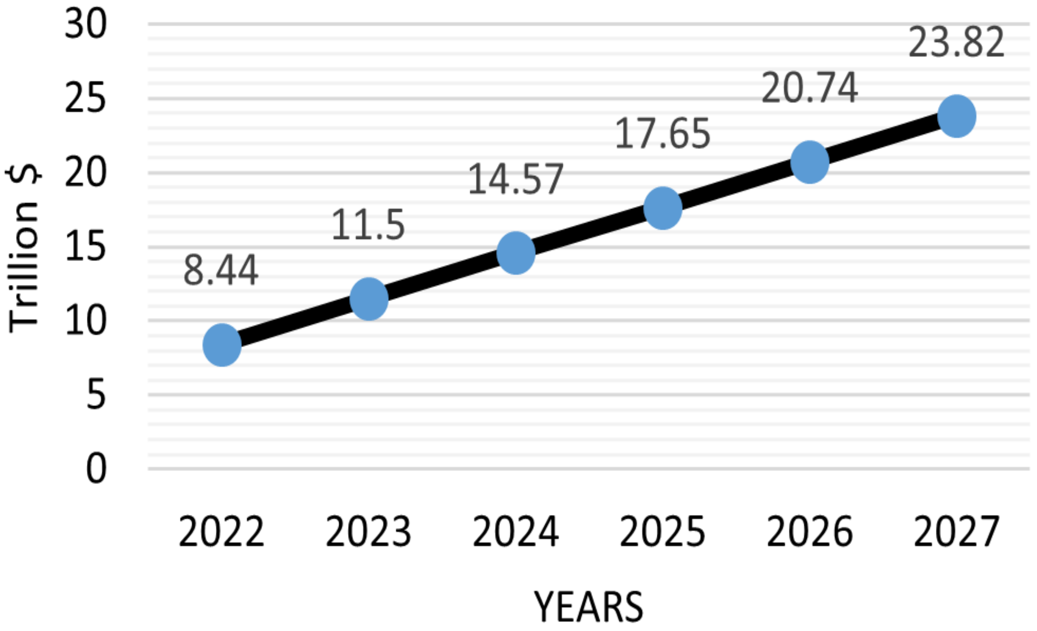


Fig 1.2.1 Statistical Data of Economic Loss Through MalwareAttack from 2022 – 2027(taken from Jahangir et al. [25])

This upward trend raises concerns for businesses heavily reliant on IoT technologies, with economic losses from malware attacks on IoT projected to increase steadily over the coming years.[25]

1.4 Deep Learning & Machine Learning Methods

Machine learning techniques for intrusion detection (IDS) fall into three main categories: reinforcement learning, unsupervised learning, and supervised learning. Tracking learning uses techniques such as decision trees, random forests, and support vector machines to identify malicious or malicious online behavior based on collected data. Unsupervised learning uses methods such as autoencoders, Gaussian mixture models, and K-means clustering to find anomalies or differences in behavior without domain data. Using algorithms such as Q-learning and deep Q-Nets, reinforcement learning uses feedback-based learning to update IDS policies. Convolutional Neural Networks (CNN) are used for image input in deep learning processes; ; Gated Repetitive Units (GRU) are used effectively. Data analysis has great potential for adaptive architectures designed for natural language processing. To achieve deep understanding, combined models of CNN, RNN, and Transformer began to emerge. This technique is flexible and can be used to improve vulnerability detection and network security. [13]

By building sophisticated inference models on huge datasets, machine learning (ML) approaches may detect complicated infiltration patterns with high accuracy without the need for explicit programming. This makes ML techniques highly valued. A branch of machine learning called deep learning (DL) uses multi-layered neural networks to simulate how the human brain learns by extracting high-level features from unprocessed input. DL works well with a variety of input formats, including text, audio, and images. It has a quick learning curve and a high accuracy rate, making it very useful for Big input applications. [10]

IDS are essential for identifying and stopping cyberattacks such as DDoS and malware. Classification techniques based on deep learning increase the accuracy of intrusion detection. Considerable research in intrusion detection systems is driven by the demand for precise and high-performing devices and models. [7]

In order to reduce complexity, feature selection and simplification are made easier with the help of feature importance assessment. Lightweight techniques are preferred for Internet of Things applications, as feature extraction methods improve feature representation. It is preferable to keep precision while increasing speed. [11]

CHAPTER 2

RELATED WORK

Rabbi et al. [17] focused on detecting IoT botnet attacks, specifically the Mirai botnet, using the IoT-23 dataset and various neural network models. They developed a detection technique to distinguish between benign and malicious behavior, achieving an accuracy greater than 90% and an AUC value above 0.93 in all cases. The LSTM algorithm excelled, showing a 99% accuracy in predicting the Hakai strain of the Mirai botnet, demonstrating the effectiveness of deep learning techniques in mitigating cybersecurity threats.

In context of Internet of Things-based intrusion detection systems. Othman et al. [18] investigated impact of an unbalanced dataset on accuracy rates of three machine learning techniques. CNN SVM and ANN. They evaluated accuracy of these algorithms. Overcame dataset imbalances by using Synthetic Minority Oversampling Technique (SMOTE). The results reveal that SMOTE increases accuracy for all three models. KNN performed steadily. ANN displayed a minor accuracy decline. SVM showed sensitivity to imbalances in class ratio. The study highlights how crucial it is to take dataset balance into account. When detecting intrusions.

A machine learning-based method for categorizing data points from IoT-23 dataset is put forth by Sharma et al. [19]. With possibility of being implemented on ESP32 devices. Internet of Things devices can now independently discern between malicious and legitimate network connections. They identify malicious programs. By converting device opcodes into vector space through use of deep learning techniques. The study shows how well their method works. To stop code insertion assaults. It also demonstrates identifying malware. They present a Federated Learning (FL) framework for IoT malware detection that allows development and assessment of models. Without jeopardizing sensitive data. Their solution can be installed on network nodes. That provide Internet of Things (IoT) devices access. Enabling devices to carry out calculations on their own. They determine that best algorithm for their strategy is Decision Tree classifier.

Nanthiya et al. [20] tested effectiveness of machine learning techniques including Support Vector Machine Decision Tree and Random Forest classifying packets subjected to DDoS attacks. They employed Principal Component Analysis (PCA) to lower dimensionality. To enhance algorithm performance. Their study evaluated number of metrics. Including accuracy, precision recall and F1 score. To investigate efficacy of algorithms with and without PCA. The results showed that PCA reduced feature count and algorithm execution time without compromising performance. SVM did not classify DDoS packets. As accurately as Decision Trees and Random Forests did. The research demonstrated utility of PCA for feature selection. And ML algorithm performance augmentation using IoT-23 dataset.

