INTRUSION DETECTION SYSTEM FOR INDUSTRIAL IOT USING MACHINE LEARNING

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF DEGREE

OF

MASTER OF TECHNOLOGY

IN

SOFTWARE ENGINEERING

Submitted by:

ADITYA CHAUHAN

2K21/SWE/02

Under the supervision of:

Prof. RUCHIKA MALHOTRA

(Professor and Head of Department)

DEPARTMENT OF SOFTWARE ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi – 110042

MAY 2023

MTech(Software Engineering)

Aditya Chauhan

2023

DEPARTMENT OF SOFTWARE ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi – 110042

CANDIDATE'S DECLARATION

I, **Aditya Chauhan** Roll No **2K21/SWE/02** a student of M. TECH (Software Engineering) declare that the project Dissertation titled "**Intrusion detection system for industrial IoT Using Machine Learning**" which is submitted by me to Department of Software Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma, Fellowship or other similar title or recognition.

Aditya Chauhan

Place: DTU, Delhi

ADITYA CHAUHAN

Date: 30/05/2023

2K21/SWE/02

DEPARTMENT OF SOFTWARE ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi – 110042

CERTIFICATE

I, hereby certify that the Project titled "**Intrusion detection system for industrial IoT Using Machine Learning**" which is submitted by Aditya Chauhan, Roll No: 2K21/SWE/02, Department of Software Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is a record of project work carried out by the student under my supervision. To the best of my knowledge, this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Ruchika Malhotra

Place: DTU, Delhi

Prof. RUCHIKA MALHOTRA

Date: 30/05/2023

Head of Department

Department of Software Engineering, DTU

ACKNOWLEDGEMENT

I am very thankful to Prof. Ruchika Malhotra (Head of Department, Software Eng. Dept.) and all the faculty members of the Software Engineering Department of DTU. They all provided immense support and guidance for the completion of the project undertaken by me. It is with their supervision that this work came into existence.

I would also like to express my gratitude to the university for providing the laboratories, infrastructure, test facilities, and environment which allowed me to work without any obstructions.

I would also like to appreciate the support provided by our lab assistants, seniors, and peer group who aided me with all the knowledge they had regarding various topics.

Aditya Chauhan

ADITYA CHAUHAN

M.TECH (SWE)

2K21/SWE/02

ABSTRACT

The Internet of Things (IoT) is a term that refers to all of the gadgets that may connect to the Internet in order to collect and share data. Cybersecurity has been widely used in a variety of applications like including intelligent manufacturing processes, homes, personal gadgets, and automobiles, and has resulted in new advances that continue to confront hurdles in tackling problems linked to IoT device security approaches. The increasing number of cyber security attacks against IoT devices and intermediary media for communication supports the assertion. Attacks against the IoT, if undetected for a long period of time, inflict serious service interruption and financial loss. It also poses the risk of identity theft.

This project addresses the security difficulties that the Internet of Things IoT and industrial IoT devices confront, and it presents a machine learning technique-based intrusion detection solution for IoT devices. The proposed testbed is divided into seven levels, each of which contains new technologies that meet the critical requirements of basic IoT and industrial IoT applications. The dataset used in this study is Edge-IIoTset, which is a complete, realistic cyber security related dataset of IoT and industrial IoT applications that can be used by machine learning technique driven intrusion detection systems. The dataset comprises sensors for humidity, temperature, water level detection, level of pH, moisture in the soil, pulse rate, and flame detection, among other things. The collection also identifies and assesses fourteen attacks on IoT and industrial IoT communication procedures, which are divided into five categories. The study examines the performance of the different machine learning algorithms after processing and analysing the dataset. Real-time intrusion-based detection on Internet of Things devices is critical for making IoT-enabled services dependable, safe, and profitable.

Keywords: Internet of things, cybersecurity, Machine learning algorithm

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Definition
DNN	DEEP **N**EURAL **N**ETWROK
IOT	INTERNET OF THINGS
IIOT	INDUSTRIAL INTERNET OF THINGS
CNN	CONVOLUTIONAL NEURAL NETWORK
RNN	RECURRENT NEURAL NETWORK
TCP	TRANSMISSION CONTROL PROTOCOL
UDP	USER DATAGRAM PROTCOL
ICMP	INTERNET CONTROL MESSAGE PROTOCOL
SVM	SUPPORT VECTOR MACHINE
RF	RANDOM FOREST
DT	DECISION TREE
KNN	K NEAREST NEIGHBOUR
ARP	ADDRESS RESOLUTION PROTOCOL

CHAPTER 1

INTRODUCTION

The Internet of Things IoT and Industrial IoT[19] is a network of interconnected devices that communicate with one another and deliver data to users over the Internet. IoT has grown rapidly in recent0years, thanks in part to its broad0applicability, scalability, and support for0smart applications. The vast majority of IoT application perform functions automatically, with little or no contact from people. Industrial IoT (IIoT) is a subset of IoT in which IoT related devices are often employed in closed industrial contexts. The Internet of Things has been~~0~~successful in reducing resource use while increasing productivity. The Internet of Things (IoT) is a crucial enabler of0industry 4.0, often known as the Fourth Industrial Revolution. There are currently over 8 billion IoT-connected devices,0and the number is still growing.

1.1 IOT and IIOT

IoT refers to a network of physical devices, vehicles, appliances, and other objects that are embedded with sensors, software, and connectivity, enabling them to collect and exchange data over the internet. These connected devices can communicate with each other and with humans, making it possible to gather and analyze large amounts of data for various purposes. Industrial IoT (IIoT), also known as Industry 4.0, focuses specifically on the application of IoT technologies in industrial settings. It involves connecting industrial machinery, equipment, and processes to the internet, allowing for enhanced automation, monitoring, and optimization of industrial operations.

IIoT devices[21] are designed to withstand harsh environments and are used in sectors such as manufacturing, energy, transportation, agriculture, and healthcare. Some examples of industrial IoT devices are sensors, smart meters, assets tracking system, connected machines, robotics, automation machines and industrial gateway devices. There are various types of sensors are used to collect data from the physical environment. Examples include temperature sensors, pressure sensors, vibration sensors, and proximity sensors. These sensors provide real-time information about the condition and performance of machines and equipment. Smart Meters are IoT-enabled meters used in industries to monitor and measure energy consumption, water usage, and other utilities. They provide accurate and timely data for efficient resource management and cost optimization. Asset Tracking Systems are those

IIoT devices equipped with GPS or RFID technology are used to track and monitor the location and condition of valuable assets, such as vehicles, containers, or equipment. This helps in optimizing logistics, improving asset utilization, and preventing theft or loss.

Connected Machines are industrial machinery and equipment can be connected to the internet to enable remote monitoring, predictive maintenance, and performance optimization. Data from these machines can be analysed to detect anomalies, predict failures, and schedule maintenance activities efficiently. Robotics and Automation Systems are IoT devices play a crucial role in industrial robotics and automation. Connected robots and automated systems can communicate with each other and with other devices, enabling seamless integration and coordination in manufacturing and production processes. Industrial gateways and edge devices act as intermediaries between the IIoT devices and the central cloud-based systems. They aggregate data from multiple sensors and devices, perform local data processing, and transmit the relevant information to the cloud for further analysis and decision-making.

Device Exploitation are attacks where attackers can target vulnerabilities in IoT devices to gain unauthorized access or control. This can involve exploiting weak default passwords, software vulnerabilities, or insecure network connections. Once compromised, attackers can take control of the device, manipulate its functionality, or extract sensitive data. Data Breaches on IoT devices often collect and transmit ensitive data. Attackers may attempt to intercept or tamper with this data during transmission, leading to potential data breaches.

Physical Attacks on IoT devices located in public spaces or exposed environments can be physically tampered with or damaged. Attackers may attempt to manipulate or sabotage the devices, leading to service disruptions or unauthorized access. This can result in unauthorized surveillance, disclosure of sensitive information, or compromise of user privacy[18]. Supply Chain Attacks are those attacks where attackers can compromise the supply chain of IoT devices, injecting malicious code or tampering with the device's firmware or hardware. This can lead to pre-installed backdoors, unauthorized access, or compromised device integrity.

Cloud is adapted by many companies in the form of public cloud, private cloud, or hybrid cloud based upon their usage. Cloud provides many types of services like infrastructure as a service (IaaS), Software as a service (SaaS), and platform as a service (PaaS) with this cloud also provides storage as a service for customers based upon rental format [7]. The main advantage of moving towards a cloud environment is cost reduction for maintaining servers and which can be used for further enhancement in security areas.

With a large number of consumers shifting towards cloud-based systems it has many security challenges to be addressed. Due to its security issues, many top companies are not moving toward cloud-based systems due to its sensitive information which cannot be shared on cloud systems [11]. One of the main security challenges between the cloud service provider and the consumer is the trust between them as the consumer thinks that his secure data can be accessed by the cloud service provider and according to the cloud service provider there is a chance that any customer can provide any of attack inside the cloud [8].

Some of the attacks are phishing attacks, DoS attacks, and man-in-the-middle attacks which are possible inside the cloud. There are many types of attacks that are possible inside the cloud that led to the stealing of sensitive data, unavailability of service, and so on [4]. There are many existing studies that show that there many challenges associated with security inside the cloud. It becomes important to identify the security threat and being prevented it. There are many types of attacks in traditional methods that are not sufficient enough to handle those attacks.

Machine learning is an area in which models are trained with data to predict accordingly. Machine learning techniques are mainly classified into three types of learning methods, supervised learning in which labeled data is been provided, unsupervised learning which contains unlabeled data, and semi-supervised learning which contains both labeled as well unlabeled data for training [16].

1.1.1 ATTACKS IN IOT AND IIOT DEVICES

Cyber security is considered a major issue in IoT-based systems. In recent years many researchers are doing studies in this field to identify the major attacks which can be possible inside the IoT devices and how we can prevent them. Many security-based systems are being provided inside the IoT and IIoT devices but mainly all these security systems are based on identifying a specific type of attack or on the most common signature attacks [13]. Many attacks are not based upon signature attacks which lead to the failure of a security system. In this case, it needs the intervention of the cyber security team to test, identify threats and generate any security stem for that attack. It takes hours and days to identify and fix the problem which makes it a disturbance in IoT devices [19]. To deal with this problem machine learning models are being proposed which are well trained to identify any threat.

There are many common types of attacks that are possible inside the IoT and IIoT devices are:

Injection attack: Injection attacks in cyber security refer to a class of attacks where untrusted or malicious data is inserted into a system or application, leading to unintended and potentially harmful consequences. These attacks exploit vulnerabilities in input validation mechanisms and can allow an attacker to execute arbitrary commands, manipulate data, or gain unauthorized access to a system.

