People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research University of Mohamed Khider – BISKRA
Faculty of Exact Sciences, Science of Nature and Life
Computer Science Department

Ordre N° : IA9 /M2/2021

Memory Thesis

Submitted in fulfilment of the requirements for the Masters degree in

Computer Science

Option: **Artificial Intelligence**

Machine learning for early detection of IoT Botnets

By :

MEDGHAGHET SELSSABIL

Members of the jury :

Ayad Soheyb	MCA	President
Sahraoui Somia	MCB	Supervisor
Megague khadidja		Member

Session 2021

Dedication

I dedicate this modest work to:

My dear family,

My teacher on computer science department,

My friends and classmates,

To everyone who ever supported me in my life.

Dedicated to my precious parents and siblings.

Acknowledgements

First and foremost, praises and thanks to the God, my creator, my source of inspiration, knowledge and understanding, the Almighty, for His blessings throughout my research work to complete the research successfully.

I would like to express my deep and sincere gratitude to my research supervisor **Dr. Sahraoui Somia** for her immense knowledge, precious guidance and helping during our work.I really admire her support and understanding all along the way.I extend my appreciation to all teachers of the **Computer Science Department** who helped me in my education.

I'm extremely grateful to my parents for their love, care, encouragements and prayers. Special thanks to all my family members, sisters, brothers for contributing by their love and support,to my precious Nieces:Aseel,Retal,Layan and nephew Mohammed.

My thanks also go to my precious friends, classmates.

To everyone i love and care about, thank you.

Medghaghet Selssabil

Abstract

As the number of Internet of Things (IoT) devices connected to the network rapidly increases, network attacks such DDos botnet.These attacks cause network disruption and denial of service to IoT devices.However, a large number of heterogenous devices deployed in the IoT environment make it difficult to detect IoT attacks using traditional rule-based security solutions.It is challenging to develop optimal security models for each type of the device.Machine learning (ML) is an alternative technique that allows one to develop optimal security models based on empirical data from each device.Our proposed model tackles the security issue concerning the threats from bots.Different machine learning algorithms such as KNN,Decision Tree,Logistic Regression and BernoulliNB, were used to develop a model where data are trained by BoT-IoT dataset.Based on the findings, Decision Tree and Logistic Regression algorithm was found to be the most reliable in botnet detection with 99.99% accuracy and 99.99% ROC_AUC for both.

Keywords: *internet of things; botnet attacks; machine learning.*

Résumé

Alors que le nombre d'appareils Internet des objets (IoT) connectés au réseau augmente rapidement, les attaques de réseau telles que le botnet DDos. Ces attaques provoquent des perturbations du réseau et un déni de service aux appareils IoT.Cependant, un grand nombre d'appareils hétérogènes déployés dans l'environnement IoT rendent difficile la détection des attaques IoT à l'aide de solutions de sécurité traditionnelles basées sur des règles.Il est difficile de développer des modèles de sécurité optimaux pour chaque type d'appareil.L'apprentissage automatique (ML) est une technique alternative qui permet de développer des modèles de sécurité optimaux basés sur les données empiriques de chaque appareil.Notre modèle proposé aborde le problème de sécurité concernant les menaces des robots. Différents algorithmes d'apprentissage automatique tels que KNN, Arbre de décision, Régression logistique et BernoulliNB ont été utilisés pour développer un modèle où les données sont entraînées par l'ensemble de données BoT-IoT.Le meilleur algorithme a été sélectionné par un point de référence basé sur le pourcentage de précision et l'aire sous le score de la courbe des caractéristiques de fonctionnement du récepteur (ROC AUC). L'ingénierie des caractéristiques et la technique de suréchantillonnage des minorités synthétiques (SMOTE) ont été combinées avec des algorithmes d'apprentissage automatique (MLA).Sur la base des résultats, l'algorithme de régression logistique de l'arbre de décision s'est avéré le plus fiable dans la détection de botnets avec une précision de 99,99% et un ROC_AUC de 99,99% pour les deux.

Mots clés *:Internet des objets; attaques de botnets ; apprentissage automatique.*

ملخص

نظرًا للزيادة السريعة في عدد أجهزة إنترنت الأشياء (IoT) المتصلة بالشبكة، هجمات الشبكة مثل(DDos botnet) ، تتسبب هذه الهجمات في تعطيل الشبكة وحرمان أجهزة إنترنت الأشياء من الخدمة. تجعل بيئة إنترنت الأشياء من الصعب اكتشاف هجمات إنترنت الأشياء باستخدام حلول الأمان التقليدية المستندة إلى القواعد، ومن الصعب تطوير نماذج أمان مثالية لكل نوع من الأجهزة. التعلم الآلي (ML) هو أسلوب بديل يسمح للفرد بتطوير نماذج أمان مثالية تعتمد على على البيانات التجريبية من كل جهاز ، يعالج نموذجنا المقترح مشكلة الأمان المتعلقة بالتهديدات من الروبوتات ، وقد تم استخدام خوارزميات مختلفة للتعلم الآلي مثل (KNN) و (Decision Tree) و Logistic) (Regressionو (BernoulliNB) لتطوير نموذج حيث يتم تدريب البيانات بواسطة مجموعة بيانات BoT-IoT ، واستنادًا إلى النتائج ، تم العثور على خوارزمية (Decision Tree) و (Logistic Regression)الأكثر موثوقية في اكتشاف الروبوتات بنسبة 99.99٪ (accuracy)و 99.99٪ (ROC_AUC)لكليهما.

الكلمات المفتاحية: إنترنت الأشياء؛ هجمات الروبوتات التعلم الالي.

List of Figures

1.1 IoT Applications. 5
1.2 Three-layered IoT architecture. 5
1.3 RFID technology. 6
1.4 6LoWPAN technology. 7
1.5 BLE technology. 8

2.1 Man in the middle attacks. 11
2.2 Sybil Attack. 12
2.3 RFID Cloning 13
2.4 Replay attack 14
2.5 Ddos Attack [41]. 15
2.6 Botnet attack. 15
2.7 Spear Phishing Attack. 16
2.8 Malicious Code Injection. 17
2.9 Sniffing Attack. 17
2.10 Illustration of a centralized Botnet[33]. 20
2.11 P2P Architecture [35]. 21
2.12 Mirai workflow [32]. 22

3.1 Machine Learning Algorithms [64]. 25
3.2 Components of a Generic Machine Learning model [46]. 28
3.3 The Reinforcement Learning Model [54]. 30
3.4 Decision Tree [43]. 32

3.5 Logistic function transformation example [64]. 33

4.1 Proposed Model. 39
4.2 Visual Representation of Extra Trees Classifier [82]. 41
4.3 The Synthetic Minority Oversampling Technique (SMOTE) [73]. 42
4.4 Confusion matrix, illustrating the calculation of precision, recall,and F1-score [83]. 44
4.5 K-fold cross validation with 5 folds [81]. 45
4.6 Size of dataset. 46
4.7 Size of dataset. 46
4.8 Bar graph showing class imbalance in data-set. 46
4.9 First 7 rows of dataset before transformation and Normalization. 47
4.10 Before Data Transformation. 47
4.11 First 7 rows of dataset after transformation. 48
4.12 After Data Transformation. 48
4.13 First 7 rows of dataset after Normalization. 48
4.14 Size of dataset before and after SMOTE technique. 49
4.15 Top 10 respective feature score. 49
4.16 Feature Description. 50
4.17 BernoulliNB model performance without using SMOTE technique. 51
4.18 BernoulliNB model performance without using SMOTE technique. 51
4.19 BernoulliNB model performance using SMOTE technique. 51
4.20 matrix confusion for BernoulliNB without using SMOTE technique. 52
4.21 matrix confusion for BernoulliNB using SMOTE technique. 52
4.22 ROC AUC graph from BernoulliNB model without using SMOTE Technique. 53
4.23 ROC AUC graph from BernoulliNB model using SMOTE Technique. 53
4.24 BernoulliNB model performance using SMOTE technique (8 Feature). 54
4.25 matrix confusion for BernoulliNB using SMOTE technique (8 Feature). 54
4.26 ROC AUC graph from BernoulliNB model using SMOTE Technique (8 Feature). . . 55
4.27 KNN model performance without using SMOTE technique. 55
4.28 KNN model performance using SMOTE technique. 56

4.29 matrix confusion for KNN model without using SMOTE technique. 56
4.30 matrix confusion for KNN model using SMOTE technique. 57
4.31 ROC AUC graph from KNN model without using SMOTE Technique. 57
4.32 ROC AUC graph from KNN model using SMOTE Technique. 58
4.33 KNN model performance using SMOTE technique (8 Feature). 58
4.34 matrix confusion for KNN using SMOTE technique (8 Feature). 59
4.35 ROC AUC graph from KNN model using SMOTE Technique (8 Feature). 59
4.36 DecisionTreeClassifier model performance without using SMOTE technique. . . . 60
4.37 DecisionTreeClassifier model performance using SMOTE technique. 60
4.38 matrix confusion for DecisionTreeClassifier model without using SMOTE technique. 61
4.39 matrix confusion for DecisionTreeClassifier model using SMOTE technique. 61
4.40 ROC AUC graph from DecisionTreeClassifier model without using SMOTE Technique. 62
4.41 ROC AUC graph from DecisionTreeClassifier model using SMOTE Technique. . . . 62
4.42 DecisionTreeClassifier model performance using SMOTE technique (8 Feature). . 63
4.43 matrix confusion for DecisionTreeClassifier using SMOTE technique (8 Feature). . 63
4.44 ROC AUC graph from DecisionTreeClassifier model using SMOTE Technique (8 Feature). 64
4.45 LogisticRegression model performance without using SMOTE technique. 64
4.46 LogisticRegression model performance using SMOTE technique. 65
4.47 matrix confusion for LogisticRegression model without using SMOTE technique. . 65
4.48 matrix confusion for LogisticRegression model using SMOTE technique. 66
4.49 ROC AUC graph from LogisticRegression model without using SMOTE Technique. 66
4.50 ROC AUC graph from LogisticRegression model using SMOTE Technique. 66
4.51 LogisticRegression model performance using SMOTE technique (8 Feature). 67
4.52 matrix confusion for LogisticRegression using SMOTE technique (8 Feature). . . . 67
4.53 ROC AUC graph from LogisticRegression model using SMOTE Technique (8 Feature). 68
4.54 Comparison of MLAs performance. 69
4.55 Comparison of MLAs performance. 69
4.56 Graphical presentation of machine learning Accuracy performance comparison. . 69