With use of IoT-23 dataset Jeelani et al. [21] developed anomaly detection system for IoT security. This system utilizes machine learning and deep learning techniques. They discovered startling results. In their model, Decision Tree method produced best accuracy. To determine which learning algorithm is best for efficient performance the study examined time costs and performance of several learning algorithms. Out of all ML/DL techniques, results showed Decision Trees had the best accuracy. They also had the least amount of time cost. In contrast Naïve Bayes performed worst.

In order to identify fraudulent data flows and anomalies in IoT networks Ahli et al. [22] suggested using machine learning methods such as Random Forest (RF), Multi-layer Perceptron (MLP) and Gradient Boosting (GB). They developed trained, evaluated models. For binary and multi-label classification using IoT-23 dataset. Random Forest and Gradient Boosting classifiers in particular obtained 98.6% and 97.7% accuracy respectively. This demonstrates great accuracy of their models. Their contribution is creation of machine learning models. These models use supervised learning techniques to classify traffic flows as benign or malicious and to detect anomalies in IoT networks. While Multi-layer Perceptron produced somewhat lower results. Random Forest and Gradient Boosting performed better overall. With good precision recall and F1-score across both classification scenarios.

Using IoT-23 dataset, Fowder et al. [23] examined use of Random Forest, Logistic Regression Naive Bayes and Decision Tree machine learning methods for detection of malicious traffic in IoT networks. The Decision Tree produced best outcomes. Because of imbalance in dataset. Accuracy was not thought to be best metric. For evaluation. Emphasis was placed on F1 Score. Precision and Recall. Best F1 Score was displayed by Decision Tree. And Random Forest (with 50 estimators), demonstrating their efficiency in striking a balance between precision. And recall.

Using IoT-23 dataset Gul et al. [24] applied Random Forest. Also, Naive Bayes and Decision Tree machine learning methods. Their analysis states Random Forest is most effective algorithm. It exhibits best accuracy. Fastest execution times. They further demonstrated feature engineering methods. Preparing datasets. Improving detection and categorization of IoT network attacks.

Avast IoT-23 dataset was used in study by Jahangir et al. [25] to determine optimal algorithm regarding efficiency and performance. Great accuracy and minimal time complexity made Decision Tree (DT) the best option. Their suggested methodology included classifier execution time analysis. Also classification for studying Avast IoT-23 dataset. Purpose was to detect malware. Their findings demonstrated that Decision Trees outperformed other deep learning and machine learning techniques. Decision Tree excelled in both accuracy and computing efficiency.

Using artificial neural networks Ahmed et al. [26] presented effective DDoS attack detection method for smart home networks. High accuracy rates of 99.78% for Multilayered Perceptron (MLP). Also 99.98% for Long-Short-Term Memory (LSTM) models. Their solution attained exceptional results. Their method provides remarkably accurate security for consumers of smart homes. It tackles problem of precisely detecting DDoS attacks in smart home networks.

SEE unsupervised feature engineering method for anticipating DDoS attacks, was first presented by Neira et al. [27]. Tests on three different datasets. (CTU-13 CIC-DDoS2019 and IoT-23). Showed that SEE could 100% accurately anticipate DDoS assaults up to 30 minutes in advance. SEE efficiently detects indications of DDoS attack readiness. Creating new features from network traffic data and using unsupervised machine learning for prediction. With use of cutting-edge data visualization tools, this method improves real-world applicability. Provides quick detection. Helps prevent zero-day attacks.

A novel IDS architecture for IoT devices that makes use of Deep Reinforcement Learning (DRL) was presented by Baby et al. [28] They proposed an AI-based DRL model for IoT attack detection. They analyzed DRL issues. They created intruder attacks using LFA. They also offered two defense techniques. Label-based LSD and CSD were the suggested methods. The NSL-KDD IoT-23 and NBaIoT dataset evaluation results showed. DRL outperforms traditional methods in managing dynamically produced traffic.

Using IoT-23 dataset Bentaleb et al. [29] proposed Convolutional Autoencoder-based model for network intrusion detection in IoT networks. Their method showed promise in detecting different types of attacks. High recall rates and 99.88% accuracy. They successfully and almost completely decreased dimensionality of data by using deep autoencoder neural networks The excellent accuracy on IoT-23 dataset was possible by chosen architecture. Based on convolutional neural layers.

Using methods like RNN LSTM, BiLSTM and GRU Ullah et al. [30] presented deep learning-based anomaly detection model for IoT networks. They presented lightweight model for binary classification. Along hybrid model that included recurrent and convolutional neural networks. NSLKDD, BoT-IoT IoT-NI, IoT-23, MQTT, MQTTset and IoT-DS2 were datasets used to evaluate their models. They achieved good levels of accuracy, precision recall. F1 score when compared to other implementations.

An RNN-based anomaly detection model incorporating kernel bias and activity regularizers was created by Kumari et al. [31] for Internet of Things networks. To improve learning and reduce overfitting, they used activity regularization layers. Layer normalization layers were also applied. They used class weights and borderline SMOTE algorithm to synthesize samples. This method resolves class imbalances Their models performed well in multiclass. Also in binary classification tasks. Tested on a variety of datasets including NSLKDD, BoT-IoT IoT Network Intrusion, IoT-23 MQTT, MQTTset and IoT-DS2.

With emphasis on Internet of Things scenarios Gangone et al. [32] conducted comparative study of machine learning classifiers to identify intrusion in network traffic. Their research aimed to discover which algorithm performed best at identifying different kinds of harmful activities. They assessed performance criteria like recall, accuracy and precision. For IoT Network intrusion dataset. High accuracy rates of 99.11% and 99.99% respectively, were achieved. Using standardized and one-hot encoded features. Their suggested model improved classification performance. They emphasized how well the algorithms for decision trees (DT) and random forests (RF) identify malicious activities in Internet of Things network traffic. They also demonstrated solution for botnet detection that makes use of attributes chosen from IoT-23 dataset. Combines machine learning and deep learning approaches. A deep learning model called GRU fared better than CNN with accuracy of about 99.87% The study underlined how crucial temporal complexity is to IoT device real-time botnet identification.

In order to find anomalies in IoT systems Balega et al. [33] investigated XGBoost's classification capabilities on IoT-23 dataset. They evaluated classification outcomes based on accuracy, precision recall. Other measures were also considered. Contrasting XGBoost with SVM and DCNN, XGBoost demonstrated most efficient execution time. It achieved accuracies up to 99.98%. This outperformed SVM and DCNN. Their research which highlighted XGBoost's better performance in anomaly categorization. Concentrated on supervised machine learning techniques.