SQL injection attacks occur when an attacker inserts malicious SQL statements into an application's input fields that are then executed by the underlying database. The goal is to manipulate the SQL query structure or gain unauthorized access to the database, potentially extracting sensitive information or modifying data. Cross-Site Scripting (XSS) attacks involve injecting malicious scripts into web pages viewed by other users. By exploiting vulnerabilities in input validation, attackers can inject JavaScript or other executable code that is executed by users' browsers, leading to data theft, session hijacking, or defacement of websites. Command injection attacks occur when an attacker injects malicious commands into system commands executed by an application. If input validation is not performed correctly, an attacker can execute arbitrary commands on the underlying operating system, potentially gaining control over the system. By injecting malicious LDAP statements, attackers can manipulate LDAP queries, potentially gaining unauthorized access or extracting sensitive information.

XML injection attacks exploit vulnerabilities in applications that parse XML input without proper validation. Attackers can inject malicious XML content, potentially leading to data exposure, denial of service, or remote code execution. OS command injection attacks occur when untrusted input is passed to the operating system shell as a command. Attackers can execute arbitrary commands on the underlying system, potentially compromising the system's integrity or gaining unauthorized access.

Denial of service (DoS) and Distributed Denial of service (DDoS) attack: DOS (Denial of Service) and DDoS (Distributed Denial of Service) attacks are malicious attempts to disrupt the normal functioning of a network, system, or website by overwhelming it with a flood of illegitimate traffic. In a DOS attack, a single source (usually a single computer or network) attempts to overload a target system or network with a massive number of requests. The goal is to exhaust the target's resources, such as CPU power, memory, or network bandwidth, making it unable to respond to legitimate requests. Common types of DOS attacks include SYN flood, ICMP flood, UDP flood, and HTTP flood.

In a DDoS attack, multiple compromised computers or devices (known as a botnet) are coordinated to simultaneously launch an attack on a target system or network. The combined traffic from the distributed sources overwhelms the target's resources, making it inaccessible to legitimate users. DDoS attacks are more powerful and harder to mitigate compared to DOS attacks. Attackers often use techniques like IP spoofing to hide the source of the attack and make it difficult to trace back to them.

Both DOS and DDoS attacks can cause significant disruption to targeted systems, resulting in financial losses, reputation damage, and loss of user trust. Organizations employ various strategies and technologies, such as traffic filtering, rate limiting, and load balancing, to detect and mitigate these attacks.

Man in the middle attack: Man in middle attack in which an attacker changes the communication between client and server. This type of changes is made without acknowledging them. This type of attack is mainly possible inside the software as a service environment. A Man-in-the-Middle (MitM) attack is a type of cyber-attack where an attacker intercepts and relays communication between two parties without their knowledge. The attacker positions themselves between the communicating parties, allowing them to intercept, alter, or eavesdrop on the communication.

The attacker positions themselves between the legitimate sender and receiver, often by exploiting vulnerabilities in the network infrastructure or by tricking users into connecting to their malicious network. The attacker intercepts the communication flow between the parties, acting as a relay. This can be achieved by hijacking network connections, manipulating routing protocols, or using techniques like ARP spoofing. In some cases, the attacker may modify the communication by injecting malicious content or altering the messages exchanged between the legitimate parties.

ARP Spoofing/Poisoning where the attacker manipulates the address resolution protocol (ARP) to associate their MAC address with the IP address of the target, redirecting traffic intended for the target to the attacker's machine. DNS Spoofing where the attacker intercepts DNS requests and responds with falsified DNS information, redirecting users to malicious websites or services. Session Hijacking where the attacker steals session cookies or session identifiers to impersonate a legitimate user and gain unauthorized access to their account or session. SSL/TLS Stripping where the attacker downgrades secure HTTPS connections to unencrypted HTTP, allowing them to intercept and manipulate the communication without triggering warnings. Wi-Fi Eavesdropping where the attacker sets up a rogue Wi-Fi access point with a similar name to a legitimate network, tricking users into connecting to the malicious network and capturing their traffic.

Information Gathering: Information gathering techniques in cyber security refer to the methods and approaches used to gather intelligence and collect data about a target system, network, or organization. These techniques are often employed by security professionals and ethical hackers to assess the security posture of a target and identify potential vulnerabilities.

One of the common information gathering techniques in cyber security is port scanning. Port scanning is the process of scanning a target system or network to identify open ports and services running on those ports. Port scanning helps an attacker identify which ports on a target system or network are open and listening for connections. Each open port corresponds to a specific service or application that is running and accessible over the network. By identifying open ports, an attacker can gain insights into the potential vulnerabilities or services that can be exploited in subsequent stages of the attack.

OS fingerprinting is an information gathering technique used in cyber security to determine the operating system (OS) running on a target system or device. It involves collecting and analyzing various network characteristics and responses to identify the specific OS being used. OS fingerprinting helps attackers gather information about the target system's underlying operating system. By identifying the OS, attackers can tailor their attacks to exploit known vulnerabilities or weaknesses specific to that OS. OS fingerprinting also provides insights into the target's network architecture and potential security measures in place.

Malware Attacks: Malware attacks in cyber security involve the use of malicious software to compromise the security and integrity of computer systems, networks, and user data. Malware, short for "malicious software," is designed to infiltrate systems, perform unauthorized activities, and often cause harm.

Viruses are self-replicating programs that infect other files or systems by attaching themselves to them. They can cause damage by modifying or corrupting files, degrading system performance, or spreading to other systems. Worms are standalone programs that replicate and spread across networks without the need for a host file. They exploit vulnerabilities in network services or operating systems to propagate and can cause network congestion, data loss, or system crashes. Trojans, or Trojan horses, are disguised as legitimate programs or files but contain malicious code. They trick users into executing them, allowing attackers to gain unauthorized access, steal sensitive information, or control compromised systems. Ransomware is a type of malware that encrypts the victim's files or locks their system, making them inaccessible until a ransom is paid. It can cause significant financial and operational damage to individuals and organizations.

Spyware is designed to gather sensitive information, such as login credentials, browsing habits, or personal data, without the user's consent. It operates in the background, monitoring user activities and transmitting data to the attacker. Adware displays unwanted advertisements, often in the form of pop-ups or banners, to generate revenue for the attacker. It can slow down system performance, track user behaviour, and compromise user privacy. Botnets can be used for various malicious activities, including distributed denial-of-service (DDoS) attacks, spam distribution, or data theft.

TCP is a reliable and connection-oriented protocol used in computer networks. It ensures the reliable delivery of data by establishing a connection between the sender and receiver. TCP provides features like error detection, flow control, and congestion control. It guarantees that data packets arrive in the correct order and handles retransmission of lost packets if necessary.

UDP is a simple and connectionless protocol that operates on top of IP (Internet Protocol). It is known for its low overhead and lack of reliability guarantees compared to TCP. UDP provides a lightweight and fast data transfer mechanism without the need for establishing a connection or maintaining a session.

ICMP is a network layer protocol used for sending error messages and operational information related to IP packet processing. It is mainly used by network devices, such as routers and hosts, to communicate error conditions or provide diagnostic information. ICMP messages include features like ping (echo request/reply), network unreachable notifications, time exceeded notifications, and fragmentation-related messages.

ARP is a protocol used to map an IP address to a physical (MAC) address in a local network. It is used when a device wants to send data to another device on the same network but only knows the IP address of the destination.

DDoS attacks are specifically designed to target web servers, aiming to exhaust their resources and render them inaccessible to legitimate users. These attacks are executed by exploiting numerous compromised devices, which are often assembled into a botnet, in order to generate an overwhelming volume of traffic or requests directed towards the targeted server.

In the case of DDoS attacks on HTTP (Hypertext Transfer Protocol), the attackers typically concentrate their efforts on overwhelming the web server's ability to handle incoming requests. By inundating the server with an immense number of HTTP requests, its capacity can be overwhelmed, rendering it unable to respond to legitimate traffic. As a result, genuine users experience a denial of service, impeding their access to the server.

Participating in or engaging in DDoS attacks is both illegal and unethical, as it inflicts harm and disrupts the availability of online services. It is crucial to uphold the law and utilize the internet responsibly and ethically. Should you encounter any suspicious activity or suspect that you are a victim of a DDoS attack, it is strongly advised to report the incident to the appropriate authorities or seek assistance from a cybersecurity professional.

1.2 MOTIVATION

IoT and IIoT devices are vulnerable to various types of attacks due to their interconnected nature and potential security weaknesses. Here are some common types of attacks that can target IoT devices. Botnets and DDoS Attacks in which IoT devices can be infected with malware and used to form botnets, which are large networks of compromised devices. These botnets can be used to launch Distributed Denial of Service (DDoS) attacks, overwhelming a target system with a flood of traffic and causing service disruptions. One technique that can be used for the detection of various types of attacks is using different machine learning techniques. Machine learning contains different types of algorithms like linear regression, support vector machine, and so on. These machine learning algorithms0are used to predict any security threat inside the cloud. Machine learning algorithms are being trained with

labeled anomalies and which helps us to find any security threat happening inside the IoT and IIoT devices.

1.3 OBJECTIVE

The objective which are being followed for this research study are mainly related to cyber security. It tells about different IoT and IIoT devices related attacks which are possible on it. For classification of different types of attack research study has used different types of machine leaning techniques like SVM, RF, DT and KNN. It has also been performed on deep neural network. All the study has been analyzed on 2 feature classification and 15 feature classification. For the balancing out the dataset sampling techniques has been implemented which modifies the minority classes of attacks. Performance has been measured for edge IoT dataset based before applying sampling techniques and compared with after applying the sampling techniques.

1.4 **THESIS OUTLINE**

The thesis work has total divided into six chapters with each having something related to the techniques.

Chapter 1 has been divided into three sections with first section focuses in the basic of IoT and IIoT related devices . Section second focuses on motivation and third section focuses on the objective part of the thesis.

Chapter 2 gives brief overview related work which has been done earlier about the IoT related datasets in cyber security.

Chapter 3 gives outline about the dataset and techniques which are carried out in this thesis. In this all the ML techniques have been discussed.

Chapter 4 gives overview about software and hardware tools which are used under this research work.

Chapter 5 explains results about the research study. It gives performance of the different ml techniques with or without sampling techniques .

Chapter 6 contains conclusion and future work regarding this research study

CHAPTER 2

RELATED WORK

Machine learning can play a significant role in enhancing cybersecurity by detecting and preventing various types of cyber threats. Anomaly detection can be done using machine learning algorithms can learn the normal behaviour patterns of systems, networks, and users. They can then identify deviations from these patterns, which may indicate potential cyber-attacks or anomalies. By leveraging techniques like unsupervised learning, clustering, or outlier detection, machine learning can detect previously unseen or unknown threats. Intrusion Detection Systems where ml models can be trained on historical data to identify patterns and signatures of known attacks. This enables them to recognize and classify new attacks in real-time. IDS based on machine learning can analyse network traffic, system logs, and other relevant data sources to detect and respond to intrusion attacks. It's worth noting that while machine learning can enhance cybersecurity, it is not a silver bullet and should be used in conjunction with other security measures and practices. It requires careful training, validation, and monitoring to ensure accuracy and avoid false positives or false negatives. Regular updates and retraining are also necessary to keep up with evolving cyber threats.