4.57 Graphical presentation of machine learning ROC_AUC performance comparison. 70
4.58 Graphical presentation of machine learning precision performance comparison. . 70
4.59 Graphical presentation of machine learning recall performance comparison. . . . 71
4.60 Graphical presentation of machine learning f1-score performance comparison. . . 71
4.61 Graphical presentation of machine learning Accuracy performance comparison. . 72
4.62 Graphical presentation of machine learning ROC_AUC performance comparison. 72
4.63 Graphical presentation of machine learning precision performance comparison. . 73
4.64 Graphical presentation of machine learning recall performance comparison. . . . 73
4.65 Graphical presentation of machine learning f1-score performance comparison. . . 74

List of source code

List of Tables

3.1 Development of ML. 26

Contents

Dedication ii

Acknowledgements iii

Abstract iv

Résumé v

List of Figures **xii**

List of Tables **xiii**

1 Generalities on IoT **3**

1.1 Introduction 3

1.2 Internet of things definition 3

1.3 IoT Applications 4

1.4 IOT architecture 5

1.4.1 Perception Layer 6

1.4.2 Network Layer 6

1.4.3 Application layer 8

1.5 Conclusion 9

2 Presentation of security issues in IoT: A focus on Botnets **10**

2.1 Introduction 10

2.2 IoT security attacks 10

2.3 IoT security threats 17

2.3.1 Preception Layer Attacks 18

2.3.2 Network Layer Attacks 18

2.3.3 Application layer Attacks . . . 18
2.4 Overview of IoT Botnet . . . 19
2.4.1 Types of Botnet based on the architecture . . . 19
2.4.2 Mirai . . . 21
2.5 Security concepts . . . 22
2.6 Conclusion . . . 23

3 Study of the proposed solutions to overcome Botnets 24
3.1 Introduction . . . 24
3.2 Definition of Machine Learning . . . 24
3.3 Development of Machine Learning . . . 25
3.4 The Generic Model of Machine Learning . . . 27
3.5 Machine Learning Paradigms . . . 29
3.5.1 supervised learning . . . 29
3.5.2 Unsupervised Learning . . . 29
3.5.3 Reinforcement Learning . . . 30
3.5.4 Artificial Neural Network . . . 31
3.6 Machine Learning Algorithms . . . 31
3.6.1 Supervised Learning . . . 31
3.6.2 Unsupervised Learning . . . 35
3.7 Applications of Machine Learning . . . 35
3.8 Conclusion . . . 35

4 Scope on the realized solution 37
4.1 Introduction . . . 37
4.2 Related Works . . . 37
4.3 Research approach . . . 38
4.3.1 BoT-IoT dataset . . . 39
4.3.2 Data preprocessing . . . 40
4.3.3 Feature Engineering . . . 41
4.3.4 Synthetic Minority over-sampling technique(SMOTE) . . . 42

4.3.5 Experimental Scenario 42
4.3.6 Machine Learning Model Evaluation and Cross validation 43
4.4 Experiment and Discussion 45
4.4.1 Transformation 47
4.4.2 Oversampling minority class 49
4.4.3 Feature Score 49
4.4.4 Train-Test Split 50
4.4.5 Comparison of performance of Machine learning algorithms 50
4.4.6 Observations: 68
4.5 Conclusion 74

References **76**

General Introduction

General context

The Internet of Things (IoT) represents the concept of a massive system where things on the Internet communicate through omnipresent sensors.Since the inception of the Internet of Things, consumers have connected smart devices to the network at an exponential rate,bringing us closer to a future where everyday things all interconnect.

According to the latest research report, the smart home market is expected to be valued at US$137.91 billion by 2023, growing at a compound annual growth rate (CAGR) of 13 percent between 2017 and 2023 [89].

It is comprised of a wildly diverse range of device types- from small to large, from simple to complex – from consumer gadgets to sophisticated systems found in DoD (department of defense), utility and industrial systems.The IoT enables the exchange of information in a variety of application scenarios, each having unique characteristics and requiring unique performance guarantees, and together they bring potentially tremendous benefits to human being, such as: home automation, environmental monitoring, health and lifestyle, smart cities, etc.

Problematic and Objectives

In a study by Hewlett Packard in 2015, it was shown that out of a number of IoT devices that were investigated, 80% raised privacy concerns, with 60% lacking any mechanisms that verify the authenticity of security updates or even their integrity, allowing an adversary to modify the firmware without being noticed [90]. Seeing as IoT devices are manufactured with various pre existing inherent limitations and vulnerabilities, it should come as no surprise that they have been targeted and recruited by botnets.In 2016, Anna Senpai created a malicious program,

called Mirai, which is possible to take control of vulnerable connected objects such as surveillance cameras and routers, and generate distributed denial of service attacks (DDoS) [11].Mirai transforms the infected objects into autonomous and intelligent agents that are controlled remotely.

In order to solve the above problems, the way of DDoS attacks detection through machine learning algorithms has gradually become the focus of research.The machine learning algorithm can find out the abnormal information behind the massive data.Many detection approaches have been proposed to detect the DoS Attacks.

In this thesis, Bot-Iot dataset is used for the experiment, and Machine learning algorithms such as KNN,Decision Tree,Logistic Regression and BernoulliNB are used to detect it.We have done the experiment on real time data-set and balanced dataset and presented the effect of imbalance data and its impact on machine learning.

Outlines

This dissertation is organized as follows:

- **Chapter 1**defines the concept of IoT, presents the enabling technologies that motivate the emergence of IoT, and introduces common applications and elements of IoT.
- **Chapter 2**presents the different IoT security attacks, their definitions and purposes.
- **Chapter 3** introduces the concepts and techniques of machine learning.
- **Chapter 4** presents the experimental results and discussion.
- **conclusion** conclusion general.

Chapter 1

Generalities on IoT

1.1 Introduction

Over the past few years,the IoT has gained significant attention since it brings potentially tremendous benefits to the human.The concept of the IoT has been introduced by Kevin Ashton in 1999,it aims to connect anything at anytime in anyplace.”Things” in IoT are embedded with sensing, processing and actuating capabilities and cooperate with each other to provide smart and innovative services autonomously.The IoT spans many diverse application domains such as home automation, environmental monitoring, healthcare, and so on.The primary objective of the IoT is unification of these numerous diverse application domains under the same umbrella referred as smart life.The architecture of IoT supports a large number of heterogeneous devices and integrates various communication technologies that enable the connectivity of IoT devices to provide the required services to end-users.The present chapter provides an overview of fundamental concepts of IoT.It introduces the IoT definition, potential applications and architecture including major elements and protocols used in IoT [11].

1.2 Internet of things definition

Internet of things (IoT) is a collection of many interconnected objects,services, humans,and devices that can communicate,share data,and information to achieve a common goal in different areas and applications [1].

1.3 IoT Applications

The IoT provides a large number of applications to enhance peoples' daily lives and activities.

- **Home automation:** or smart building is called as domotics.By using a centralized hub generally a smart phone (which contains sensors like accelerometer),the various things in the home can be controlled.That is,smart television,air conditioner,water heaters,lights, fans etc.. will be connected to the smart phone using NFC ,Bluetooth,Zigbee or any other short range low power protocols [2].

- **Fleet management:** The manger of the fleet can schedule the vehicles and drivers based on the business requirements and the real-time position information collected by the vehicles [3].

- **Agriculture Smart farms:**are a fact.The quality of soil is crucial to produce good crops, and the Internet of Things offers farmers the possibility to access detailed knowledge and valuable information of their soil condition.Through the implementation of IoT sensors,a significant amount of data can be obtained on the state and stages of the soil.Information such as soil moisture,level of acidity,other chemical characteristics,helps farmers control irrigation,make water use more efficient, specify the best times to start sowing,and even discover the presence of diseases in plants and soil [4].

- **Remote medical monitoring:** IoT can analyze the recur-ring indicator data collected from the device placed on patients' body and provide the users with health trends and health advice [3].

- **Smart Cities:** The IoT has the potential to transform entire cities by solving real problems citizens face each day.With the proper connections and data,the Internet of Things can solve traffic congestion issues and reduce noise,crime,and pollution [5].

Figure 1.1: IoT Applications.

1.4 IOT architecture

The architecture of IoT is not standardized, typical IoT architecture has three layers: perception, network and application [12] as shown in Fig. 1.2.

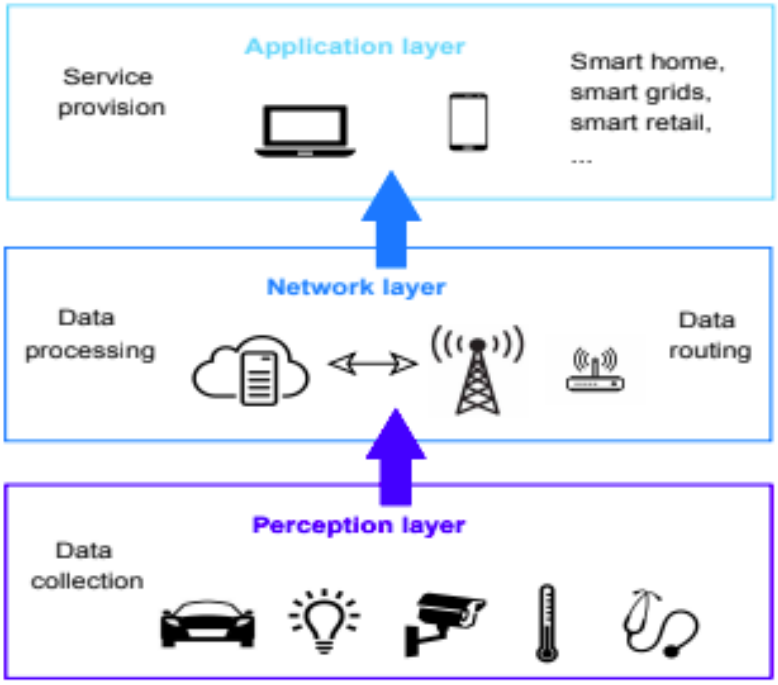

Figure 1.2: Three-layered IoT architecture.

1.4.1 Perception Layer

The perception layer is also known as the “Sensors” layer in IoT.The purpose of this layer is to acquire the data from the environment with the help of sensors and actuators.This layer detects,collects,and processes information and then transmits it to the network layer.This layer also performs the IoT node collaboration in local and short range networks [1].