Using IoT-23 dataset Teja et al. [34] used Deep Learning. And Machine Learning algorithms to identify anomalies in IoT networks. According to their research, Decision Trees fared better than other models. With 73% accuracy at model time of only 7 seconds. At model prediction time of two minutes Random Forest achieved 73% accuracy. Demonstrating a similar trade-off between execution time and accuracy. CNN offered trade-off that works well for complicated datasets. With an accuracy of 69.4%. And an execution duration of about 4 minutes. While SVM provided competitive accuracy. Its execution time roughly two hour was noticeably longer. With an execution time of only 16 seconds, Naive Bayes achieved lowest accuracy of only 30%

In order to increase number of attack categories Ullah et al. [35] created fresh datasets for their CNN-based anomaly detection model for IoT networks. Their method used BoT-IoT, MQTT-IoT-IDS2020 IoT-23 and IoT-DS-2 datasets. The results outperformed previous techniques in terms of accuracy. Precision, recall and F1 score were also enhanced. Minimum detection rates of 99.74% (CNN1D) 99.42% (CNN2D) and 99.03% (CNN3D) were shown using CNN models. Their results point to model's potential for effective anomaly-based intrusion detection in Internet of Things networks. Demonstrating low false alarm rates and high detection rates.

Using IoT-23 dataset Jyothsna et al. [36] created IDS for IoT. This system can detect. It can stop many types of threats. They used ensemble classifiers. They employed min-max normalization and chimpanzee optimization. Logistic regression served as the meta-classifier. Random Forest, K Nearest Neighbor and XG Boost acted as base classifiers. In order to speed up training. This method decreased number of features while maintaining performance. Their work is vital due to time-sensitive nature of IoT. Also importance of fast attack response.

Using IoT-23 dataset Bains et al. [37] showed how effective machine learning is at identifying fraudulent traffic in IoT networks. Accuracy ranges from 98.9% to 100%. Their research revealed that ML-based IDS/IPS systems can improve IoT network security by efficiently detecting attacks. Plus, reducing false positives their approach shows adaptability to changing traffic in IoT networks. It does not rely on network protocol semantics. IoT network security and resistance to unidentified threats can be increased by integrating ML-based technologies into IDS/IPS systems.

Garcia et al. [38] developed technique that prioritizes feature selection to improve efficiency in identifying DDoS assaults. And other intrusions in Internet of Things networks. They found best feature sets to minimize data dimensionality by using machine learning techniques. With 5-fold cross-validation Random Forest feature significance and sequential forward procedure. This technique resulted in high accuracy rates. Achieving 99.89% for DDoS detection and 98.89% for identifying other attacks. Proved successful in detecting unusual and hostile activities in Internet of Things networks.

CHAPTER 3

METHODOLOGY

3.1 Steps Followed

Some predefined steps are followed to fulfil the aim of detection of malicious or benign in the network. This is a common approach for all model but it may differ in the various steps as different model has different parameter requirement and ways of doing its thing. Finally we study the model based on it performance metrics and find which works best for the dataset.

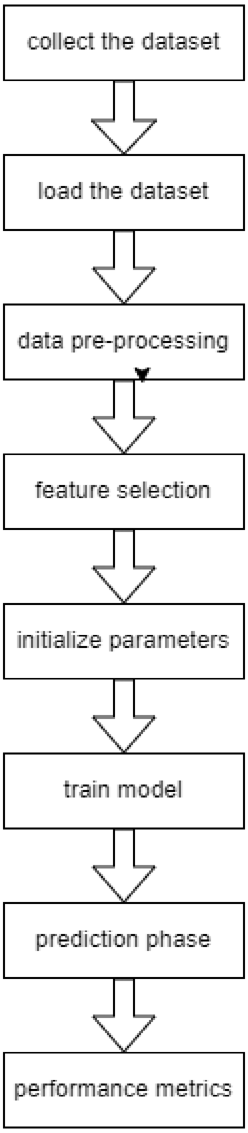


Fig 3.1.1 steps of execution

3.2 Data Preprocessing

An essential part of the machine learning pipeline is data preprocessing, which is the cleaning, transforming, and organizing of raw data to make it fit for analysis. This includes handling missing values, encoding categorical features, normalizing or scaling numerical data, and dividing the dataset into training and testing sets. It can also involve feature extraction or selection to reduce dimensionality and improve model performance. A well-prepared dataset guarantees that machine learning algorithms can learn from it and produce more accurate and dependable predictive models.

1) Clean and encode the data

Find data by removing fields that are similar and therefore considered redundant; Here, the .pcap file is first converted to csv file format. Then remove some unnecessary lines from the configuration file, including ts, uid, id.orig_h, id.org_p, id.resp_h, id.resp_p, service, local_orig, local_resp and origin Km. Add -1 for each empty or missing key. One way to convert categorical data into numerical representation suitable for machine learning algorithms is label coding. In this way, a different number is assigned to each category or category recorded in the data set. After preprocessing, the data is scaled using the normal scaler method.

2) Label binarization

Change the dataset to create a new row with values 1 and 0 representing bad and bad cars, and this row contains targets such as negative text or negative text. instead it is binarized. This is done by using the label encoder to assign a value to each group in the column. After returning the result, call the transform function and pass the parameters, setting the positive value to 1 and all other values to 0.

3.3 Machine Learning Techniques Used

DECISION TREES

Decision trees which have tree-like structure with nodes representing decisions based on feature values, branches signifying outcomes. And leaf nodes indicating final predictions are basic machine learning model used for classification and regression problems. They are made by recursively dividing dataset according to information gain or Gini impurity criteria. The aim is minimizing prediction error or maximizing the separation of various classes. ID3, C4.5 and CART are well-known algorithms. These employ various techniques for managing and dividing data. Decision trees are susceptible to overfitting despite their ease of use interpretability and versatility. However, this can be lessened with use of strategies like pruning. They are a popular choice because of their intuitiveness. Although they have bias towards characteristics and are sensitive to slight changes in data.

RANDOM FOREST

Building many decision trees during training. Generating the mean prediction (regression) or mode of classes (classification) of individual trees (classification). This is how random forests an ensemble learning technique, are formed. Random subset of characteristics is taken into account at each split in tree. Increasing variety and decreasing overfitting. Each tree in forest is trained on random subset of data bootstrap aggregating or bagging. Since of this randomness, random forests are more reliable. They are also more accurate than individual decision trees since it lowers variance of model. High accuracy and resistance to overfitting. They have capacity to manage sizable datasets with increasing dimensionality. Due to huge number of trees they can be more computationally costly. They are also less interpretable than single decision trees. Random forests are a potent model. They are extensively used in many different applications due to their ability to balance robustness and accuracy.