2.1 LITERATURE VIEW

According to Chkirbene *et al.* [5] proposes a supervised machine learning algorithm that uses a decision tree as a classifier over the UNSW-NB-15 dataset. In this paper, a new methodology has been proposed which makes use of past decision history along with a developed model using a decision tree to identify the malicious user. In this, it has been found that impact of new techniques has increased accuracy from 66% to 90% .

Nandita *et al.* [6] proposes a method based upon a support vector machine to secure the cloud. This paper it has been discussed about intrusion detection systems for the detection of intruders inside cloud servers. It has been found that SVM has poor performance over the larger dataset and by using 34 attributes of the NSL-KDD dataset upon 41 total attributes has an accuracy of 95% .

According to Andrey *et al.* [7] proposes a self-learning-based model for distributed denial of service attack detection where it detects any network changes inside it and minimizes the false detection of a legitimate users as malicious users and vice versa. It has been observed an increase in the accuracy of the model by using this self-learning method.

According to Abdul *et al.* [8] proposes ml algorithms as support vector machine, Random Forest and Naive Bayes for intrusion detection inside the cloud. It mainly focuses on attacks that come under DDoS attacks. It has been found that SVM algorithm has higher performance over other two machine learning algorithms. It can be further used for intrusion detection.

According to Xingxin *et al.* [9] proposes an approach using naive Bayesian classification over encrypted datasets. This paper it has been provided a theoretical concept for security inside the cloud. In this scheme, the provided approach cannot learn anything useful from data over the cloud.

According to Zecheng *et al.* [10] proposes a DDoS intrusion detection method inside the source side of the cloud using nine ml algorithms. It has been found that SVM outperforms better in accuracy than all other machine learning algorithms used here and can detect four kinds of DDoS attacks.

According to Alan *et al*. [11] proposes a method using the ANN algorithm used to detect and alleviate already identified and unidentified DDoS attacks. This paper generally discusses some patterns based upon feature detection to separate the attack traffic from the general traffic. It has been found that it has overall 98% accuracy in compared to other approaches.

According to Munawar *et al.* [12] proposes a method for the classification of data using the KNN data classification algorithm. The main objective of this paper is divided data based upon their working categories: sensitive and non-sensitive data. Data encryption has been applied to sensitive data so that it can secure the data from any type of attack.

According to T. Salman *et al.* [13] proposes a technique for detection of different anomalies of intrusion detection systems and categorization of those anomalies based on different types of attacks on the UNSW dataset. In this, they have used two supervised learning techniques for attack detection in which random forest technique shows 99% accuracy over linear regression technique and categorization has lower accuracy compared to detection system.

According to R. Kumar *et al.* [14] proposes a method to protect the virtual machines inside the cloud against different DoS attacks. In this paper, researcher has classified different DoS attacks using support vector machine algorithms for one class. All the accuracy of different types of DoS attacks has been analysed by calculating the confusion matrix.

According to B. Gulmezoglu *et al.* [15] identifies the problem regarding cross virtual memory attacks on commercial clouds. It mainly attacks the last-level cache inside the virtual machine. In this paper, they have proposed a machine learning technique using a support vector machine which is used for the classification of applications inside the cloud.

According to E. Hesamifard *et al.* [16], all the machine learning algorithms require sensitive data in raw format to train their model in the detection of attacks in the cloud. To overcome this, it proposes a neural network algorithm-based techniques which are applied over encrypted data and return the outcome in encrypted form. Based on the received data we can classify the type of attack related to the cloud.

According to C. Modi *et al.* [17] proposes a network intrusion detection model on NSL-KDD and KDD datasets. This model is used for the detection of DoS attacks inside the cloud by making use of snort and decision tree classifiers. It has been found that by using this model detection of attacks has an accuracy of more than 95%.

According to M. Marwan *et al.* [18], there is a huge security issue over the cloud in terms of healthcare systems for medical image analysis. In this paper various methods have been projected to secure the image inside the cloud using SVM and FCM algorithm. As a result, it has been found it helps in the protection of the cloud without increasing the cost of the encryption process to be embedded inside this model.

According to *et al.* [19], it talks about the security challenges faced by security-defined networks inside the cloud. In some of the techniques data traffic is used for the detection of distributed DoS attacks. In this paper, an XgBoost machine learning-based algorithm has been proposed which has a much higher rate of accuracy and a very low false-positive rate.

According to J. Grover *et.al.* [20], This paper presents a comprehensive review of the existing literature on cloud computing security. The study does not utilize a specific dataset as it focuses on analyzing and summarizing previous research on cloud security issues. The paper contributes to the understanding of security concerns in cloud computing and serves as a valuable resource for researchers and practitioners in the field.

According to M. Mohammadi *et al.* [21], The paper covers a wide range of deep learning algorithms and architectures used in IoT big data and streaming analytics, including deep neural networks (DNNs), convolutional neural networks (CNNs), recurrent neural networks (RNNs), and their variants. The paper also emphasizes the integration of deep learning with edge computing to enable real-time processing and decision-making at the network edge. While the paper does not present specific accuracy values, it serves as a valuable resource for understanding the state-of-the-art in deep learning for IoT big data and streaming analytics. For specific accuracy values associated with deep learning algorithms in IoT data analytics, it is recommended to refer to the original research papers cited in the survey or conduct further research in the specific domain of interest.

According to Y. Meidan *et al.* [22], The authors conducted experiments using real-world IoT botnet attack data collected from a large-scale IoT testbed. They compared the performance of the N-BaIoT system with traditional machine learning algorithms, such as Random Forest, Support Vector Machine (SVM), and k-Nearest Neighbors (k-NN). The results reported in the paper demonstrate the superiority of the N-BaIoT system over traditional machine learning algorithms in detecting IoT botnet attacks. The deep autoencoder-based approach shows promising performance in identifying abnormal network behaviors associated with botnet activities. While the specific accuracy values are not provided in the paper, it highlights the improved detection capabilities of the N-BaIoT system using deep autoencoders. The focus of the paper is on the effectiveness and advantages of the proposed system, rather than reporting precise accuracy metrics.

According to P. Kumar *et al.* [23], The paper highlights the utilization of ensemble learning techniques in the IDS framework, which combines multiple classifiers to enhance the detection accuracy and robustness. It incorporates features such as fog computing to enable distributed processing and decision-making closer to the edge devices in IoT networks. This paper consists of UNSW-NB15 dataset with original IOT based data model of DS2OS. It has been found with ensemble method using XgBoost, KNN and Naïve Bayes algorithm. The proposed model has been found with accuracy with 93.21%.

According to D. Rani *et al.* [24], This paper performs binary classification on TON-IOT dataset for IoT and industrial IoT. Ensemble based technique of machine learning algorithm has been used. In this paper bagging and boosting of classification has been used. It has been analysed that model with normal and anomaly class has been detected and accuracy has been found as 96.2%.

According to M. Zolanvari *et al*. [25], In this paper SCADA based IoT model has been used for intrusion detection. It has been applied over water sensor datasets and mainly classified into two binary classifications as attack or normal. In this pape4r different machine learning algorithms has been used like RF, SVM, DT and KNN. It has been found that RF has highest accuracy of 99.2% for binary features.

Table 2.1 shows the different types of datasets which has been analysed for the detection of attacks. Attacks can vary from single to multiclass with different encoding techniques. Table also shows which type of machine learning techniques has been used or the detection of attacks.

Table 2.1: Comparison of different datasets with Models

Paper Title	**Data set Used**	**Security Threat Addressed**	**Machine Learning Technique Used**
[5]	UNSW-NB-15	Intrusion detection	Decision Tree
[6]	NSL-KDD	Intruder inside cloud.	Support Vector Machine
[7]	RSVNET	Distributed Denial of Service	Relearning Method
[8]	Data of Snort algorithm	Distributed Denial of Service	SVM, Naïve Bayes, RF
[9]	Real time data set	Attacks over encrypted data in cloud	Naïve Bayesians classification
[10]	Data collected from Virtual machines	Different types Denial of Service attack	LR, SVM, Naïve Bayes, RF, DT, k-means
[11]	DoS attacks between 2000-2013	Distributed Denial of Service attack	Artificial Neural Network

[12]	Attack Domain	Data Confidentiality and Sensitivity	K-Nearest Neighbor
[13]	UNSW	Normal, Generic, DoS, Backdoor and Worm attacks	Linear Regression, Random Forest
[14]	Eucalyptus cloud	DoS attacks on Virtual Machine	Support Vector Machine
[15]	Instruction and data cache	Cross VM attacks	Support Vector Machine
[16]	MNSIT	Privacy Preserving	Neural Network
[17]	NSL-KDD and KDD	DoS attacks	Decision Tree
[18]	Health Information Technology	Data Protection and Data Privacy	SVM and Fuzzy C-means Clustering
[19]	KDD 99	DDoS attack	XgBoost Classifier
[20]	NSL KDD	DOS, Malware	LR, RT, RF
[21]	Edge IIoT	DOS, Malware, Injection, Phishing	RT, SVM, KNN, ANN, DT
[22]	N-BaIoT	IoT botnet Attack	SVM, KNN, RF
[23]	UNSW-NB15	DOS, Malware, Injection, Phishing	XgBoost, KNN and Naïve Bayes
[24]	TON-IOT	Normal, Attack	Ensemble learning
[25]	SCADA	Normal, Anamoly	RF, SVM, DT and KNN

CHAPTER 3

DATASET AND TECHNIQUES USED

This section mainly explains about dataset used for intrusion detection in IoT and IIoT devices using different machine learning techniques. In this chapter has been divided into two parts where the first section explain brief about dataset and other section explains about different machine learning techniques used in this project.

3.1 Dataset

This chapter section explains about the dataset used for intrusion detection IoT and IIoT devices. Edge IIoT dataset has been designed based upon seven steps:

In the first configuration of network equipment step where refers to the process of preparing and adjusting network devices to establish a functional and secure network infrastructure. It involves tasks such as physical installation, connection of cables, and configuring various settings to enable communication between devices and ensure proper network operation. Proper setup and configuration of network equipment are essential for establishing a robust and secure network that meets the needs of the users or organization.

In the second step threat and attack modelling in cybersecurity for IoT devices involves the systematic identification, assessment, and analysis of potential threats and attacks that can target IoT devices and their associated networks. It is an essential step in designing effective security measures to protect IoT devices from various malicious activities.