Radio frequency identification (RFID)

Radio Frequency Identification (RFID) is a type of passive wireless technology that allows for tracking or matching of an item or individual.It has three parts:tags,reader and a database. The tags are attached to the objects and read the state of the objects while a reader is used to read the information from tags [6].

Figure 1.3: RFID technology.

Wireless Sensor Networks

Wireless Sensor Network (WSN) technology uses interconnected smart devices for sensing and monitoring.Its applications include environmental monitoring,medical monitoring,industrial monitoring,traffic monitoring,etc [7].

1.4.2 Network Layer

The network layer of IoT serves the function of data routing and transmission to different IoT hubs and devices over the Internet.At this layer,cloud computing platforms,Internet gateways,switching,and routing devices etc.operate by using some of the very recent technologies such as WiFi,LTE,Bluetooth,3G,Zigbe

etc.The network gateways serve as the mediator between different IoT nodes by aggregating,filtering,and transmitting data to and from different sensors [1].

6LoWPAN

6LoWPAN is a somewhat contorted acronym that combines the latest version of the Internet Protocol (IPv6) and Low-power Wireless Personal Area Networks (LoWPAN).6LoWPAN,therefore, allows for the smallest devices with limited processing ability to transmit information wirelessly using an internet protocol.It's the newest competitor to ZigBee. The concept was created because engineers felt like the smallest devices were being left out from the Internet of Things.6LoWPAN can communicate with 802.15.4 devices as well as other types of devices on an IP network link like WiFi.A bridge device can connect the two [8].

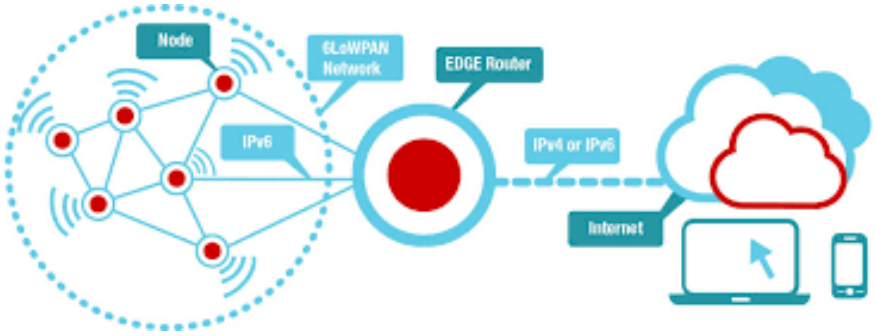

Figure 1.4: 6LoWPAN technology.

ZigBee

Zigbee is follows IEEE 802.15.4 standard.It is a short range (around 20meters) protocol, used to create a small network.It is generally used in home automation [3].

BLE

BLE has a simpler design than Bluetooth,focusing mainly on health and medical applications.Compared to Classic Bluetooth,BLE is intended to provide considerably reduced power consumption and cost,while maintaining a similar communication range.The ultra-low power requirement of BLE makes it ideal for small devices,including wearable technologies,in which minimal battery life requirement and small form factor are critical design and engineering considerations [9].

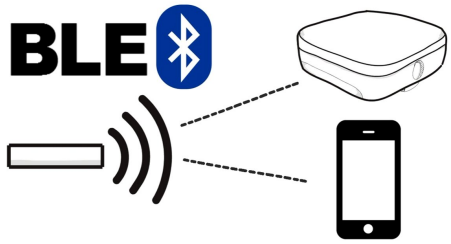

Figure 1.5: BLE technology.

LoRaWAN

Long range wide area network (LoRaWAN) has the potential for long range communications in sustainable health IoT applications (e.g., the LoRa) It also supports 868 MHz frequency and 915 MHz spectrum operation with data rates of more than 50 kb/s [10].

1.4.3 Application layer

Application layer defines all applications that use the IoT technology or in which IoT has deployed.The applications of IoT can be smart homes, smart cities,smart health,animal tracking, etc.It has the responsibility to provide the services to the applications.The services may be varying for each application because services depend on the information that is collected by sensors [6].

CoAP

Since IoT devices are resource-constrained, HTTP protocol is not suitable for low power devices due to its complexity.CoAP was designed to include features of HTTP dedicated to IoT devices.CoAP is a messaging protocol based on representational state transfer (REST) architecture. It has four message types: confirm-able, non confirm-able, acknowledgment and reset. It provides features that are not available on HTTP such as push notification(i.e., the server can store the list of devices) [11].

MQTT

MQTT (Message Queue Telemetry Transport) is a machine to machine IoT protocol.It is used in cloud based and fog based architectures.The important feature of this protocol is,it can transmit information from one source to many users (one to many function) via an intermediate node [3].

1.5 Conclusion

Connected Things have become pervasive for every individual. In fact, the IoT benefits has human life evolving with the Things. IoT is in smart cities (e.g. smart parking), smart environment (e.g. for air pollution), in smart metering (e.g. smart grid), etc.They are in every domain even in critical ones like military, health care and buildings security.

In this chapter, we have introduced the IoT network and presented its enabling technologies that motivate to the emergence of IoT. Moreover, we reviewed different applications provided by the IoT paradigm and discussed the major elements and protocols integrated in the three-layered IoT architecture.

In next chapter, we focus on security vulnerabilities and requirements for IoT. We present different security attacks that threaten the IoT environments.and later provide a valuable taxonomy to highlight the security threats of IoT.

Chapter 2

Presentation of security issues in IoT: A focus on Botnets

2.1 Introduction

The increasing number of Internet of Things (IoT) devices, combined with their limited capabilities, have led to the proliferation of malware targeting IoT devices [38]. The majority of IoT malware simply takes advantage of weak built-in defenses and factory default credentials to compromise devices and to turn them into a bot, i.e.,remotely controlled device connected to a centralized control entity [38]. When an IoT device joins a botnet, that device is used for different purposes, including launching Distributed Denial-of-Service (DDoS) attacks. This chapter presents the different IoT security attacks, their definitions and purposes.

2.2 IoT security attacks

The enormous growth in the IoT has widely attracted the cyber-attackers that take advantage of the vulnerabilities and scarce resources of IoT devices so that to launch various security breaches. To include the security requirements carefully into the IoT systems, it is firstly necessary to analyze the IoT vulnerabilities and attacks. The IoT is prone to various types of attacks since it combines different existing technologies such as WSN and RFID. we going to explained different security attacks that threaten the IoT networks in the following subsections.

- **Man in the middle attacks:** Man in the middle attackers pretend to be a part of the com-

munication systems where the attackers are directly connected to another user device Therefore, it can easily interrupt communications by introducing fake and misleading data in order to manipulate original information [13].

Figure 2.1: Man in the middle attacks.

- **Malicious Scripts:**Usually the IoT network is connected to the Internet. The user that controls the gateway can be fooled into running executable active-x scripts which could result in a complete system shut down or data theft [16].

- **Data tampering:**In data tampering, the attackers manipulate the user's information intentionally to disrupt their privacy using unwanted activities.The IoT devices that carry important user's information such as location, fitness, billing price of smart equipment are in great danger to encounter these data tampering attacks [14].

- **Node Tampering:**The attacker can cause damage to a sensor node, by physically replacing the entire node or part of its hardware or even electronically interrogating the nodes to gain access and alter sensitive information, such as shared cryptographic keys (if any) or routing tables, or impact the operation of higher communication layers [15].

- **Sybil Attack:** In this attack, malicious node that takes the identities of multiple nodes and acts as them. For e.g. in Wireless Sensor Network, voting system single node can vote many times [17].

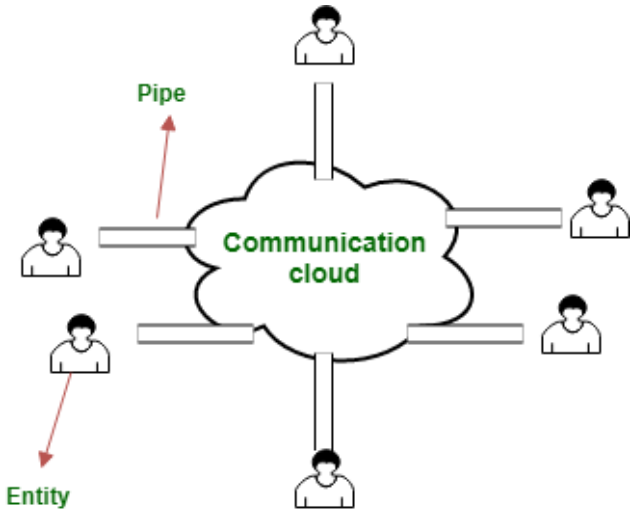

Figure 2.2: Sybil Attack.

- **RFID Spoofing:** An adversary spoofs RFID signals. Then it captures the information which is transmitted from a RFID tag. Spoofing attacks give wrong information which seems to be correct and that the system accepts [18].

- **RFID Cloning:** In this attack, adversary copying data from pre-existing RFID tag to another RFID tag.It does not copy original ID of RFID tag.The attacker can insert wrong data or control the data passing via the cloned node [19].

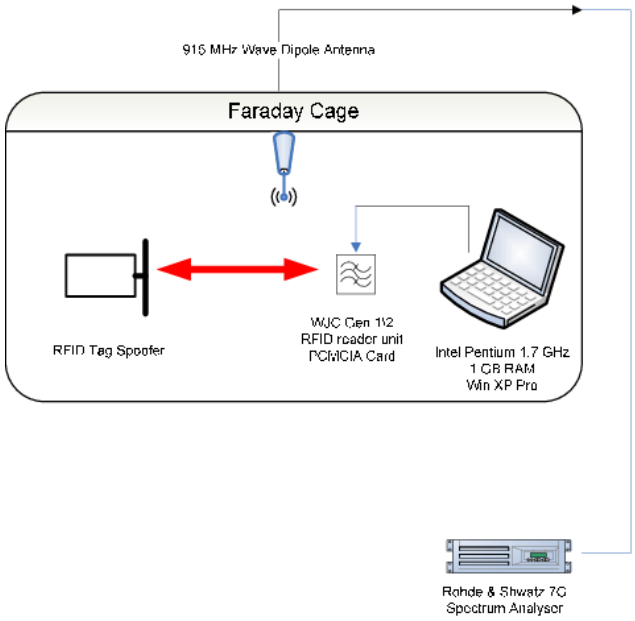

Figure 2.3: RFID Cloning .

- **RFID Unauthorized Access:** If the correct authentication is not provided in the RFID systems,then theadversary can observe, alter or remove information on nodes [19].