XGBoost (Extreme Gradient Boosting)

It is potent scalable ensemble learning technique with great performance. It is applied to regression and classification problems. Gradient boosting is used to build ensemble of trees sequentially. Each new tree aims to use its predecessors' mistakes as guide. The model optimizes a regularized objective function to improve the trade-offs between bias and variance. This function consists of regularization term to prevent overfitting and a convex loss function to assess prediction error. Handling missing data internally is essential. So is column (feature) subsampling, which adds randomness and decreases overfitting. Shrinkage (learning rate) adjusts the contribution of each tree. XGBoost efficiently handles large-scale datasets. This allows parallel processing. Its benefits include fast speed accuracy, scalability and versatility in handling different kinds of data. However because to its complexity and multiple hyperparameters, careful tuning may be necessary. This is needed to achieve the best results. Because of these qualities XGBoost is top option for numerous challenging machine learning tasks and practical applications.

1.4 Deep Learning Techniques Used

CNN

Among deep learning models Convolutional Neural Networks (CNNs) are especially well-suited to handle grid-like data like photographs. They are made up of several layers. Fully connected, pooling and convolutional layers are some of these layers. They are intended to automatically and adaptively learn spatial hierarchies of features. Pooling layers decrease spatial dimensions. This improves computing efficiency and lowers overfitting. Convolutional layers apply filters (kernels) across input data to recognize features like edges textures and patterns. The learnt features are integrated for final classification or regression tasks by fully linked layers, usually found at conclusion. CNNs are very useful for image identification object detection and other vision tasks. Their key characteristics include translation invariance, weight sharing and local connection.CNNs are preferred because of their superior accuracy. Generalization performance in challenging visual tasks despite the fact that training them requires substantial amount of labeled data and processing resources. Their accomplishments in contests. And real-world uses highlight their significance in deep learning space

SLSTM

By layering numerous LSTM layers on top of one another stacking Long Short-Term Memory (Stacked LSTM) networks create hierarchical structure for processing sequential input. This improves the performance of standard LSTMs. Each layer catches different levels of temporal patterns. This allows for successful description of complicated dependencies over longer sequences. Lower layers learn fundamental features, higher layers collect more sophisticated patterns. LSTMs solve vanishing gradient problem by controlling information flow through gating methods (forget input and output gates). Higher-level temporal feature abstraction is possible in the stacked architecture. The output of one LSTM layer feeds into the subsequent one. Because of their depth, stacked LSTMs perform better in applications like language modeling machine translation and time series forecasting. Nevertheless, they necessitate substantial computer resources. Meticulous hyperparameter tweaking is also required. Because of its hierarchical method Stacked LSTMs are effective in handling challenging sequential data problems since they can capture intricate temporal connections.

GRU

Recurrent neural network (RNN) architecture known as Gated Recurrent Units (GRUs) was created to solve vanishing gradient issue that conventional RNNs have. It efficiently captures relationships in sequential input. GRUs employ gating methods to control information flow just like Long Short-Term Memory (LSTM) networks. They simplify architecture by doing away with output gate. They also combine input and forget gates into a single update gate.This leads to a simpler, more parameter-rich design. While still performing similarly to LSTMs GRUs frequently facilitate quicker training and lower computing requirements. Update gate chooses how much historical data to keep. Reset gate chooses how much historical data to discard. These two primary gates make up GRUs. GRUs may effectively manage long-term dependencies in sequential data. This is due to their gating mechanism. Because of their ease of use, efficiency and potency in simulating sequential dependencies GRUs are extensively employed. Applications such as speech recognition, natural language processing and time series forecasting are common.

1.5 Evaluation Metrices

In classification tasks accuracy is key parameter measuring model's ability to produce accurate predictions for all classes. It can be expressed mathematically as proportion of correctly identified instances to all occurrences in the dataset. Accuracy is useful statistic for overall correctness. But it may not be the most accurate one in imbalanced datasets. This occurs when one class predominates over others. In certain situations, high accuracy score could be deceptive. The algorithm might only forecast the majority class without accurately identifying subtle differences between minority groups.

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

An important measure to evaluate a model's ability to prevent false positives is accuracy. Precision calculates the percentage of events correctly predicted by the model that successfully predicted it correctly. The correct sample is obtained by dividing the number of correct predictions by the total number of positives and false positives. The importance of this becomes apparent in situations such as a fraud investigation. or a negative diagnosis that could lead to serious consequences. That's why truth is especially important. A low negative value means the sample is less negative than positive. High accuracy scores reflect this.

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

Remember to measure the model's ability to capture all relevant events of a group. This is often called precision or accuracy quality. Calculates the percentage of actual events that belong to the correct class. These are really good guesses. The regression model is calculated by dividing the total number of positives and false negatives. This is divided by the number of actual positive predictions. In cases such as diagnostic testing or error analysis, the cost of faulty products can be high. Memory plays an important role. High recall indicates how well the model preserves relevant information. It shows that he rarely misses good events.

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

The F1 score is especially useful for evaluating the performance of the model on non-smooth data. Provides equal parts precision and recall. The F1 score provides an overall assessment of the model's predictive ability. It is calculated as the harmonic mean of precision and return. The F1-score algorithm demonstrates the ability to reduce the negative. It also captures valuable events by combining facts and returns into a single metric. This makes it a strong benchmark for classification problems. Especially when precision and recall must be considered simultaneously. A higher F1 score indicates better model performance. This minimizes false negatives and false positives. It also shows good sensitivity and recovery.

$$\mathrm{F1-SCORE} = 2 * \frac{\mathrm{PRECISION} * \mathrm{RECALL}}{\mathrm{PRECISION} + \mathrm{RECALL}}$$

The balance between true positives (sensitivity) and false positives (specificity) of the binary classification model as a decision variable is presented by a gain operating (ROC) curve. The performance of the model exceeds the proportion range. This is represented by a curve. Each item has a different value; higher values indicate better performance. The area under the ROC curve (AUC-ROC) measures the overall discriminatory ability of the model. Range from 0 to 1.

CHAPTER 4

RESULTS

4.1 IoT-23 dataset description

This document, called IoT-23, covers benign and malicious Internet of Things (IoT) network traffic. First published in January 2020. Twenty pieces of malware built into IoT devices were also caught. IoT network traffic is collected by AIC Group, affiliated with the Stratosphere Laboratory at FEL CTU University in the Czech Republic. The goal is to provide insight into real data and record IoT malware infections and benign IoT traffic. Researchers can benefit from this. Design machine learning algorithms. Materials for their work were supported by Avast Software in Prague. [39]

23 captures various IoT network traffic scenarios and constitutes the IoT-23 dataset. The scenes are separate. Our website captures actual IoT devices with a list of devices capturing traffic. Twenty network capture pcap files. From infected IoT devices. This will in all cases contain the name of the successful malware model. We ran a specific malware sample on the Raspberry Pi. It uses many rules. Many jobs have led to many bad situations. Three different IoT devices were used: Somfy smart door locks, Amazon Echo home smart personal assistant, and Philips HUE smart LED lights. Collect network traffic from the right events.