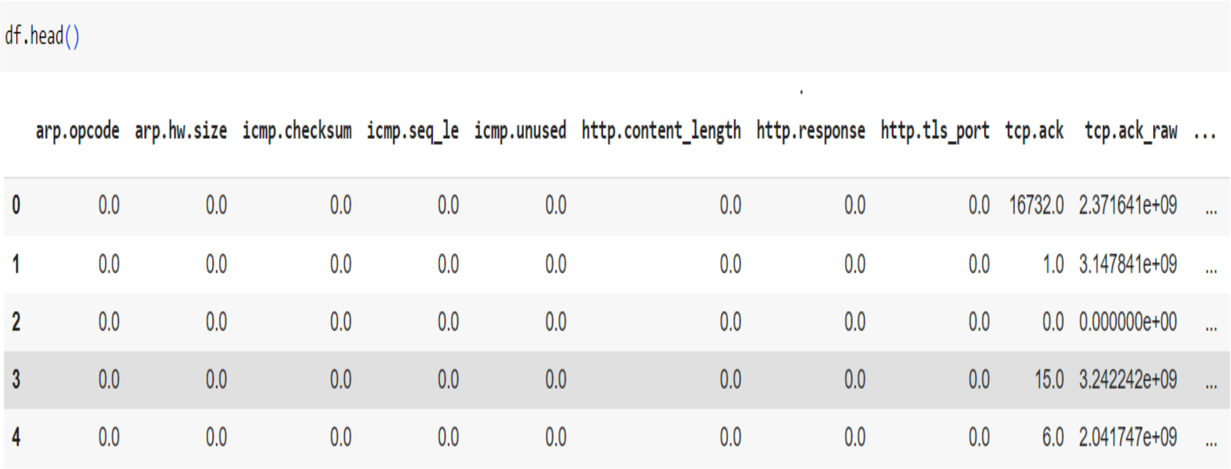
In the third step normal and attack IoT data generation in IIoT devices involves the generation of data that simulates both normal and malicious activities in an industrial IoT environment. This process is important for various purposes, such as evaluating the effectiveness of intrusion detection systems, training machine learning models for anomaly detection, and testing the resilience of IIoT systems against potential cyber-attacks.

In the fourth step normal and attack IoT data collection in cybersecurity involves the process of gathering data from IoT devices to analyze and understand both normal and malicious activities. This data is essential for developing effective intrusion detection systems, training machine learning models, and improving the overall security of IoT environments.

In the fifth step feature extraction in cybersecurity refers to the process of selecting and transforming relevant attributes or characteristics from raw data to represent patterns, behaviours, or indicators of security-related events or anomalies. It plays a crucial role in building effective machine learning models, intrusion detection systems, and security analytics frameworks. Total 61 features have been extracted from 1176 features.

In the sixth step of data processing, we have added attack label where normal is refereed as 0 and all the attack as 1. We have added attack type where it denotes different types of attacks like DDoS, malware, injection attacks and others. After that removal of incorrect and NAN data and the splitting the dataset as train and test.

In the seventh step analyse the performance of the dataset using different machine learning algorithms. Compare the algorithms performance from all and apply curve plotting of it. In the figure 3.1 shows head of the dataset.



```
df.head()
```

	arp.opcode	arp.hw.size	icmp.checksum	icmp.seq_le	icmp.unused	http.content_length	http.response	http.tls_port	tcp.ack	tcp.ack_raw	...
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	16732.0	2.371641e+09	...
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	3.147841e+09	...
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000e+00	...
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	15.0	3.242242e+09	...
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.0	2.041747e+09	...

Figure 3.1: Dataset Explanation

3.1.1 Data Pre-processing

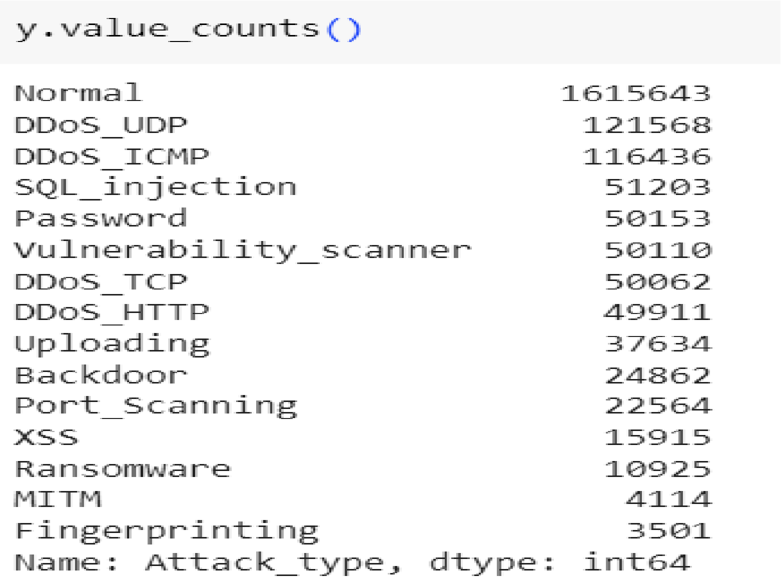
This step is to pre-process the Data Frame by removing unnecessary columns, handling missing values, removing duplicates, shuffling the data, and checking the count of each attack type in the 'Attack type' column. Defines a list called drop columns containing the names of columns to be dropped from the Data Frame. Drops the specified columns from the Data Frame using the drop function with axis=1. Drops any rows that contain missing values (Nan) using the drop function with axis=0. Removes duplicate rows from the Data Frame using the drop duplicate's function. Only the first occurrence of each duplicate row is kept. Shuffles the rows of the Data Frame using the shuffle function from scikit-learn. This step randomizes the order of the rows.

3.1.2 Data Normalization

Data normalization using StandardScaler is a common preprocessing step in machine learning. It is used to transform numerical data into a standardized scale, making it easier for models to interpret and compare features. The StandardScaler class in scikit-learn is a transformer that standardizes features by subtracting the mean and scaling to unit variance. Compute the mean and standard deviation of each feature in the dataset. Subtract the mean from each feature value to center the distribution around zero. Divide each feature value by its standard deviation to scale the values, resulting in a unit variance. The purpose of this normalization is to ensure that all features have similar scales and follow a standard normal distribution (mean=0, variance=1). This can be beneficial for certain machine learning algorithms, such as those based on distance calculations or gradient descent optimization, as it prevents features with large scales from dominating the learning process.

3.1.3 Data Balancing Using SMOTE and Random Oversampling

Random Oversampling and SMOTE (Synthetic Minority Over-sampling Technique) are two commonly used techniques for addressing class imbalance in machine learning. Random Oversampling is a simple technique that involves randomly duplicating samples from the minority class to balance the class distribution. It increases the number of instances in the minority class to match the majority class. This approach can lead to overfitting since it duplicates existing data without introducing any new information. SMOTE is an oversampling technique that generates synthetic samples to balance the class distribution. Instead of simply duplicating existing samples, SMOTE creates new synthetic samples by interpolating between existing minority class samples. It selects a minority class sample and finds its k-nearest neighbours. It then randomly selects one of the neighbours and creates a new sample by combining features from the selected neighbour and the original sample. This process is repeated until the desired level of class balance is achieved.

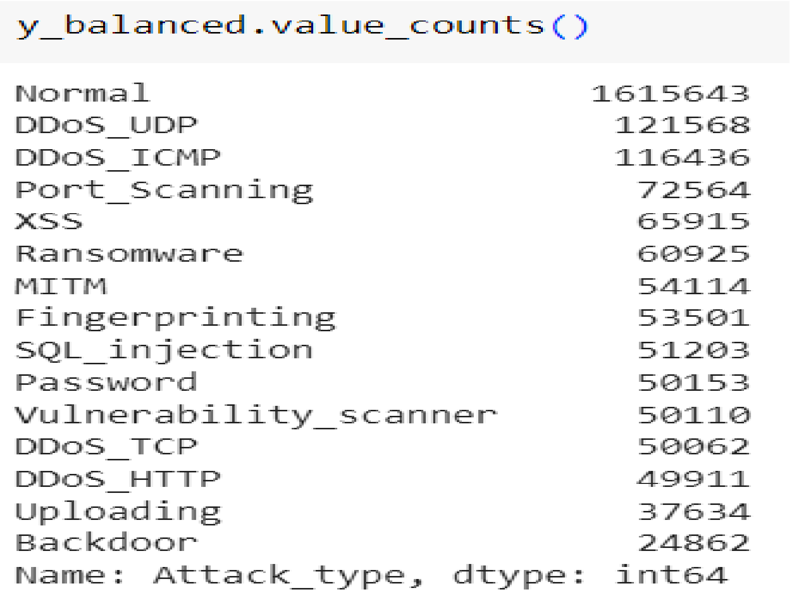


```
y.value_counts()

Normal                     1615643
DDoS_UDP                    121568
DDoS_ICMP                   116436
SQL_injection                51203
Password                     50153
Vulnerability_scanner        50110
DDoS_TCP                     50062
DDoS_HTTP                    49911
Uploading                    37634
Backdoor                     24862
Port_Scanning                22564
XSS                          15915
Ransomware                   10925
MITM                          4114
Fingerprinting                3501
Name: Attack_type, dtype: int64
```

Figure 3.2: Different type of attack count before sampling

In the figure 3.2, it shows the different count of attacks present inside the dataset. It has total of 15 features with normal and fourteen types of attacks which are possible for IoT and Industrial IoT devices.



```
y_balanced.value_counts()

Normal                     1615643
DDoS_UDP                    121568
DDoS_ICMP                   116436
Port_Scanning                72564
XSS                          65915
Ransomware                   60925
MITM                         54114
Fingerprinting               53501
SQL_injection                51203
Password                     50153
Vulnerability_scanner        50110
DDoS_TCP                     50062
DDoS_HTTP                    49911
Uploading                    37634
Backdoor                     24862
Name: Attack_type, dtype: int64
```

Figure 3.3: Different type of attack count after sampling

In the figure 3.3, it shows the count of different types of attacks after applying sampling method with random oversampling method.

3.2 Machine Learning Techniques Used

Machine learning is an area in which models are trained with data to predict accordingly. Machine learning techniques are mainly classified into three types of learning methods, supervised learning in which labeled data is been provided, unsupervised learning which containsiunlabeled data, and semi-supervised learning which contains both labeled as well unlabeled data for training [16]. Machine learning contains different types of algorithms like linear regression, support vector machine, and so on. These ml algorithms are used to predict any security threat inside the IoT devices [12]. Machine learning algorithms are being trained with labeled anomalies and which helps us to find any security threat happening inside the cloud. The main advantage of using machine learning algorithms is they become more accurate and precise by training on proper datasets.