- **Jamming attack:** Attackers send fake signals to interrupt the ongoing radio transmissions of IoT devices and further deplete the bandwidth, energy, central processing units (CPUs), and memory resources of IoT devices or sensors during their failed communication attempts [20].

- **Replay attack:** An attacker my capture a signed packet, and even if it cannot decrypt it,it may gain the trust of the destined entity by re-sending the packet at a later time. Replay attacks can be circumvented by using message sequence numbers and message authentication code (MAC) [21].

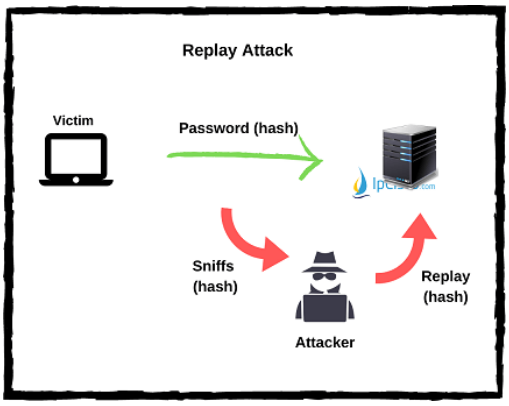

Figure 2.4: Replay attack .

- **Denial of Service (DoS)** DDoS attack is the most common cyberattack in which attacker's computers send large number of malicious traffic to the target server at the same time to overwhelms the target network [39]. DDoS attacks intend to significantly interrupt normal functioning of target server by flooding the target device with massive traffic such as fraudulent request to over saturate its capacity causing a disruption or denial of service to the legitimate traffic [40]. DDoS attacks affect the server's system resources such as CPU, memory and can also cause the network bandwidth to saturate with large number of traffic, as a result, legitimate computers are going to be denied service because the server is preoccupied in dealing with DDoS attack. Hackers use botnet to launch DDoS attack. IoT devices get involved in DDoS attack after they gets infected by the malicious software that the attacker distributes over the internet. Infected IoT devices acts as a bot and they are used by the attacker to launch DDoS attacker [41].

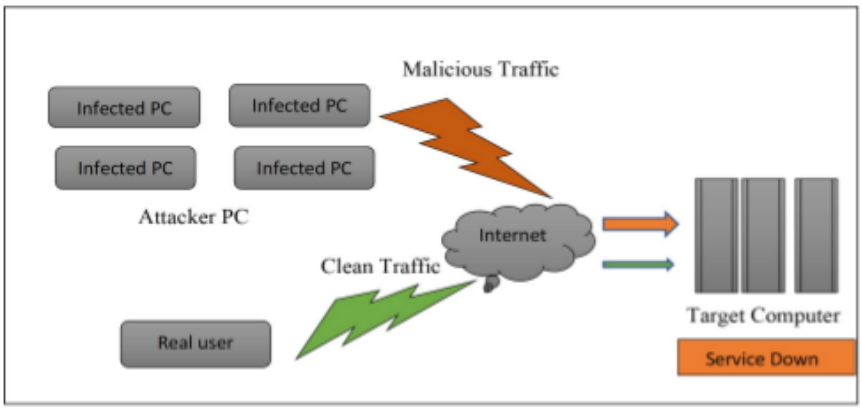

Figure 2.5: Ddos Attack [41].

- **Botnet formation:** In this attack, machine in the network gets converted into a bot (i.e., software robot).This bot finds more vulnerable nodes and converts them into bot and forms botnet.This is a looping process. Eventually, all the machines in the network become a bot. Using botnet you can gain control on the network, data can be hacked and, DDOS attack can be performed The system will be not available/loss of confidentiality or can result in loss of integrity [25].

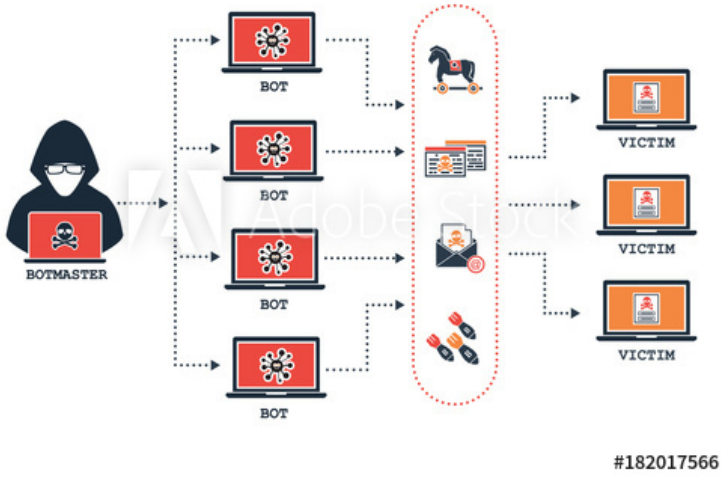

Figure 2.6: Botnet attack.

- **Blackhole attack:**Insert a new node or compromise the node in the existing network so that all the neighbors of this node will change the routing table and transmit the data

from this node only.The node once received the packet will never forward it Loss in data the impact of this attack Access to private data and join the network [25].

- **Wormhole attack:** Create a false one-hop transmission (tunnel) to deliver more data through this tunnel this attack Breach the data confidentiality and launch additional attacks [26].

- **routing attack:** These are immediate attacks that the enemy by spoofing, replaying or changing routing data can con-volute the system and make routing loops, permitting or dropping movement,sending the false error messages, shortening or amplifying source courses or notwithstanding parceling the network [27].

- **Spear Phishing Attack:** It is an email spoofing attack in which victim,a high ranking person,is lured into opening the email through which the adversary gains access to the credentials of that victim and then by retrieves more sensitive information [29].

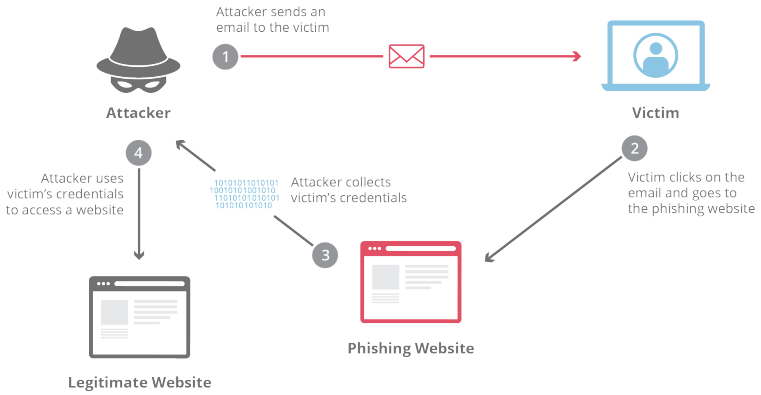

Figure 2.7: Spear Phishing Attack.

- **Malicious Code Injection:** Malicious Code Injection: An attacker can leverage the attack on the system from end-user with some hacking techniques that allows the attacker to inject any kind of malicious code into the system to steal some kind of data from the user [29].

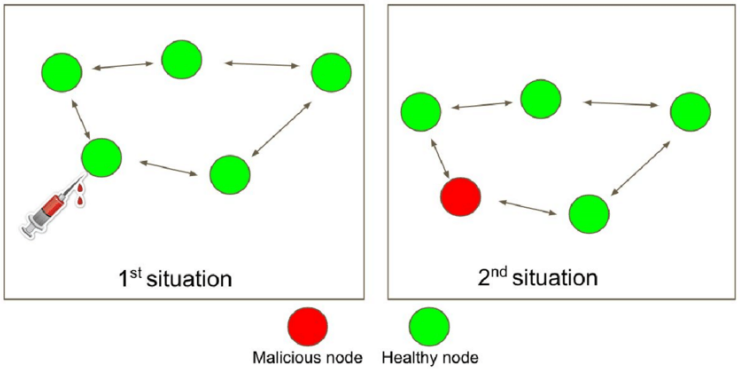

Figure 2.8: Malicious Code Injection.

- **data transit attack:** Various attacks like sniffing, Man in the middle attacks on the integrity & confidentiality during data transit [30].
- **Sniffing Attack:** A sniffer application is introducing by the attacker into the system which collect all the information from the network about the devices and communication [28].

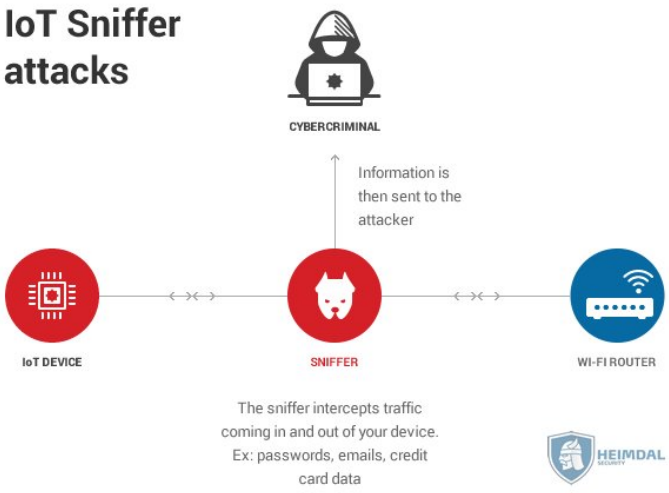

Figure 2.9: Sniffing Attack.

2.3 IoT security threats

In this section, we focus on the security vulnerabilities of IoT at the three layers. Levels examine the security issues of IoT at the three layers. Perception layer threats address the security attacks

within major elements of IoT such as WSNs and RFID. Network layer threats analyze vulnerabilities of the aforementioned communication protocols. Application layer threats include attacks related to IoT software and end-user devices.

2.3.1 Preception Layer Attacks

Hardware attacks are the most common attacks on perception layer. Perception layer generally includes WSN, RFID, zigbee and other kind of sensors.The attacker needs to be in the network or physically close to the nodes of the IoT system. Some of the common attacks on the perception layer are blackhole, wormhole, sybil, denial of service (DoS) [27].

2.3.2 Network Layer Attacks

This layer involved in the transmission of data across the network, which it receives from the perception layer. Thread for this layer is increased because this layer is receiving data from different heterogeneous devices. Which increase the security threads. Operation of IoT network layer is same as TCP/IP which will come with the same problems that we are facing in TCP/IP network layer model Different attacks on the network layer are possible which include, DoS Attack, Storage attack, man in middle attack routing attack and data transit attack. These attacks can create a challenging situation for IoT environment [28].