Label	Summary Description
Attack	Various attack types towards a different host
Benign	The connections do not exhibit abnormals
C&C	The infected devices connect to CC server
DDoS	The comprehend devices launch a DDoS
File Download	The infected device given a downloaded file

HeartBeat	The target host is tracked by the C&C server through the packets via this connection
Mirai	The connections have style of a Mirai botnet
Okiru	The connections have style of a Okiru botnet
Part Of A Horizontal PortScan	The connections perform a port scan horizontally to gather information
Torii	The connections have style of a Torri botnet

Table 4.1.1 types of labels in iot-23 dataset (taken from Nguyen et al. [38])

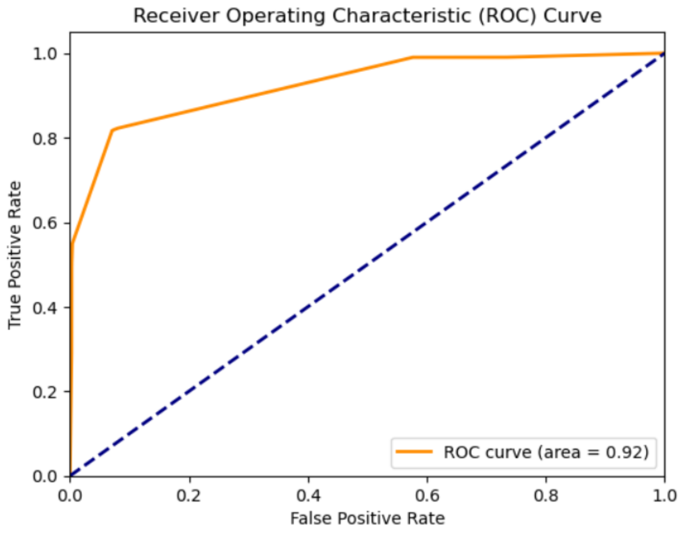
PartOfAHorizontalPortScan	825939	Okiru	362364
Benign	198012	DDoS	138777
C&C	15100	Attack	3915
- benign -	1820	C&C-HeartBeat	471
C&C-FileDownload	43	C&C-Torii	30
FileDownload	13	C&C-HeartBeat-FileDownload	8
C&C-Mirai	1		

Table 4.1.2 no of label types

4.2 Results

4.2.1 Results of Decision Tree

The classification model performed well with 93.78% accuracy, showing that most of the predictions were correct. An accuracy of 93.88% indicates that the model is 93.88% accurate when predicting classes correctly, while a recall of 93.78% indicates that the model can identify 93.78% of all classes well. The F1 score of 93.02% equates to precision and recall and indicates the model's performance in controlling false positives and negatives. The confusion matrix showed that only 1,019 of 268,645 true negatives were misclassified as positives, and 18,213 of 21,422 true positives were misclassified as false negatives; This demonstrates the distinctive power and reliability of the model.



```
ROC AUC Score: 0.9248
```

fig 4.2.1 roc curve of Decision tree

4.2.2 Results of Random Forest

The random forest model was found to perform well with an accuracy rate of 93.85%, indicating that most of the predictions were accurate. A high value of 93.94% indicates that the model is correct 93.94% of the time when it predicts the correct class, while a return value of 93.85% indicates that the model correctly predicts all classes 93.85% of the time. Demonstrates identification ability. An F1 score of 93.11% equates to accuracy and recall and indicates the ability to control the negative and negative. The confusion matrix shows that only 1,051 of 268,613 false positives were incorrectly classified as positive, while 17,969 of 21,666 true positives were incorrectly classified, indicating strong model separation and reliability.

```
ROC AUC: 0.7714
```

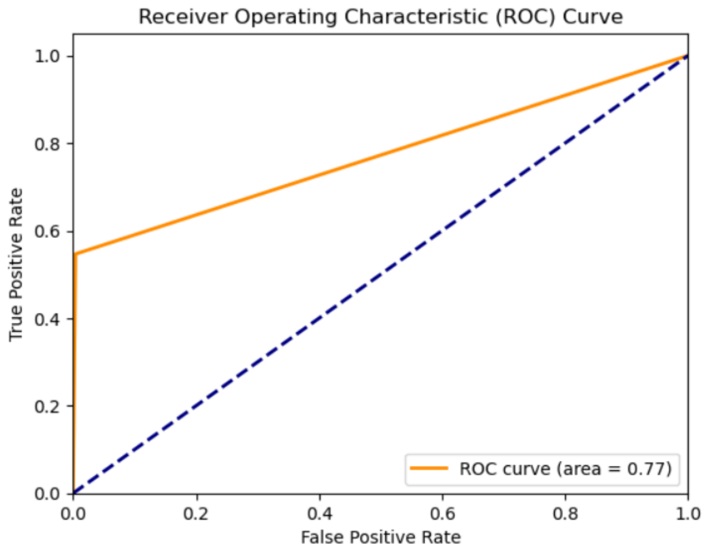


Fig 4.2.2 roc curve of Random forest

4.2.3 Results of XGB

The overall accuracy of the XGBoost model is as high as 93.91%, indicating high prediction accuracy. However, the accuracy of the model is 94.41%; This means that it performs well in reducing negativity at 94.41% when it predicts the correct class. The recovery rate is lower at 55.76%; This shows that the model only identifies 55.76% of all positive cases, indicating that there are some negative cases. The F1 score of 70.11% provides a balance between accuracy and return, but low return affects this balance. The ROC AUC is 0.7764, indicating the strength of discrimination. The confusion matrix showed that 1,308 of 268,356 true negatives were misclassified as positive and 17,534 of 22,101 positives were misclassified as negative. These tests showed that the model had difficulty recovering when accurate, causing many people to disagree.