A few Classifiers can be utilized to construct an attack prediction model like Bayesian Network algorithm, Multi-Layer Perceptron algorithm, Logistic Regression algorithm, Decision Trees algorithm, Radial Basis function network algorithm.

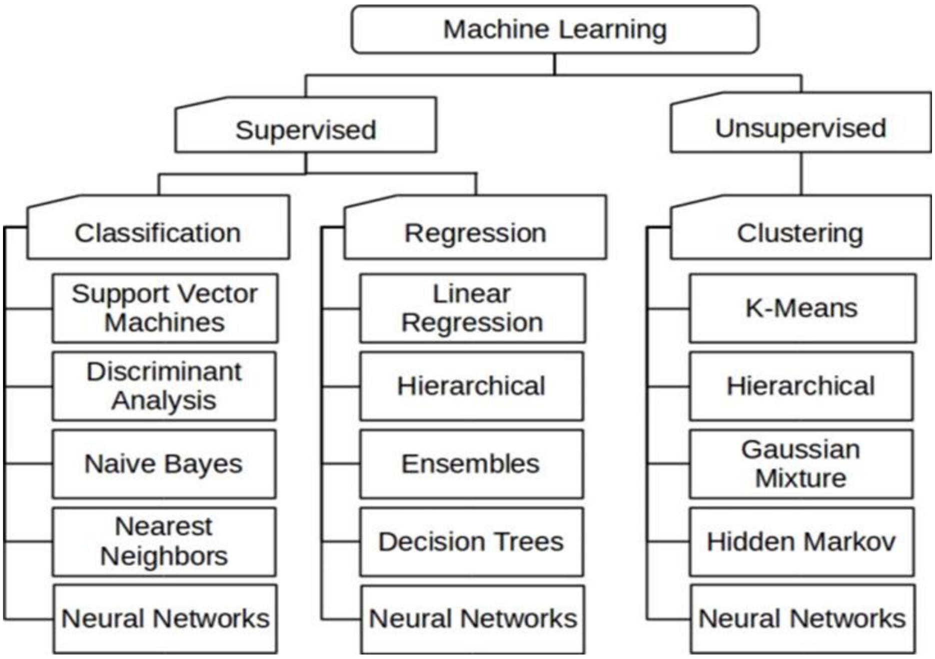


Figure 3.4: Different ML Techniques [5]

In the figure 3.4, it talks about the different types of machine leaning techniques which are classified under supervised and unsupervised learning.

In this study different ML classification techniques been used. All the techniques have been used are explained below.

3.2.1 Random Forest

Random Forest is a popular machine learning algorithm that is widely used for both classification and regression tasks. It is an ensemble learning method that combines the predictions of multiple decision trees to make accurate and robust like predictions. In a Random Forest, a collection of decision trees is trained on different subsets of the training data, with each of the tree using a random subset of all features. During training, each tree independently makes predictions, and the outcome prediction is determined by aggregating the predictions of all the trees. One of advantages of Random Forest is its capability to provide feature importance ranking, which helps identify the most influential features in the prediction process. This can aid in feature selection and understanding patterns in the data.

3.2.2 Decision Tree

Decision Tree (DT) is a popular machine learning technique used for classification. It is a non-parametric algorithm that builds a tree-like model of decisions based on the features in the input data. The Decision Tree algorithm works by recursively splitting the data based on the values of different features. At each split, it selects the feature that best separates the data into distinct classes or reduces the impurity of the classes. The algorithm continues to split the data until it reaches a stop criterion, such as reaching a maximum depth or a minimum number of samples at a node.

3.2.3 K-nearest Neighbor

K-Nearest Neighbors (KNN) is a popular machine learning technique used for both classification and regression tasks. It is a non-parametric algorithm that makes predictions based on the similarity of the input data points to the labeled data points in the training set. In KNN, the number "K" represents the number of nearest neighbors considered for making predictions. When a new instance needs to be classified, the algorithm calculates the distance between that instance and all the instances in the training set. The K nearest neighbors, based on the calculated distance, are then used to determine the class label of the new instance.

3.2.4 Support Vector machine

Support Vector Machines (SVM) is a popular machine learning algorithm used for both classification and regression tasks. It is a supervised learning algorithm that can effectively handle both linear and nonlinear data by finding an optimal hyperplane or decision boundary that separates different classes or predicts continuous values. The decision boundary is determined by the support vectors, and the margin represents the distance between the decision boundary and the support vectors.

3.2.5 Deep Neural Network

Deep Neural Networks (DNN) is a powerful machine learning algorithm that is widely used for solving complex tasks such as image recognition, natural language processing, and speech recognition. DNNs are a type of artificial neural network that consists of multiple hidden layers between the input and output layers. The architecture of a DNN allows it to learn hierarchical representations of the input data. Each hidden layer extracts and learns features from the previous layer, gradually building a hierarchy of increasingly abstract representations. This enables DNNs to capture intricate patterns and relationships in the data, making them capable of solving highly complex problems. DNNs utilize a large number of neurons and parameters, which enables them to learn from vast amounts of data and model highly nonlinear relationships. They can automatically learn feature representations from raw data, eliminating the need for manual feature engineering in many cases.

The training of DNNs involves two main steps: forward propagation and backpropagation. During forward progation, the input data is passed through the network, and the activations of each neuron are computed. The output of the network is compared to the true labels, and the error is calculated. In the backpropagation step, the error is propagated backward through the network, and the gradients of the weights and biases are computed. These gradients are then used to update the parameters of the network using optimization techniques like gradient descent. DNNs can have various activation functions, such as sigmoid, ReLU (Rectified Linear Unit), or SoftMax, which introduce nonlinearity into the network and enable it to model complex relationships. The choice of activation function depends on the task and the characteristics of the data.

CHAPTER 4

EXPERIMENTAL SETUP

This chapter provides information about software and hardware used for the model. It also talks about the flow of data execution and in next chapter we will see about the results.

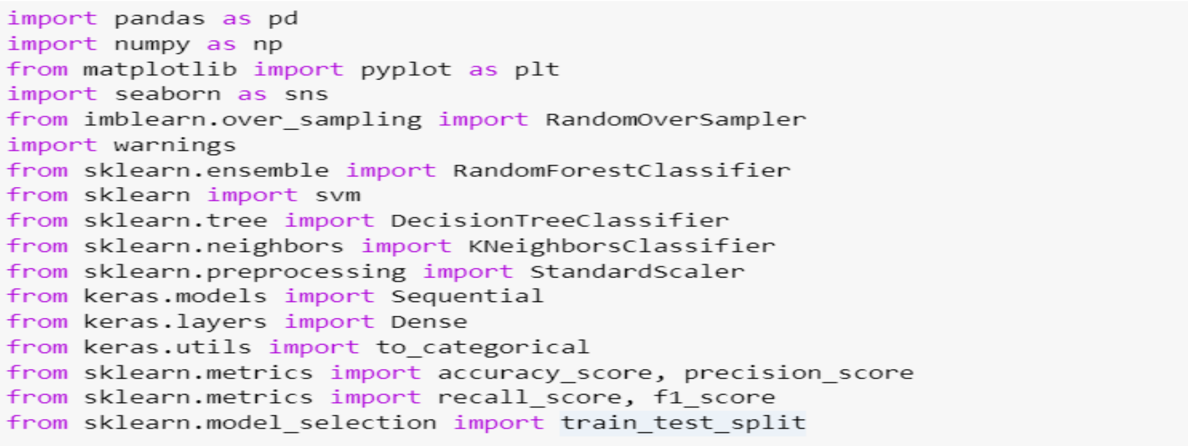
4.1 Hardware

The proposed work on machine learning techniques. It requires a laptop or pc of basic configuration provided below:

- System: Windows, Mac
- Processor: Core i5 processor
- RAM: 8 GB
- Hard Disk: 512 GB

4.2 Software

For implementation of the project it requires anaconda, Jupyter collab with python modules. All machine leaning model require python for implementation. All the packages have been installed with mount to drive for use of collab. In the figure 4.1, it shows all the library installed for the implementation of the model.



```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
from imblearn.over_sampling import RandomOverSampler
import warnings
from sklearn.ensemble import RandomForestClassifier
from sklearn import svm
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import StandardScaler
from keras.models import Sequential
from keras.layers import Dense
from keras.utils import to_categorical
from sklearn.metrics import accuracy_score, precision_score
from sklearn.metrics import recall_score, f1_score
from sklearn.model_selection import train_test_split
```

Figure 4.1: Library used by techniques

4.3 Workflow

In the figure 4.2, Framework of the model we have used in the project. It will start from data collection and then data preprocessing where we will remove Nan values and other corrupt values. The preprocessed data is splitted to train and test data. Then we perform different machine learning algorithms and we have compared the data.