2.3.3 Application layer Attacks

Application layer mainly includes the devices for effective decision making. Each of these has some vulnerability which leads to be an issue of the security of IoT. The attacker is likely to destroy privacy in the application layer by a known vulnerability (e.g., buffer overflow, cross site scripting, and SQL injection), error configuration (e.g., simple password), or improperly obtained higher permission access. Some of the common attacks on the Application layer are Denial of Service Attack,Spear Phishing Attack,Sniffing Attack,Malicious Code Injection [29].

2.4 Overview of IoT Botnet

IoT bot represents a software robot which scans for vulnerable devices and once found converts it into bot just like a traditional bot.It is an automated process of extending malware.IoT botnet is a network of bot, i.e., infected machines.IOT botnet is controlled by botmaster who execute coordinated activities with the help of these bots.The coordinated activity could be DDOS attack, spamming, phishing campaign, click fraud, and spyware.IoT devices turn into bot due to lack of primitive security, virus infection, or opening a malicious email attachment [32].

2.4.1 Types of Botnet based on the architecture

Centralized Botnets

The old approach used by Botnet for their Command and control (C&C) architecture was the centralized mechanism (hierarchical). In this approach, the Bot-master (attacker) distributes the command over the Botnet via various Bot-Controllers in order to hide attacker's real identity. The uses of multiple Bot-Controllers prevent security professionals from shutting down C&C channel shown in Fig. 2.10. In Figure, the Bot-Controller retrieves the command from the Bot-master and then Bot-controller distributes these commands further to all the Bots in the Botnet [33].

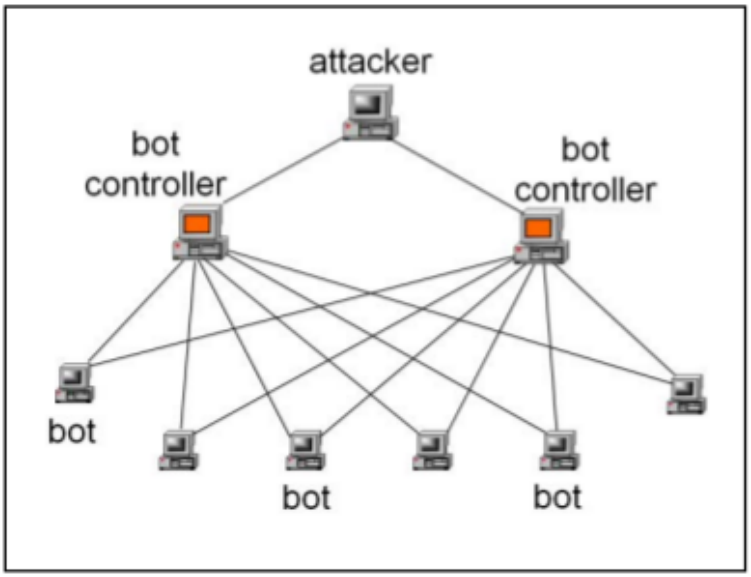

Figure 2.10: Illustration of a centralized Botnet[33].

Peer to Peer (P2P) Botnet Architecture

To remove the draw backs of centralized architecture, the hacker's focused on the peer to peer model characteristics for Botnet, which is actually hard to manage for the Supervisor-Bot but also hard to detect, monitored and blocked by security managers. Supervisor-bot transfer command to an infected zombie peer who transfers it to other peers, acting both as Supervisor-bot and zombie army soldier. Similarly it can transfer commands from any zombie, which lead to a slow but effective undetectable communication between zombie army [34]. Examples of bots using P2P are Phatbot and Peacomm.

P2P uses several controllers for hiding and not to be seized and closed along with encrypted keys for misuse of the technology other than the supervisor-bot. It works in various phases and periods without using bandwidth significantly at same time. Data mining technique gave some promising results in detecting P2P attacks [35].

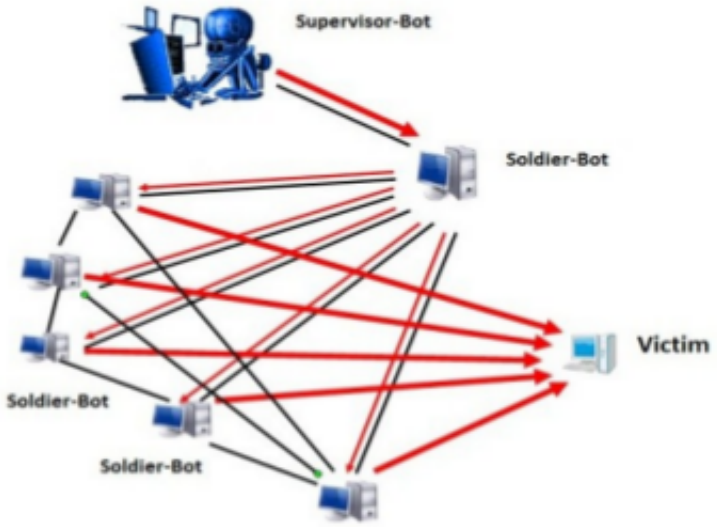

Figure 2.11: P2P Architecture [35].

2.4.2 Mirai

On October 2016, the largest DDoS attack was launched on DNS service provider (Dyn) by using an IoT botnet. The botnet was made possible by a malware named Mirai which led to shutdown of huge portions of the Internet including Twitter, Netflix etc [36].

Mirai Botnet Principle and Working

The main goal of the Mirai botnet is to perform a DOS attack. Fig. 2.12 depicts the working of Mirai. It can compromise IoT devices very efficiently. The command and control server runs two-socket listeners: one for Telnet connections and one for a programmatic API. The Telnet socket will listen on port 23 and route any valid connections to it to the appropriate bot or admin handler.(CnC) a portion of Mirai is written in Go, an efficient and compiled language made by Google. The API socket will listen on port 101 and route any valid attack commands sent to it to the connected bots. Each connected bot will scan the Internet for new vulnerable devices. When one is discovered, the credentials, IP address, and port used to gain access to it are sent to a loader server. This loader will output the information to the console to allow the data to be optionally stored into a file as well, and then will use the information to download and execute

the malware on the device [32].

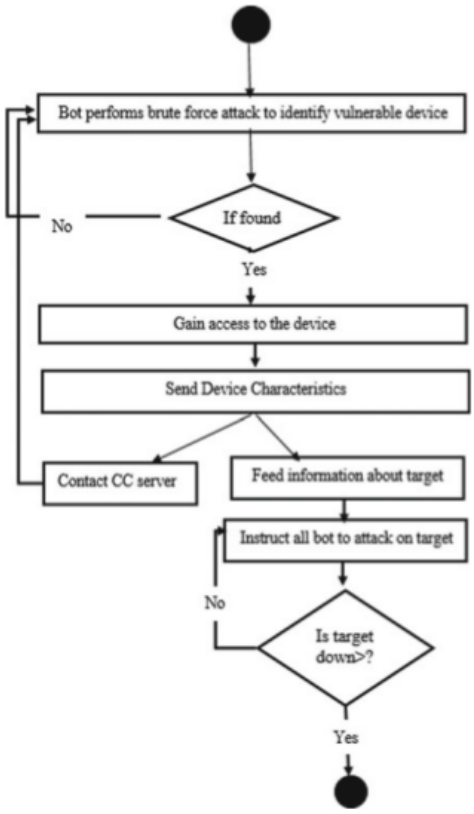

Figure 2.12: Mirai workflow [32].

2.5 Security concepts

The term security subsumes a wide range of different concepts [31] In the first place, it refers to the basic provision of security services including:

- **Privacy:** Privacy such as identity or commercial interest of an individual user should be protected by the secure IOT system [37].
- **Authorization:** The process of determining whether someone or something is, in fact, who or what it is declared to be. We distinguish two kind of attacks related to authentication namely, impersonation attack where an attacker pretends to be another entity, and Sybil attack where the attacker uses different identities at the same time [31]

- **Integrity:** Integrity property ensures that only authorized users can modify the information of the IoT devices while using a wireless network for communication [13].
- **Authentication:** The process of determining whether someone or something is, in fact, who or what it is declared to be. We distinguish two kind of attacks related to authentication namely, impersonation attack where an attacker pretends to be another entity, and Sybil attack where the attacker uses different identities at the same time [31].
- **Confidentiality:** Sensitive Information shall not be leak to any unauthorized reader by using an RFID electronic tag [37].
- **Non-repudiation:** Set of means and techniques to prove the involvement of an entity in a data exchange. Attacks on non-repudiation consist of a denial of participation in all or part of communications [31].
- **Availability:** An authorized user can able to use various services provided by IOT and can prevent DOS attack for the availability of the services. DOS attack is major cause for threat to the availability [37].

2.6 Conclusion

With the gradual popularization of the Internet of Things in everyday life, the security of IoT faces more and more challenges. In this chapter, we analyzed the security vulnerabilities and threats of IoT networks Where Botnets have played an important role as a major security threats on the Internet of Things. That is why one has to work in advance then the hackers not only on its after effect but before the attacks are done. In next chapter, we present solution Which is Machine learning algorithms that have been proposed to achieve security requirements in IoT.We also review related works that address the security of IoT systems using machine learning.

Chapter 3

Study of the proposed solutions to overcome Botnets

3.1 Introduction

Machine learning is used to teach machines how to handle the data more efficiently.Sometimes after viewing the data,we cannot interpret the pattern or extract information from the data.In that case,we apply machine learning [42].With the abundance of datasets available,the demand for machine learning is in rise [43].

The value of machine learning is that it allows you to continually learn from data and predict the future.This powerful set of algorithms and models are being used across industries to improve processes and gain insights into patterns and anomalies within data [44].Machine learning (ML) algorithms have been widely used in many applications domains,including advertising,recommendation systems,computer vision,natural language processing,and user behavior analytics [45].In this chapter,we will be learning about the concepts and techniques of machine learning.

3.2 Definition of Machine Learning

Machine learning is defined by (Arthur Samuel, 1959) as the field of study that gives computers the ability to learn without being explicitly programmed. from this definition we can see that we can perfectly apply it to the text classification problems. The following Fig. 3.1 shows an

overview of classes of some machine learning algorithms in the field [64].