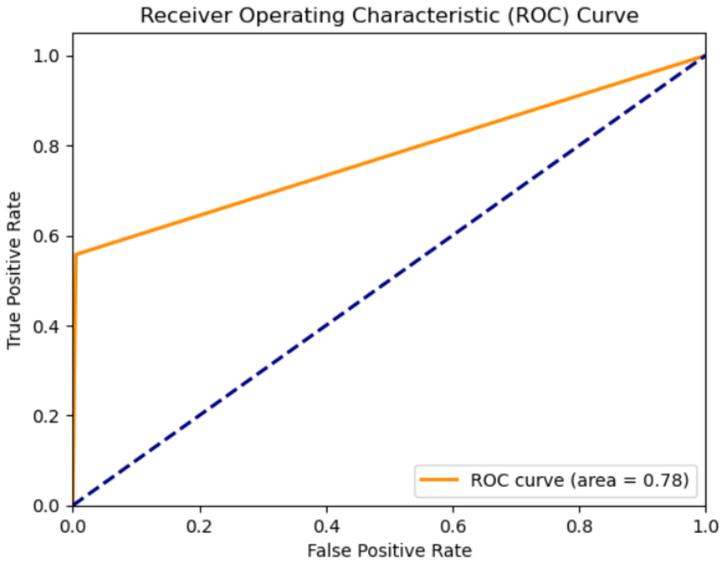


Fig 4.2.3 roc curve of XGB

4.2.4 Results of CNN

The convolutional neural network (CNN) model performed well with an accuracy rate of 89.74%, showing that most of its predictions were correct. The accuracy of 89.46% indicates that 89.46% if the model predicts the class well; This shows that there is more good government than bad. A recovery rate of 89.74% indicates that the model correctly identified 89.74% of true positive samples; hence there is an equal probability of positive examples. The F1 score of 89.59% equates to precision and recall, reflecting the overall performance of the model. The confusion matrix showed that 14,542 of 255,122 true negatives were misclassified as positive and 17,178 of 22,457 true positives were misclassified as positive. This shows that although CNN is good, there is room for improvement to reduce the downside and negativity.

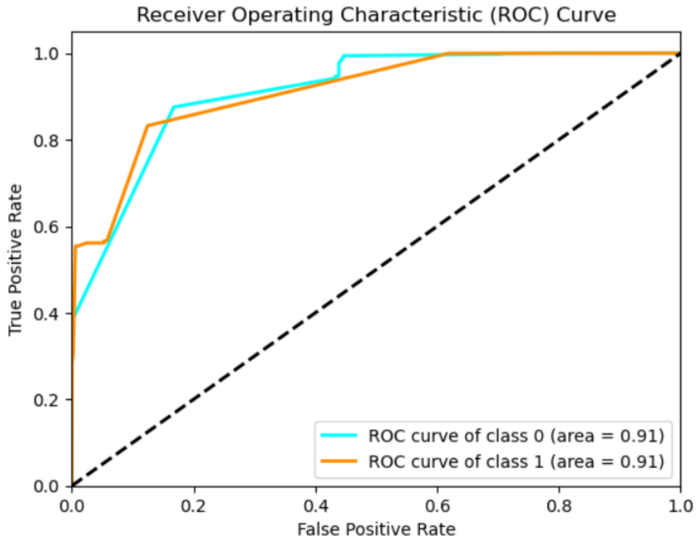


Fig 4.2.4 roc curve of CNN

4.2.5 Results of GRU

The gated recurrent unit (GRU) model showed a robust ROC AUC of 0.8765; this indicates a strong ability to discriminate between positive and negative classes. It achieved an accuracy of 91.23%, which means it was classified in most cases. An accuracy of 88.21% indicates that if the model predicts the class well, it is 88.21% and indicates better management than poor. However, the recovery rate of 69.47% indicates that the model identified 69.47% of the true positives; This indicates a large number of false positives. The F1 score of 77.78% equals precision and recall, indicating the effectiveness of the model. The confusion matrix showed that 9,380 of 260,234 true negatives were misclassified as positive, and 15,321 of 24,364 true positives were misclassified as false. These measurements show that although the GRU model is capable of discrimination, there is still room for improvement in reducing the number of negatives and encouraging further recovery.

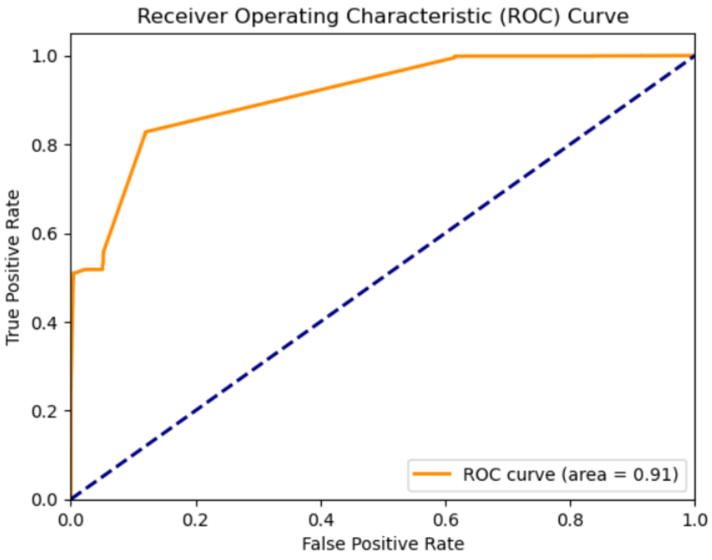


Fig 4.2.5 roc curve of GRU

4.2.6 Results of SLSTM

The testing accuracy of long-term memory model (SLSTM) reached 93.78%, indicating that the overall accuracy of prediction is high. With an accuracy rate of up to 93%, the model is very useful in identifying and predicting good examples, making it possible to reduce negativities. However, the recovery rate is less than 55%, indicating that the model only captures 55% of all positive cases and there are many negative cases. An F1 score of 70% provides a balance between accuracy and recall, but recall rarely affects this balance. The confusion matrix showed that 1,558 of 268,106 true negatives were misclassified as positive, and 17,665 of 21,970 true positives were misclassified as false. These results show that although the SLSTM model performs well in terms of precision and overall accuracy, it performs poorly in terms of recall and improves in capturing all problems well.

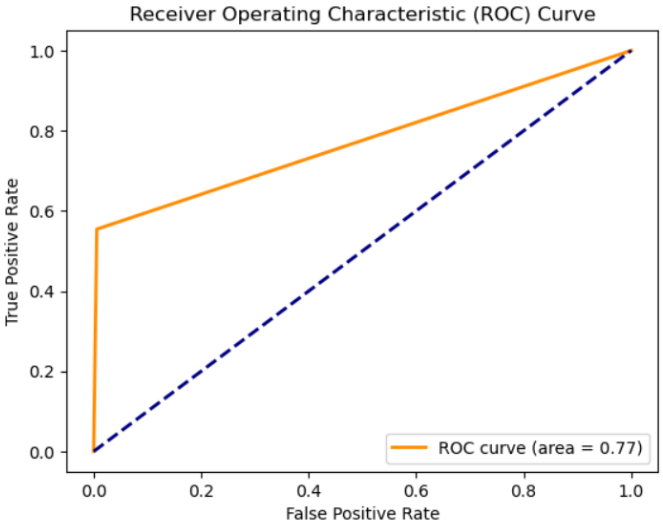


Fig 4.2.6 roc curve of SLSTM

CHAPTER 5

CONCLUSION AND FUTURE WORK

Most models show accuracy ranging from approximately 89% to 94%, indicating their overall performance in identifying most situations. However, there is an important trade-off between the actual model and the returns of individual models. For example, the Random Forest model achieves high performance (93.94%) and recovery (93.85%), while the XGBoost model shows that although it has high accuracy (94.41%), its return is lower (55.76%). False Negative value is too high. Similarly, the SLSTM model showed high accuracy (93%) but low recall (55%). The F1 score, which equates precision and recall, shows the difference, with models like Random Forest having a higher F1 score (93.11%), while models like XGBoost and SLSTM have a lower F1 (70.11% and 70% respectively). In particular, the ROC AUC values for CNN (0.9092) and GRU (0.8765) demonstrate the ability to discriminate, although there is still competition. While most models show complete accuracy and precision, recovery is still a major challenge, especially in models such as XGBoost and SLSTM. The Random Forest model is the most stable and reliable model for applications where errors and false positives must be minimized. Less in quantity.