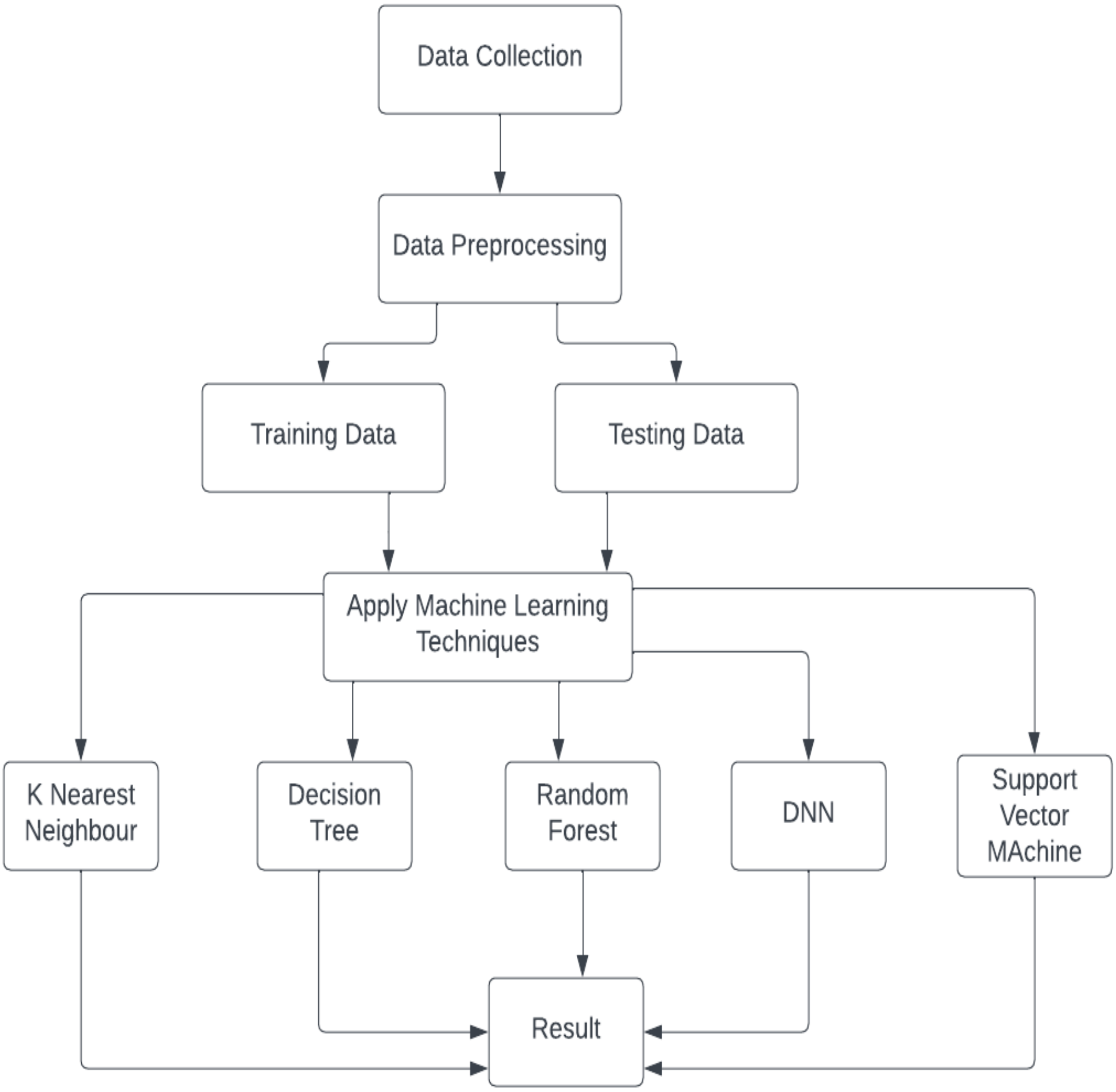


Figure 4.2: Workflow of project

4.4 Methodology

In this section we will define the workflow of the project. Dataset has been taken for industrial and IoT device for intrusion detection. Project has been implemented from based on 2 features as attack and normal. Dataset has been analyzed based on 15 features then apply different machine learning algorithm. These are the metrics which are analyzed for different ml algorithm.

- **Accuracy**: Accuracy is a commonly used evaluation metrics that measures the performance of a classification model. It represents the proportion of correctly classified instances out of total number of instances in a dataset. It has been shown in equation i.

$$Acc = \frac{TP_{Attack} + TN_{Normal}}{TP_{Attack} + TN_{Normal} + FP_{Normal} + FN_{Attack}} \quad \text{--------------------} \ \text{i}$$

- **Precision:** Precision is a performance metrics used to evaluate the quality of a classification model, particularly in binary classification tasks. It measures the proportion of true positive predictions (correctly predicted positive instances) out of the total instances predicted as positive by the model. It has been shown in equation ii.

$$\mathrm{Pr} = \frac{\mathrm{TP_{Attack}}}{\mathrm{TP_{Attack}} + \mathrm{FP_{Normal}}} \quad \text{--------------------} \ \text{ii}$$

- **Recall:** Recall measures the ability of a model to correctly identify positive instances while minimizing false negatives. A higher recall value indicates that the model has a better ability to capture positive instances, whereas a lower recall suggests that the model may be missing positive instances**.** It has been shown in equation iii.

$$Rc = \frac{TP_{Attack}}{TP_{Attack} + FN_{Attack}} \quad \text{--------------------} \ \text{iii}$$

CHAPTER 5

RESULTS AND DISCUSSION

This chapter gives data about result obtained on apply different machine learning algorithm on edge IoT dataset for intrusion detection. For comparison of different techniques it has been analyzed based on 2 features as attack or normal and based upon 15 features as normal, DDoS UDP, Backdoor, Port Scanning, DDoS ICMP, SQL injection, DDoS TCP, Vulnerability scanner, Ransomware, Fingerprinting, MITM Password, DDoS HTTP, Uploading, XSS.

Confusion matrix for 2 feature classification has been shown in figure 5.1 where one class denote with normal and other with attack. In this 2-feature extraction is plotted on the confusion matrix.

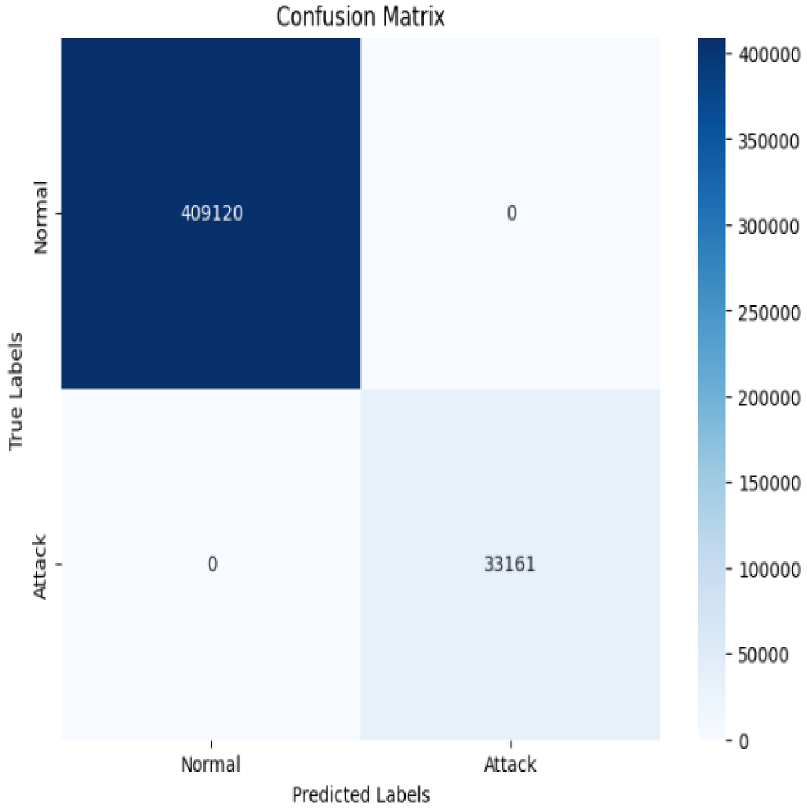
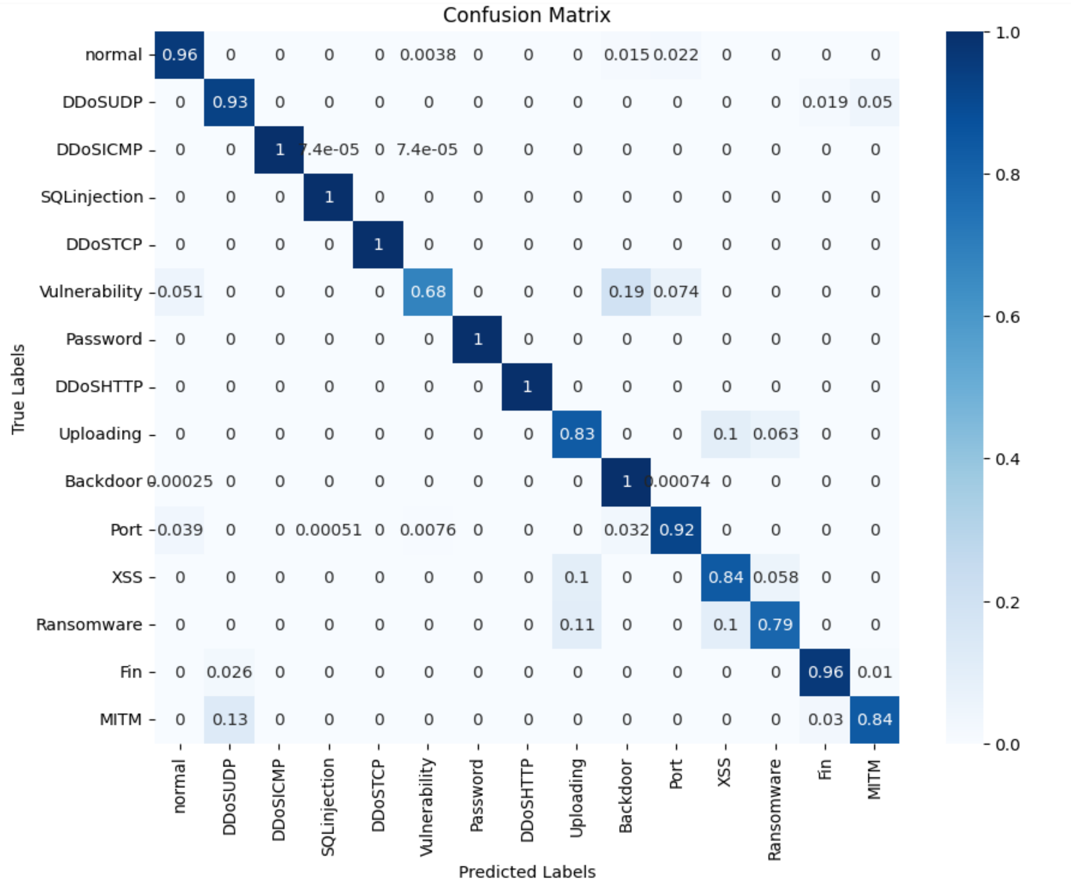


Figure 5.1: Confusion matrix for 2 feature classification



	normal	DDoSUDP	DDoSICMP	SQLinjection	DDoSTCP	Vulnerability	Password	DDoSHTTP	Uploading	Backdoor	Port	XSS	Ransomware	Fin	MITM
normal	0.96	0	0	0	0	0.0038	0	0	0	0.015	0.022	0	0	0	0
DDoSUDP	0	0.93	0	0	0	0	0	0	0	0	0	0	0	0.019	0.05
DDoSICMP	0	0	1	7.4e-05	0	7.4e-05	0	0	0	0	0	0	0	0	0
SQLinjection	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
DDoSTCP	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Vulnerability	0.051	0	0	0	0	0.68	0	0	0	0.19	0.074	0	0	0	0
Password	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
DDoSHTTP	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
Uploading	0	0	0	0	0	0	0	0	0.83	0	0	0.1	0.063	0	0
Backdoor	0.00025	0	0	0	0	0	0	0	0	1	0.00074	0	0	0	0
Port	0.039	0	0	0.00051	0	0.0076	0	0	0	0.032	0.92	0	0	0	0
XSS	0	0	0	0	0	0	0	0	0.1	0	0	0.84	0.058	0	0
Ransomware	0	0	0	0	0	0	0	0	0.11	0	0	0.1	0.79	0	0
Fin	0	0.026	0	0	0	0	0	0	0	0	0	0	0	0.96	0.01
MITM	0	0.13	0	0	0	0	0	0	0	0	0	0	0	0.03	0.84

Figure 5.2: Confusion matrix for 15 features

In the figure 5.2, all the four algorithms have been analyzed on the edge IOT dataset for 15 feature classification and confusion matrix has been shown. In the table 5.1 the accuracy of 2 feature classification ahs been shown.