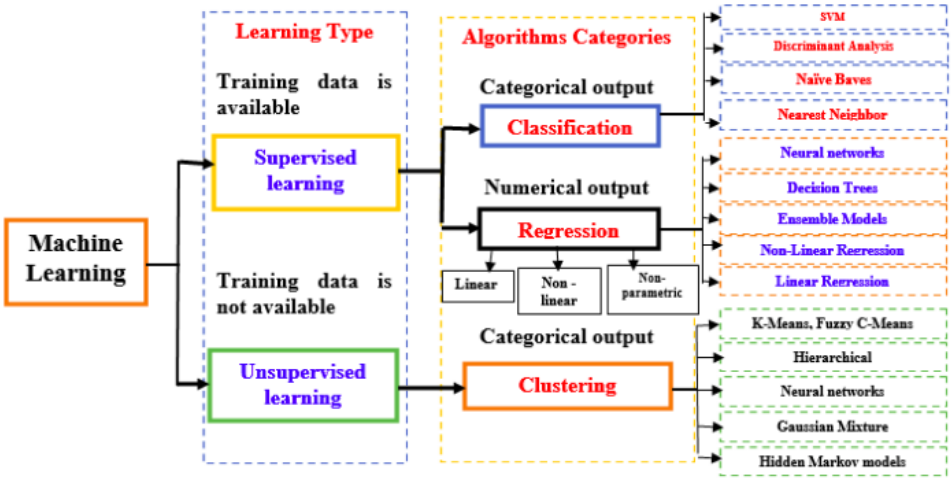

Figure 3.1: Machine Learning Algorithms [64].

3.3 Development of Machine Learning

The following table 3.1 depicts the illustrious, expansive and practical development of machine learning.

Table 3.1: Development of ML.

1950	Alan Turning created "Turning Test" to check a machine's intelligence. In order to pass the Turning Test,the machine should be able to convince humans that there they are actually talking to a human and not a machine [46].
1952	Samuel created a highly capable learning algorithm than can play the game of Checkers with itself and get self-trained [46].
1956	Martin Minsky and John McCarty with Claude Shannon and Nathan Rochester organized a conference in Dartmouth in 1956 where actually Artificial Intelligence was born [46].
1958	Frank Rosenblatt created Perceptron, which laid the foundation stone for the development of Artificial Neural Network (ANN) [46].
1967	The Nearest Neighbor Algorithm was proposed which could be used for "Pattern Recognition" [46].
1979	Stanford University students developed "Stanford Cart",a sophisticated robot that could navigate around a room and avoid obstacles in its path [46].
1981	Explanation Based Learning (EBL) was proposed by Gerald Dejong, whereby,a computer can analyze the training data and create rules for discarding useless data [47].
1985	NetTalk was invented by Terry Sejnowski,[48]which learnt to pronounce English words in the same manner that children learn.
1990s	The focus of Machine Learning shifted from Knowledge-driven to Data Driven. Machine Learning was implemented to analyze large chunks of data and derive conclusions from i t[49].
1997	IBM invented the Deep Blue computer which was able to beat World Chess Champion Gary Kasparov [46].
2006	The term "Deep Learning" was coined by Geoffery Hinton which referred to a new architecture of neural networks that used multiple layers of neurons for learning [46].

2011	IBM's Watson,built to answer questions posed in a natural language,defeats a Human Competitor at Jeopardy Game [46].
2012	Jeff Dean from Google,developed GoogleBrain, which isa Deep Neural Network to detect patterns in Videos and Images [46].
2014	Facebook invented the "DeepFace" algorithm based on Deep Neural Networks capable ofrecognizing human faces in photos [46].
2015	Amazon proposed its own Machine Learning Platform. Microsoft created "Distributed Machine Learning Toolkit" for efficient distribution of machine learning problems to multiple computers to work parallel to find a solution.Elon Musk and Sam Altman,created a non-profit organization-OoeaAI,with the objective of using Artificial Intelligence to serve human beings [46].
2016	Google proposed DeepMind which is regarded as the most complex Board Game.Google AlphaGo program becomes the first Computer Go program to beat a professional human player.It is based on the combination of machine learning and tree searching techniques [50].
2017	Google proposed Google Lens,Google Clicks,Google Home Mini and Google Nexus based phones which use Machine Learning and Deep Learning Algorithms Nvidia proposed NVIDIA GPUs-The Engine of Deep Learning.Apple proposed Home Pod which is a Machine Learning Interactive device [46].

3.4 The Generic Model of Machine Learning

The generic model of machine learning consists of six components independent of the algorithm adopted.The following Fig. 3.2 depicts these primary components[46].

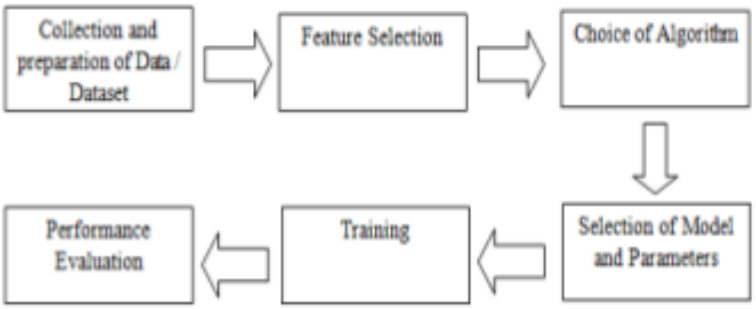

Figure 3.2: Components of a Generic Machine Learning model [46].

Each component of the model has a specific task to accomplish as described next.

- **Collection and Preparation of Data:** The primary task of in the machine learning process is to collect and prepare data in a format that can be given as input to the algorithm.A large amount may be available for any problem. Web data is usually unstructured and contains a lot of noise, i.e., irrelevant data as well as redundant data. Hence the data needs to be cleaned and pre-processed to a structured format [46].

- **Feature Selection:** The data obtained from the above step may contain numerous features, not all of which would be relevant to the learning process. These features need to be removed and a subset of the most important features needs to be obtained [46].

- **Choice of Algorithm:** Not all machine learning algorithms are meant for all problems. Certain algorithms are more suited to a particular class problem as explained in the previous section. Selecting the best machine learning algorithm for the problem at hand is imperative in getting the best possible results [46].

- **Selection of Models and Parameters:** One important step to select a model to fit the collected data is to categorize the problem [66].

- **Training:** in this step we train our model to improve its performance for better outcome of the problem.We use datasets to train the model using various machine learning algorithms.Training a model is required so that it can understand the various patterns, rules,and features [65].

- **Performance Evaluation:** Before real-time implementation of the system, the model must be tested against unseen data to evaluate how much has been learnt using various performance parameters like accuracy, precision and recall [46].

3.5 Machine Learning Paradigms

Depending on how an algorithm is being trained and on the basis of availability of the output while training,machine learning paradigms can be classified into ten categories.we going to explained some of this paradigms in the following subsections [46].

3.5.1 supervised learning

Supervised learning is a machine learning technique for learning a function from training data.The training data consist of pairs of input x objects (typically vectors),and desired outputs y.The output of the function f can be a continuous value (called regression),or can predict a class label of the input object (called classification) [51].

the goal is to obtain an optimal predictive model function f^* to minimize the cost function $\mathscr{L}(f(x), y)$that models the error between the estimated output and ground-truth labels.The predictive model function f varies based on its model structure.with limited model architectures determined by different hyper-parameter configurations,the domain of the ML model function f is restricted to a set of functions F.Thus,the optimal predictive model f^* can be obtained by [52]:

$$f^* = \arg\min_{f \in F} \frac{1}{n} \sum_{i=1}^{n} \mathscr{L}(f(x_i), y_i)$$

Many different loss functions exist in supervised learning algorithms,including the square of Euclidean distance,cross-entropy,information gain,etc [45].

3.5.2 Unsupervised Learning

Unsupervised learning is a type of machine learning where manual labels of inputs are not used.Unsupervised learning means we are only given the input Xs and some (ultimate) feedback function on our performance.We simply have a training set of vectors without function values

of them.The problem in this case,typically is to partition the training set into subsets,$\equiv_1 ... \equiv_R$,in some appropriate way [51].

3.5.3 Reinforcement Learning

Reinforcement learning is a type of learning which makes decisions based on which actions to take such that the outcome is more positive.The learner has no knowledge which actions to take until it's been given a situation.The action which is taken by the learner may affect situations and their actions in the future.Reinforcement learning solely depends on two criteria: trial and error search and delayed outcome [53].The general model [54]for reinforcement learning is depicted in the Fig. 3.3.

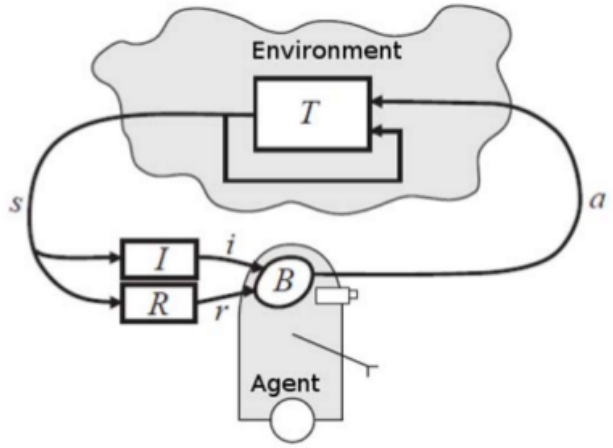

Figure 3.3: The Reinforcement Learning Model [54].

In Fig. 3.3,the agent receives an input i,current state s,state transition r and input function I from the environment.Based on these inputs,the agent generates a behavior B and takes an action a which generates an outcome [43].

One of the most common applications of reinforcement learning is in robotics or game playing,also the algorithm that is being used for self-driving cars [44].

3.5.4 Artificial Neural Network

The neural network (or artificial neural network or ANN) is derived from the biological concept of neurons.A neuron is a cell like structure in a brain.A neuron is a cell like structure in a brain [43].A neuron consists of four parts,namely dendrites (receptor),some (processor of electric signal),nucleus (core of the neuron) and axon (the transmitting end of the neuron) [46].The dendrites receive electrical signals.Some processes the electrical signal.The output of the process is carried by the axon to the dendrite terminals where the output is sent to next neuron.The nucleus is the heart of the neuron.The inter-connection of neuron is called neural network where electrical impulses travel around the brain.An artificial neural network behaves the same way [43].Analogical to a biological neural network, an ANN works on three layers: input layer, hidden layer and output layer [46].There are basically three types of artificial neural network:supervised,unsupervised and reinforcement [55].

3.6 Machine Learning Algorithms

3.6.1 Supervised Learning

Decision Tree

Decision trees are those type of trees which groups attributes by sorting them based on their values.Decision tree is used mainly for classification purpose.Each tree consists of nodes and branches.Each nodes represents attributes in a group that is to be classified and each branch represents a value that the node can take [56].An example of decision tree is given in Fig. 4.4 [43].