Future research should focus on developing large datasets to generate relevant up-to-date data that will help deep learning better predict safety. We hope to use machine learning and deep learning to detect more threats in the future. The following issues have been identified as requiring further research to improve the effectiveness of IDS. In future studies, more models should be developed so that they can work well on various data sets. We may consider combining or modifying existing algorithms to perform penetration analysis and produce more accurate results. We will improve feature extraction to make it more accurate.

CHAPTER 6

REFERENCES

1. Ismaeel, Hussain, and Wael Elmedany. "Anomaly-based detection technique using deep learning for Internet of Things: A Survey." In *2022 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, pp. 278-284. IEEE, 2022.
2. Shrivastava, Vineeta, and Anoop Kumar Chaturvedi. "A Survey on Intrusion Detection System Based on Machine Learning and Deep Learning." In *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 1-6. IEEE, 2023.
3. Musa, Usman Shuaibu, Sudeshna Chakraborty, Muhammad M. Abdullahi, and Tarun Maini. "A review on intrusion detection system using machine learning techniques." In *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, pp. 541-549. IEEE, 2021.
4. Bhardwaj, Shweta, Praveen Kumar, and Hima Bindu Maringanti. "Intrusion Detection in Internet of Things using Machine Learning Classifiers." In *2021 International Conference on Technological Advancements and Innovations (ICTAI)*, pp. 571-575. IEEE, 2021.
5. Nassif, Ali Bou, Manar Abu Talib, Qassim Nasir, and Fatima Mohamad Dakalbab. "Machine learning for anomaly detection: A systematic review." *Ieee Access* 9 (2021): 78658-78700.
6. Babaei, Aptin, Parham M. Kebria, Mohsen Moradi Dalvand, and Saeid Nahavandi. "A CNN-Based Deep Learning Approach in Anomaly-Based Intrusion Detection Systems." In 2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 3627-3632. IEEE, 2023.
7. Jangra, Rekha, and Abhishek Kajal. "A Review of Deep Learning based Intrusion Detection Systems." In 2023 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), pp. 1004-1009. IEEE, 2023.
8. Hussain, Ayaz, Hanan Sharif, Faisal Rehman, Hina Kirn, Ashina Sadiq, Muhammad Shahzad Khan, Amjad Riaz, Chaudhry Nouman Ali, and Adil Hussain

Chandio. "A Systematic Review of Intrusion Detection Systems in Internet of Things Using ML and DL." In 2023 4th International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), pp. 1-5. IEEE, 2023.

9. Lansky, Jan, Saqib Ali, Mokhtar Mohammadi, Mohammed Kamal Majeed, Sarkhel H. Taher Karim, Shima Rashidi, Mehdi Hosseinzadeh, and Amir Masoud Rahmani. "Deep learning-based intrusion detection systems: a systematic review." IEEE Access 9 (2021): 101574-101599.

10. Abraham, Jitti Annie, and V. R. Bindu. "Intrusion detection and prevention in networks using machine learning and deep learning approaches: a review." In 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), pp. 1-4. IEEE, 2021.

11. Eriza, Aminanto Achmad, and M. T. Survadi. "Literature review of machine learning models on intrusion detection for internet of things attacks." In 2021 International Conference on Electrical, Computer and Energy Technologies (ICECET), pp. 1-5. IEEE, 2021.

12. Mishra, Nivedita, and Sharnil Pandya. "Internet of things applications, security challenges, attacks, intrusion detection, and future visions: A systematic review." IEEE Access 9 (2021): 59353-59377.

13. Jakotiya, Komal Shyamsundar, Vishal Shirsath, and Raj Gaurav Mishra. "Review on Intrusion Detection System Using Deep Learning and Machine Learning." In 2023 International Conference on Integration of Computational Intelligent System (ICICIS), pp. 1-4. IEEE, 2023.

14. Omarov, Bauyrzhan, Omirlan Auelbekov, Tursynay Koishiyeva, Ruslan Sadybekov, Yerkebulan Uxikbayev, and Aizhan Bazarbayeva. "IoT Network Intrusion Detection Using Machine Learning Techniques." In 2022 International Conference on Smart Information Systems and Technologies (SIST), pp. 1-6. IEEE, 2022.

15. Yang, Xue, Xuejun Qi, and Xiaobo Zhou. "Deep Learning Technologies for Time Series Abnormality Detection in Healthcare: A Review." IEEE Access (2023).

16. Laghari, Asif Ali, Kaishan Wu, Rashid Ali Laghari, Mureed Ali, and Abdullah Ayub Khan. "A review and state of art of Internet of Things (IoT)." Archives of Computational Methods in Engineering (2021): 1-19.

17. Rabhi, Sana, Tarek Abbes, and Faouzi Zarai. "IoT botnet detection using deep learning." In 2023 International Wireless Communications and Mobile Computing (IWCMC), pp. 1107-1111. IEEE, 2023.

18. Othman, Trifa Sherko, and Saman Mirza Abdullah. "Machine Learning Techniques Evaluation with SMOTE on IoT-23 Dataset." In 2023 9th International Engineering Conference on Sustainable Technology and Development (IEC), pp. 7-13. IEEE, 2023.

19. Sharma, Badal, Rajendra Kumar, Anuj Kumar, Megha Chhabra, and Saumya Chaturvedi. "A systematic review of iot malware detection using machine learning." In 2023 10th International Conference on Computing for Sustainable Global Development (INDIACom), pp. 91-96. IEEE, 2023.

20. Nanthiya, D., P. Keerthika, S. B. Gopal, S. B. Kayalvizhi, T. Raja, and R. Snega Priya. "SVM based DDoS attack detection in IoT using Iot-23 botnet dataset." In 2021 Innovations in Power and Advanced Computing Technologies (i-PACT), pp. 1-7. IEEE, 2021.