Table 5.1: Accuracy for 2-feature classification

S.No.	Technique Used	Maximum Accuracy
1	Random Forest	99.98%
2	Support Vector Machine	99.99%
3	Decision Tree	99.99%
4	K-Nearest Neighbor	99.99%
5	Deep Neural Network	99.99%

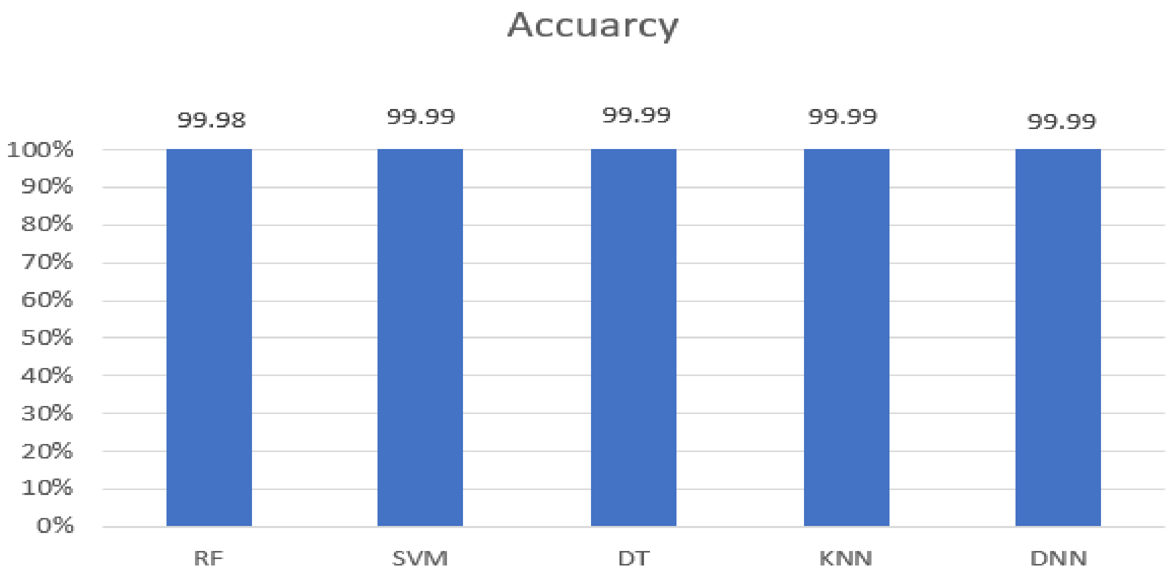


Figure 5.3: Accuracy for 2 class classification

In the Table 5.2, it shows accuracy score for 15 feature classification without sampling and applying the different machine leaning techniques which has been implemented. It also shows the accuracy score by using the random oversampling technique.

Table 5.2: Performance score for different techniques before and after sampling

S.No.	**Techniques used**	**Performance before Sampling**	**Performance after Sampling**
1	Random Forest	93.2	98.25
2	Support Vector Machine	77.8	79.31
3	Decision Tree	94.18	98.16
4	K-Nearest Neighbor	79.8	88.2
5	Deep Neural Network	94.2	94.92

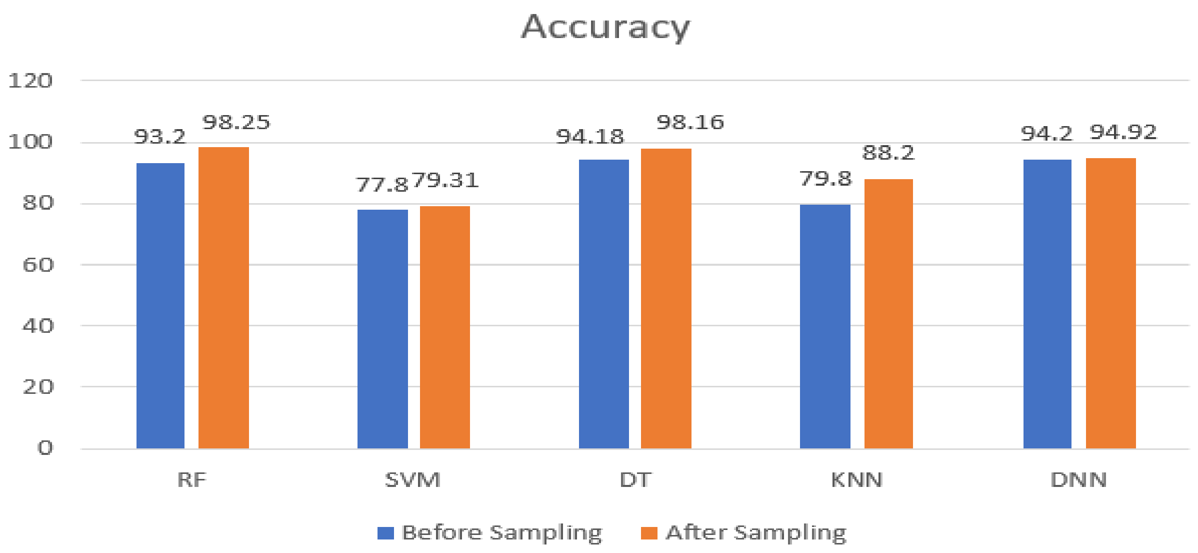


Figure 5.4: Accuracy for 15 feature classification with before and after sampling

In this project we have compared the accuracy before doing the random oversampling and SMOTE sampling with after sampling process as shown in figure 5.4. It has been found that accuracy has been increase with sampling process. In the sampling process we are tuning the minority classes.

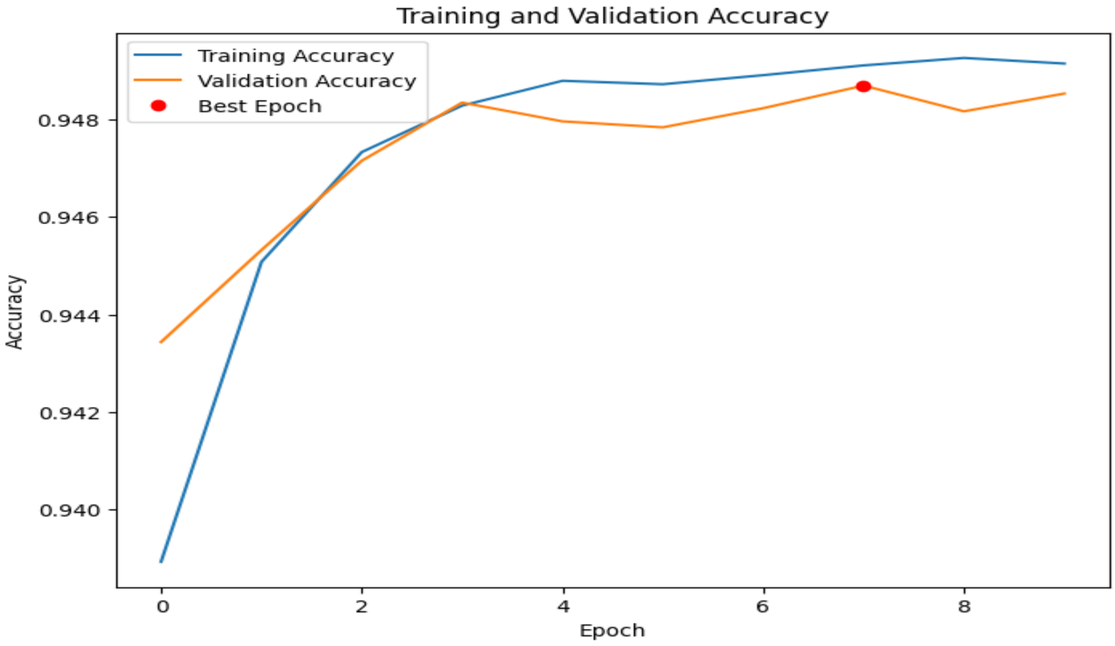


Figure 5.5: Training and validation accuracy

Now after validation of accuracy with training and validation of loss with training graph has been plotted and analysed. The graph plotting of the validation accuracy per epoch has been plotted on figure 5.5.

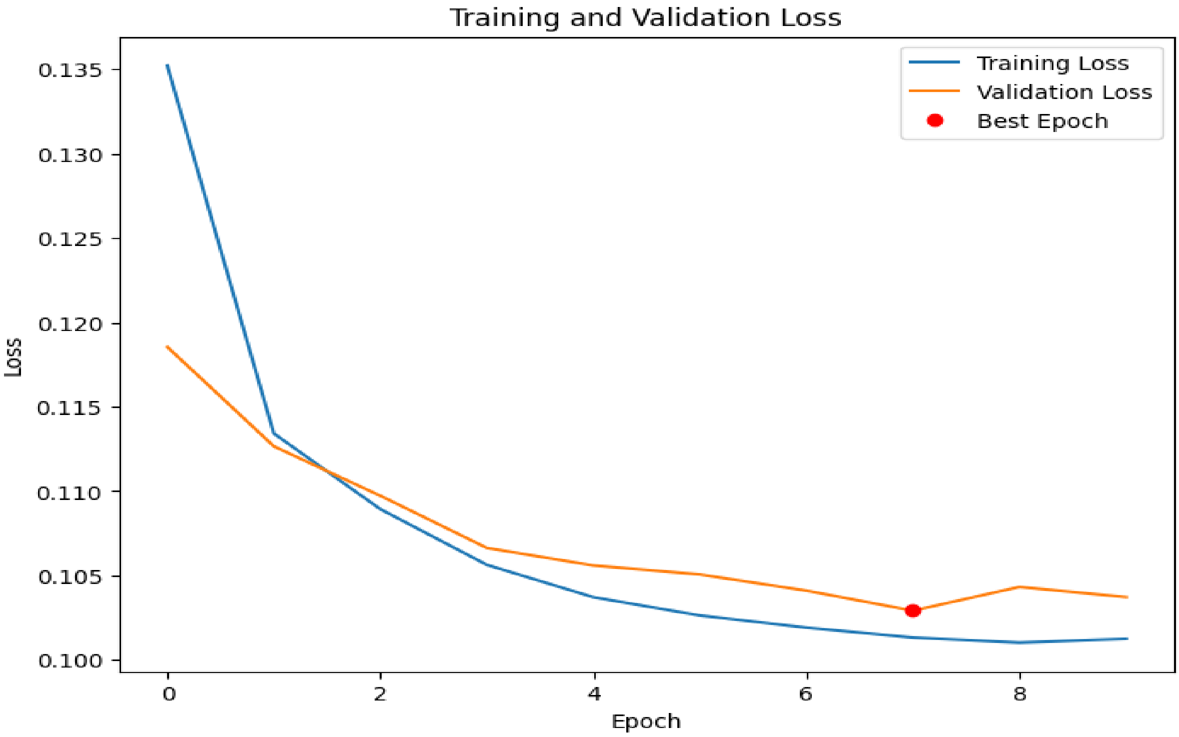


Figure 5.6: Training and Validation loss

All the machine learning models has been analysed and their plot has been done based upon validation accuracy and validation plot per epoch. It has been found that on the best epoch validation loss obtained is 0.105. The graph plot for the validation loss per epoch ahs been plotted on figure 5.6.