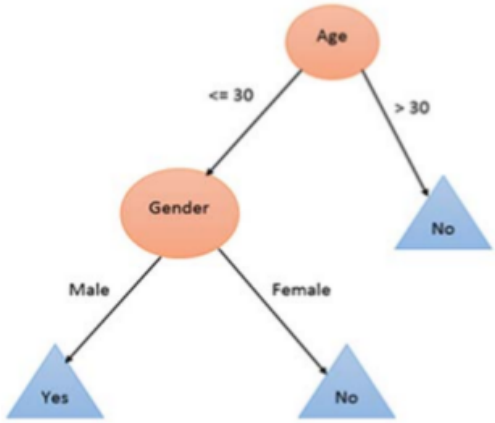

Figure 3.4: Decision Tree [43].

Naive Bayes

The Bayesian classification is another method of the supervised learning methods as well as the statistical method for classification.Assumes an underlying probabilistic model and it allows capturing uncertainly about the model in a principled way by determining probabilities of the outcomes.The basic purpose of the Bayesian classification is that it can solve predictive problems.This classification provides practical learning algorithms and can combine observed data.Bayesian classification provides useful perspective for understanding and evaluating learning algorithms.It calculates explicit probabilities for hypothesis and it robust the noise in input data. Let's consider a general probability distribution of two values $P(x_1, x_2)$.Using Bayes rule,without loss of generality we get this equation:

$$P(x_1, x_2) = P(x_1|x_2)P(x_2)$$

Similar,if there is another class variable c,we get the next equation[57]:

$$P(x_1, x_2|c) = P(x_1|x_2, c)P(x_2|c)$$

If the situation is generalized with two variables to a conditional independence assumption for a set of variables$x_1 ... x_N$ conditional on another variable c,we get the following [91]:

$$P(x|c) = \prod_{i=i}^{N} P(x_i|c)$$

Logistic Regression

This technique in machine learning which is called logistic regression was borrowed from the field of statics, as it is very suitable for technique for binary classification or bi-class problems(problems that have 2 classes of values for classification). The name for logistic regression was brought from a function in mathematics and statics called the logistic function also called the sigmoid function,this logistic function was invented by statisticians for the purpose of describing the properties of growth of populations in ecology,the fast increase and reaching the maximum carying capacity of the envitonment, Logistic function is an shaped as an S-shape curve with is capable of mapping any real-valued number into a value between 0 and 1, and not touching those limits [63].

$$\frac{1}{1+e^{-value}}$$

Where e is representing the natural logarithm, or Eulers number (EXP()) and value is the transformable numerical value, we see bellow a plot 5 that transforms numbers between -8 and 8 into a range of 0 to 1 using the sigmoid or logistic function.

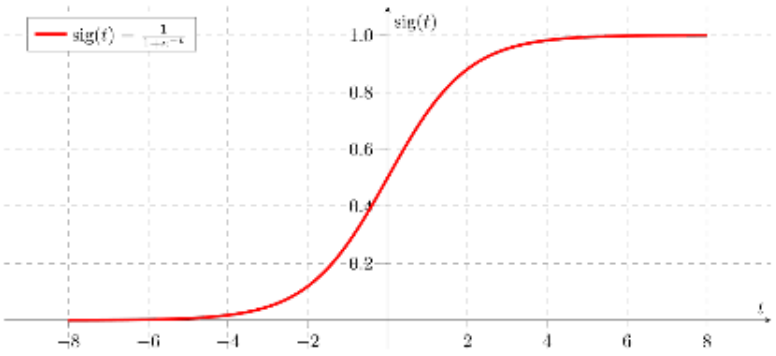

Figure 3.5: Logistic function transformation example [64].

Despite the name Logistic regression it is considered as a classification algorithm, we use

it when we have dependent variable the are binary, this means they can be either one of tow categories or values for example yes or no , true or false, one or zero. It combines weighted input features in a linear fashion which applied to the sigmoid function.This sigmoid function is main player of logistic regression and it can map value into the range of 0 to 1. Representing Hypothesis When representing or using the linear regression we use the following formula [64]:

$$h(x) = \beta_0 + \beta_1 x$$

In other hand when using logistic regression the previous formula will be modified a bit as the following [64]:

$$\sigma(Z) = \sigma(\beta_0 + \beta_1 X)$$

Finally we get the end hypotheses for logistic regression as follows[64]:

$$h(x) = \frac{1}{1 + e(^{-}(\beta_0 + \beta_1 X))}$$

K -Nearest Neighbor (kNN)

K-nearest neighbor (kNN) is one of the modest and conventional non-parametric techniques for classifying samples.It calculates the approximate distances between various points on the input vectors,and then assigns the unlabeled point to the class of its K-nearest neighbors [59].

Assuming the training set $T = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}, x_i$ is the feature vector of an instance,and $y_i \in \{c_1, c_2, ..., c_m\}$ is the class of the instance,for a test instance $i = (1, 2, ..., n)$,its class y can be denoted by [45]:

$$\arg\max_{cj} \sum_{x_i \in N_K(x)} I(y_i = c_j), i = 1, 2, ..., n; j = 1, 2, ..., m.$$

.

where I(x) is an indicator function,I=1 when $y_i = c_j$,otherwise I=0;$N_K(x)$ is the field involving the k-nearest neighbors of x [45].

In the process of creating kNN classifier,(k) is an important parameter and various (k) values can cause various performances [59],If k is too small,the model will be under-fitting;if k is too

large,the model will be over-fitting and require high computational time [45].

3.6.2 Unsupervised Learning

K-Means

K-means algorithm is a traditional clustering algorithm.It divides the data into k clusters, and guarantee that the data within the same cluster are similar, while the data in a various clusters have low similarities [59].

K-Means algorithm is be employed when labeled data is not available.General method of converting rough rules of thumb into highly accurate prediction rule [60].

3.7 Applications of Machine Learning

Research shows that machine learning technology has been widely used in marketing,finance,and network analysis.

In the field of marketing, machine learning technology is more widely used in the area of tasks classification-and-related; in the field of finance,machine learning technology is more widely used in tasks of forecasts;in the field of network analysis,machine learning technology is used in the relating tasks;in the field of telecommunications,machine learning technology is widely used in the tasks of classification,prediction and spy.In addition,machine learning is also applied in the field of data mining combination with other applications,the typical methods are based on the neural network initialization,the application of evolutionary computation in machine learning research,the study of level classification of machine learning, and machine learning based on rough set and so on [61].

3.8 Conclusion

Today each and every person is using machine learning knowingly or unknowingly.From getting a recommended product in online shopping to updating photos in social networking sites. In this chapter,we introduced the introduction to most of the popular machine learning algo-

rithms. In next chapter we will apply some of machine learning algorithms Then,we compare the use techniques in terms of detection efficiency and decide about the most suitable technique.

Chapter 4

Scope on the realized solution

4.1 Introduction

In October 2016,the Mirai botnet took down domain name system provider Dyn,waking much of the world up to the fact that Internet of Things devices could be weaponized in a massive distributed denial of service (DDoS) attack.

Although DDoS attacks have been around since the early days of the modern internet, IT communities around the globe came to realize that IoT devices could be leveraged in botnet attacks to go after all kinds of targets.

The security measures that have been used become vulnerable with the vulnerability of IoT devices.that's why we must adopted Modern security tactics on IoT network to keep IoT entities, organizations, and individuals safe.Machine learning technology based network traffic classification has become a hot topic and has achieved encouraging results in intrusion detection.

In this paper,a methodology for botnet detection is presented that comprises data preprocessing,and SMOTE technique to balance the data-set class.Also,feature engineering was done upon analyzing machine learning algorithms for classification.We presented the effect of imbalance data and number of feature selected and its impact on machine learning.

4.2 Related Works

The research on solving DDos botnet attack in Iot is widely discussed in the literature.

Narasimha et al.[84] used anomaly detection along with the machine learning algorithms

for bifurcating the normal and attacked traffics. For the experiment, real-time datasets were used. Famous naive Bayes ML algorithm was used for classification purpose.The results were compared with existing algorithms like J48 and random forest (RF).

Liang X in et al. 2018, in [85] studied the security models for the Internet of Things (IoT) using machine learning techniques.They studied, and reviews mostly review technique for IoT security based on ML techniques.

Tao [86] uses entropy change to detect attacks in traffic. Once the detection system detects an attack, it will block or limit abnormal traffic and isolate the attacker's location. Information distance is employed to differentiate DDoS attacks from flash crowds.If the information distance in the suspicious flow is less than a given threshold, it will be described as a DDoS attack, otherwise it is a network transient congestion.

Olivier Brun et al.[87] worked in the area of Internet of Things (IoT) to detect the DDoS attack. The author implemented one of the famous deep learning techniques, i.e., random neural network (RNN) technique for detection of the network.This deep- learning-based technique efficiently generates more promising results compared to existing methods.

Fok et al.[88] suggested a Botnet traffic detection technique based on machine learning. The research relies on multilayer perceptrons and decision trees on network traffic analysis for detecting traffic automatically. The researchers have used recall and false positive rate(FPR) to justify the results. The results specify that use of Decision Trees instead of the existing threshold-based decision maker may be used effectively to increase the recall of the framework, while the FPR is reduced to almost zero.

4.3 Research approach

In this section we describe the procedures followed during the botnet detection model creation.