21. Jeelani, Falaq, Dhajvir Singh Rai, Ankit Maithani, and Shubhi Gupta. "The detection of IoT botnet using machine learning on IoT-23 dataset." In 2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM), vol.

22. Ahli, Alia, Ayesha Raza, Kevser Ovaz Akpinar, and Mustafa Akpinar. "Binary and Multi-Class Classification on the IoT-23 Dataset." In 2023 Advances in Science and Engineering Technology International Conferences (ASET), pp. 1-7. IEEE, 2023.2, pp. 634-639. IEEE, 2022.

23. Fowdur, Harikeish, Sandhya Armoogum, Geerish Suddul, and Vinaye Armoogum. "Detecting Malicious IoT Traffic using Supervised Machine Learning Algorithms." In 2022 IEEE Zooming Innovation in Consumer Technologies Conference (ZINC), pp. 209-213. IEEE, 2022.

24. Gul, Muhammad Jahanzaib, and Muhammad Khaliq-ur-Rahman Raazi Syed. "Network attack detection in IoT using artificial intelligence." In 2023 International Multi-disciplinary Conference in Emerging Research Trends (IMCERT), vol. 1, pp. 1-6. IEEE, 2023.

25. Jahangir, Muhammad Talha, Muhammad Wakeel, Humza Asif, and Ahsan Ateeq. "Systematic Approach to Analyze The Avast IOT-23 Challenge Dataset For Malware Detection Using Machine Learning." In 2023 18th International Conference on Emerging Technologies (ICET), pp. 234-239. IEEE, 2023.

26. Ahamed, Ismeil, F. Ahamad, Vasile Palade, and A. Ahamed. "Neural network-based distributed denial of service (DDoS) attack detection in smart home networks." In 6th Smart Cities Symposium (SCS 2022), vol. 2022, pp. 174-179. IET, 2022.

27. de Neira, Anderson B., Ligia F. Borges, Alex M. Araujo, and Michele Nogueira. "Unsupervised Feature Engineering Approach to Predict DDoS Attacks." In GLOBECOM 2023-2023 IEEE Global Communications Conference, pp. 1644-1649. IEEE, 2023.

28. Baby, Roshan, Zahra Pooranian, Mohammad Shojafar, and Rahim Tafazolli. "A heterogenous IoT attack detection through deep reinforcement learning: a dynamic ML approach." In ICC 2023-IEEE International Conference on Communications, pp. 479-484. IEEE, 2023.

29. Bentaleb, Asmae, Chaimaa Remmach, and Jaafar Abouchabaka. "A New Hybrid Approach using Deep Learning in handling IoT Attack." In 2023 14th International Conference on Intelligent Systems: Theories and Applications (SITA), pp. 1-7. IEEE, 2023.

30. Ullah, Imtiaz, and Qusay H. Mahmoud. "Design and development of RNN anomaly detection model for IoT networks." IEEE Access 10 (2022): 62722-62750.

31. Kumari, Priyanka, Veenu Mangat, and Anshul Singh. "Comparative Analysis of State-of-the-Art Attack Detection Models." In 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1-7. IEEE, 2023.

32. Gangone, Anjali, Bhoomeshwar Bala, Swapna Gangone, and Bharat Kumar GJ. "The Deep Learning and Machine Learning Methods for Botnet Identification in the Internet of Things." In 2023 6th International Conference on Contemporary Computing and Informatics (IC3I), vol. 6, pp. 435-441. IEEE, 2023.

33. Balega, Maria, Waleed Farag, Soundararajan Ezekiel, Xin-Wen Wu, Alicia Deak, and Zaryn Good. "IoT Anomaly Detection Using a Multitude of Machine Learning Algorithms." In 2022 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), pp. 1-7. IEEE, 2022.

34. Teja, S. Ravi, and D. R. Janardhana. "Enhancing Cybersecurity Through Machine Learning-Based Classification of IoT Network Traffic." In 2023 International Conference on Integrated Intelligence and Communication Systems (ICIICS), pp. 1-7. IEEE, 2023.

35. Ullah, Imtiaz, and Qusay H. Mahmoud. "Design and development of a deep learning-based model for anomaly detection in IoT networks." IEEE Access 9 (2021): 103906-103926.

36. Jyothsna, V., E. Sandhya, R. Roopa, B. Deena Divya Nayomi, D. K. Shareef, and P. Bhasha. "Intrusion Detection System for IoT Networks." In 2023 1st International Conference on Optimization Techniques for Learning (ICOTL), pp. 1-6. IEEE, 2023.

37. Bains, Jayant Singh, Hemanth Varma Kopanati, Rahul Goyal, Bhargav Krishna Savaram, and Sergey Butakov. "Using Machine Learning for malware traffic prediction in IoT networks." In 2021 Second International Conference on Intelligent Data Science Technologies and Applications (IDSTA), pp. 146-149. IEEE, 2021.

38. Nguyen, Hoang Long, Hai-Chau Le, and Minh Tuan Nguyen. "Machine Learning-based Intrusion Detection System for DDoS Attack in the Internet of Things." In 2023 International Conference on System Science and Engineering (ICSSE), pp. 375-380. IEEE, 2023.

39. Sebastian Garcia, Agustin Parmisano, & Maria Jose Erquiaga. (2020). IoT-23: A labeled dataset with malicious and benign IoT network traffic (Version 1.0.0) [Data set]. Zenodo. http://doi.org/10.5281/zenodo.4743746

ANNEXURE-IV

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-42

PLAGIARISM VERIFICATION

Title of the Thesis Advancing Anomaly Detection in IoT: A comparative Study of Machine Learning And Deep Learning Approaches On The IOT-23 Dataset” Total Pages 38 Name of the Scholar Rajesh Kumar Sahu

Supervisor (s)

(1) Dr. Virender Ranga

(2)____

(3)____

Department INFORMATION TECHNOLOGY, DELHI TECHNOLOGICAL UNIVERSITY

This is to report that the above thesis was scanned for similarity detection. Process and outcome is given below:

Software used: Turnitin Similarity Index: 8% , Total Word Count: 8766

Date: 30/05/2024

Candidate's Signature **Signature of Supervisor(s)**

PAPER NAME

project.docx

WORD COUNT

8766 Words

CHARACTER COUNT

51654 Characters

PAGE COUNT

38 Pages

FILE SIZE

2.0MB

SUBMISSION DATE

May 29, 2024 11:38 PM GMT+5:30

REPORT DATE

May 29, 2024 11:38 PM GMT+5:30

8% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 4% Internet database
- Crossref database
- 4% Submitted Works database
- 5% Publications database
- Crossref Posted Content database

Excluded from Similarity Report

- Bibliographic material
- Small Matches (Less then 10 words)