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this section talks about the summary, conclusion and future work about this topic. All the techniques have been analysed and accuracy is been compared based upon edge IoT dataset.

6.1 Conclusion

Securing IIoT devices is of utmost importance due to the critical nature of industrial IoT applications utilizing IoT technology. However, there is a significant gap in providing adequate security measures for these systems. While big data analytics and machine learning (ML) technologies have been mostly employed to ensure security in traditional ITssystems, the unique characteristics and priorities of Industrial Control Systems (ICS) necessitate specific attention and tailored security approaches. Recognizing this need, we have highlighted the value of ML algorithms in detecting attacks through presentations and experimental evaluations. By leveraging ML techniques, we aim to enhance the security of IIoT systems and address the distinct cyber risks associated with industrial applications. Different type of SVM, KNN, random forest, decision tree and DNN has been used for analysing the dataset.

For detection of attacks using machine learning techniques and deep learning algorithms we have found that accuracy of the approaches is from 77 to 94 percent based upon the techniques. For improvisation of accuracy, we have used sampling technique as SMOTE and Random Oversampling. By using the sampling technique, it has increased the accuracy to 78 to 98 percent. It has been found that random forest has highest accuracy over edge IoT dataset. Comparison has been made on different machine learning techniques.

6.2 Future Scope

The proposed methodology for intrusion detection using machine learning technique in IoT and industrial IoT devices has been implemented with different ml techniques. It can be implemented using other ml and deep leaning techniques so that accuracy can be increased. Ensemble learning techniques have shown promising results in various domains, including IoT datasets. No any ensemble technique has been implemented on edge IoT dataset.

Ensemble learning can enhance the accuracy of IoT datasets by combining predictions from multiple models. By leveraging diverse models and aggregation techniques such as bagging or boosting, ensemble methods can effectively reduce bias and variance, resulting in improved overall accuracy.

REFERENCES

1. J. Wu, L. Ping, X. Ge, Y. Wang and J. Fu, "Cloud Storage as the Infrastructure of Cloud Computing," 2010 International Conference on Intelligent Computing and Cognitive Informatics, 2010, pp. 380-383, doi: 10.1109/ICICCI.2010.119.

2. A. Hendre and K. P. Joshi, "A Semantic Approach to Cloud Security and Compliance," 2015 IEEE 8th International Conference on Cloud Computing, 2015, pp. 1081-1084, doi: 10.1109/CLOUD.2015.157.

3. T. Eltaeib and N. Islam, "Taxonomy of Challenges in Cloud Security," 2021 8th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2021 7th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), 2021, pp. 42-46, doi: 10.1109/CSCloud-EdgeCom52276.2021.00018.

4. M. Bahrami and M. Singhal, "A dynamic cloud computing platform for eHealth systems," 2015 17th International Conference on E-health Networking, Application & Services (HealthCom), 2015, pp. 435-438, doi: 10.1109/HealthCom.2015.7454539.

5. Z. Chkirbene, A. Erbad and R. Hamila, "A Combined Decision for Secure Cloud Computing Based on Machine Learning and Past Information," 2019 IEEE Wireless Communications and Networking Conference (WCNC), 2019, pp. 1-6, doi: 10.1109/WCNC.2019.8885566.

6. N. Sengupta, “Designing encryption and IDS for cloud security,” in Proceedings of the Second International Conference on Internet of things, Data and Cloud Computing, 2017.

7. A. Rukavitsyn, K. Borisenko and A. Shorov, "Self-learning method for DDoS detection model in cloud computing," 2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), 2017, pp. 544-547, doi: 10.1109/EIConRus.2017.7910612.

8. A. R. Wani, Q. P. Rana, U. Saxena and N. Pandey, "Analysis and Detection of DDoS Attacks on Cloud Computing Environment using Machine Learning Techniques," 2019 Amity International Conference on Artificial Intelligence (AICAI), 2019, pp. 870-875, doi: 10.1109/AICAI.2019.8701238.

9. X. Li, Y. Zhu, and J. Wang, “Secure naïve Bayesian classification over encrypted data in cloud,” in Provable Security, Cham: Springer International Publishing, 2016, pp. 130–150.

10. Z. He, T. Zhang and R. B. Lee, "Machine Learning Based DDoS Attack Detection from Source Side in Cloud," 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud), 2017, pp. 114-120.

11. A. Saied, R. E. Overill, and T. Radzik, “Detection of known and unknown DDoS attacks using Artificial Neural Networks,” Neurocomputing, vol. 172, pp. 385–393, 2016.

12. M. A. Zardari, L. T. Jung and N. Zakaria, "K-NN classifier for data confidentiality in cloud computing," 2014 International Conference on Computer and Information Sciences (ICCOINS), 2014, pp. 1-6, doi: 10.1109/ICCOINS.2014.6868432.

13. T. Salman, D. Bhamare, A. Erbad, R. Jain and M. Samaka, "Machine Learning for Anomaly Detection and Categorization in Multi-Cloud Environments," 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud), 2017, pp. 97-103, doi: 10.1109/CSCloud.2017.15.

14. R. Kumar, S. P. Lal and A. Sharma, "Detecting Denial of Service Attacks in the Cloud," 2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), 2016, pp. 309-316, doi: 10.1109/DASC-PICom-DataCom-CyberSciTec.2016.70.

15. B. Gulmezoglu, T. Eisenbarth, and B. Sunar, “Cache-based application detection in the cloud using machine learning,” in Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security - ASIA CCS ’17, 2017.

16. Ehsan Hesamifard, Hassan Takabi, Mehdi Ghasemi, and Catherine Jones. 2017. Privacy-preserving Machine Learning in Cloud. In <i>Proceedings of the 2017 on Cloud Computing Security Workshop</i> (<i>CCSW '17</i>). Association for Computing Machinery, New York, NY, USA, 39–43. https://doi.org/10.1145/3140649.31406557.

17. C. Modi, D. Patel, B. Borisanya, A. Patel, and M. Rajarajan, “A novel framework for intrusion detection in cloud,” in Proceedings of the Fifth International Conference on Security of Information and Networks - SIN ’12, 2012.

18. M. Marwan, A. Kartit, and H. Ouahmane, “Security enhancement in healthcare cloud using machine learning,” Procedia Comput. Sci., vol. 127, pp. 388–397, 2018.

19. Z. Chen, F. Jiang, Y. Cheng, X. Gu, W. Liu and J. Peng, "XGBoost Classifier for DDoS Attack Detection and Analysis in SDN-Based Cloud," 2018 IEEE International Conference on Big Data and Smart Computing (BigComp), 2018, pp. 251-256, doi: 10.1109/BigComp.2018.00044.

20. J. Grover, Shikha and M. Sharma, "Cloud computing and its security issues — A review," Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2014, pp. 1-5, doi: 10.1109/ICCCNT.2014.6962991.

21. M. Mohammadi, A. Al-Fuqaha, S. Sorour and M. Guizani, "Deep Learning for IoT Big Data and Streaming Analytics: A Survey," in IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 2923-2960, Fourthquarter 2018, doi: 10.1109/COMST.2018.2844341.

22. Y. Meidan et al., "N-BaIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders," in IEEE Pervasive Computing, vol. 17, no. 3, pp. 12-22, Jul.-Sep. 2018, doi: 10.1109/MPRV.2018.03367731.

23. Kumar, P., Gupta, G.P. & Tripathi, R. A distributed ensemble design-based intrusion detection system using fog computing to protect the internet of things networks. J Ambient Intell Human Comput 12, 9555–9572 (2021). https://doi.org/10.1007/s12652-020-02696-3.

24. Deepti Rani, Nasib Singh Gill, Preeti Gulia, Jyotir Moy Chatterjee, "An Ensemble-Based Multiclass Classifier for Intrusion Detection Using Internet of Things", Computational Intelligence and Neuroscience, vol. 2022, Article ID 1668676, 16 pages, 2022. https://doi.org/10.1155/2022/1668676.

25. M. Zolanvari, M. A. Teixeira, L. Gupta, K. M. Khan and R. Jain, "Machine Learning-Based Network Vulnerability Analysis of Industrial Internet of Things," in IEEE Internet of Things Journal, vol. 6, no. 4, pp. 6822-6834, Aug. 2019, doi: 10.1109/JIOT.2019.2912022.

26. M. Özçelik, N. Chalabianloo and G. Gür, "Software-Defined Edge Defense Against IoT-Based DDoS," 2017 IEEE International Conference on Computer and Information Technology (CIT), Helsinki, Finland, 2017, pp. 308-313, doi: 10.1109/CIT.2017.61.

27. Morris, T., Gao, W. (2014). Industrial Control System Traffic Data Sets for Intrusion Detection Research. In: Butts, J., Shenoi, S. (eds) Critical Infrastructure Protection VIII. ICCIP 2014. IFIP Advances in Information and Communication Technology, vol 441. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-45355-1_5.

28. I. A. Siddavatam, S. Satish, W. Mahesh and F. Kazi, "An ensemble learning for anomaly identification in SCADA system," 2017 7th International Conference on Power Systems (ICPS), Pune, India, 2017, pp. 457-462, doi: 10.1109/ICPES.2017.8387337.

29. E. Sisinni, A. Saifullah, S. Han, U. Jennehag and M. Gidlund, "Industrial Internet of Things: Challenges, Opportunities, and Directions," in IEEE Transactions on Industrial Informatics, vol. 14, no. 11, pp. 4724-4734, Nov. 2018, doi: 10.1109/TII.2018.2852491.

30. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 29(7), 1645-1660. https://doi.org/10.1016/j.future.2013.01.0.0.

turnitin

Similarity Report ID: oid:27535:36493143

PAPER NAME

Aditya_Thesis.pdf

WORD COUNT 10436 Words	CHARACTER COUNT 59377 Characters
PAGE COUNT 48 Pages	FILE SIZE 763.7KB
SUBMISSION DATE May 30, 2023 12:05 PM GMT+5:30	REPORT DATE May 30, 2023 12:06 PM GMT+5:30

● **12% Overall Similarity**

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

- 6% Internet database
- Crossref database
- 10% Submitted Works database
- 3% Publications database
- Crossref Posted Content database

● **Excluded from Similarity Report**

- Bibliographic material
- Cited material
- Manually excluded text blocks
- Quoted material
- Small Matches (Less then 8 words)