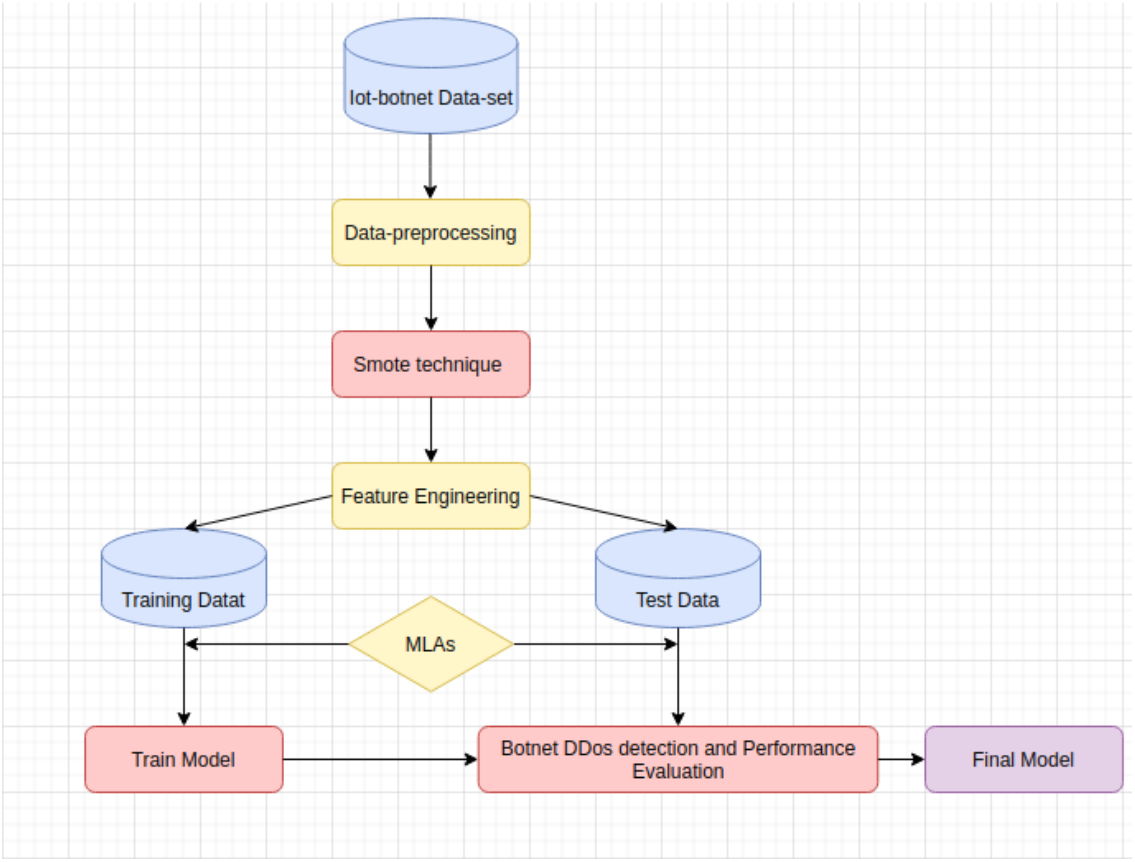

Figure 4.1: Proposed Model.

4.3.1 BoT-IoT dataset

In this project, we used Bot-IoT dataset [67] for the selection of effective machine learning (ML) algorithm to detect DDos botnet threat in the internet of thing (IoT).

The BoT-IoT dataset was created by designing a realistic network environment in the Cyber Range Lab of UNSW Canberra. The network environment incorporated a combination of normal and botnet traffic.The dataset's source files are provided in different formats,including the original pcap files,the generated argus files and csv files.The files were separated, based on attack category and subcategory,to better assist in labeling process [68].

The captured pcap files are 69.3 GB in size,with more than 72.000.000 records. The extracted flow traffic,in csv format is 16.7 GB in size.The dataset includes DDoS, DoS,OS and Service Scan, Keylogging and Data exfiltration attacks, with the DDoS and DoS attacks further organized,based on the protocol used [68].

Different supervised MLAs were used(Decision Tree,Naive Bayes,Logistic Regression,K-Nearest Neighbor) on different combination of Botnet dataset and benchmarked the result to select a best algorithm for our model.

The dataset we used contain 92.3 MB,there's major contrast in this dataset it contains more than 99% of botnet traffic while less than 1%normal traffic.

We created another dataset after processing real-time BoT-IoT dataset through SMOTE technique which provided class balance dataset with equal number of botnet traffic and normal traffic,to compare between the balanced and imbalanced dataset.then we selected different number of features to compare the performance of our model with those features.

4.3.2 Data preprocessing

Once the data is acquired,the next stage is to preprocess the data in order to bring it in a refined form. the data preprocessing can often have a significant impact on generalization performance of a supervised ML algorithm. The elimination of noise instances is one of the most difficult problems in inductive ML [69].

- **Data Cleaning:** Or data cleansing,includes operations that correct bad data,filter some incorrect data out of the data set and reduce the unnecessary detail of data.We undergo through data cleaning process to identify missing values and delete those rows.We drop the rows containing null value in BoT-IoT dataset using dropna() function of pandas [70].

- **Normalization:** Normalization is a "scaling down" transformation of the features.It is important to maintain a uniform distribution of each attribute values before starting the learning process.As some feature in our BoT-IoT dataset have data of variant range which make complex for the model to learn and will cause model learning problem taking it more time to decide to converge to result.For this purpose we use the MinMax method [71]. In MinMax the values of features are scaled to the interval [0,1] as follows:

$$Y_norm = (Y - Y_{in})/(Y_max - Y min)$$

We get Ymin and Ymax by using .min() and .max() functions of pandas.

- **Transformation:** In computing, Data transformation is the process of converting data from one format or structure into another format or structure.In BoT-IoT dataset we use there are many categorical features containing non-numeric data which needed to be converted into numeric format for the MLAs to process it as the MLAs we were using were in algebraic format. We classify two types of fields in Bot-Iot dataset: text fields,and numerical fields We transform the representation for the content of text field.

4.3.3 Feature Engineering

Feature engineering technique can be an important part of the machine learning process as it has the ability to greatly improve the performance of our models by dimensionality reduction which thereby minimized the problem of over-fitting. Also,it beneficial in selecting appropriate features that contain most important information about target variable Appropriate [72].

The whole data-set is trained using Extra Trees Classifier. After training,every feature gets a feature importance score assigned by the Extra Trees Classifier. This is assigned depending upon the number of times that feature got selected as the best split for a decision tree, predicting the majority vote [82]. An Algorithmic Visualization of Extra Trees Classifier is given in Figure. 4.2.

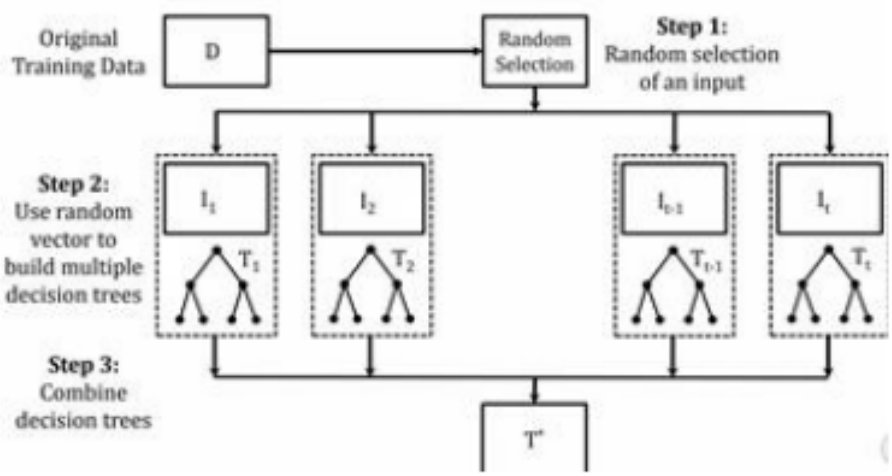

Figure 4.2: Visual Representation of Extra Trees Classifier [82].

4.3.4 Synthetic Minority over-sampling technique(SMOTE)

SMOTE is an over-sampling technique.This technique increases a number of new minority class instances by interpolation method.The minority class instances that lie together are identified before they are employed to form new minority class instances [73]. This technique is able to generate synthetic instances rather than replicate minority class instances; therefore, it can avoid the over-fitting problem. The algorithm is described in Fig. 4.3.

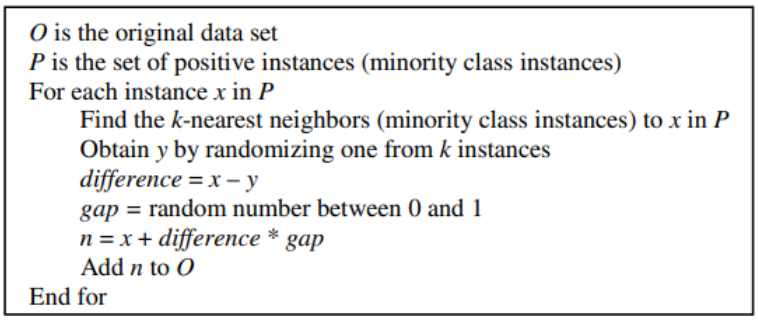

Figure 4.3: The Synthetic Minority Oversampling Technique (SMOTE) [73].

4.3.5 Experimental Scenario

we used google colab for model creation and testing ,Colaboratory,or “Colab” for short,is a product from Google Research.

Colab allows anybody to write and execute arbitrary python code through the browser,and is especially well suited to machine learning, data analysis and education.More technically,Colab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs [74]. Google Colab provides RAM of 12 GB and a disk space of 107 GB python 3.7. The model is trained KNN,Decision Tree,Logistic Regression and BernoulliNB algorithms with the training datasets of 92.3 MB and the accuracy is the benchmarked to select the best algorithm for our detection system.

the Bot-Iot dataset we used is imbalanced there’s more than 99% of botnet traffic while less than 1%normal traffic. To make our BoT-IoT dataset we used Synthetic Minority over-sampling technique also referred to as SMOTE.We implemented MLAs on unbalanced dataset and on balanced dataset.

To solve problem of overfitting we select different number of features and compare the performance of our model with those features.

To process the dataset and implement machine learning,i use numerous python libraries.I mainly used Sklearn, numpy, matplotlib and pandas.

- **NumPy:** NumPy is a Python library used for working with arrays.It also has functions for working in domain of linear algebra,fourier transform, and matrices [75].
- **matplotlib:** Matplotlib is a comprehensive library for creating static, animated,and interactive visualizations in Python [76].
- **Pandas:** Pandas library support data analysis. we use panda's library to import dataset in .CSV file format and for data manipulating [77].
- **Sklearn:** Sklearn: Sklearn library is mainly used to create confusion matrix,to construct machine learning models, for splitting dataset,to perform data preprocessing and for feature engineering procedure [77].

4.3.6 Machine Learning Model Evaluation and Cross validation

- **Precision:** Precision means the positive predictive value.It is a measure of the number of true positives the model claims compared to the number of positives it claims [78].
- **Recall:** The recall is known as the actual positive rate which means the number of positives in the model claims compared to the actual number of positives there are throughout the data [78].
- **Accuracy:** It is defined as the ability of the system to correctly classify the attack packet as an "attack packet" and normal packet as a "normal packet".It tells about the ratio of correct predictions with respect to all samples.
  Accuracy=((TPs+TNs)/(TPs+TNs+FPs+FNs) * 100).
- **F1-Score:** F-Score metric combine precision and recall giving a single score value to balance both the concerns of precision and Recall.F1-Score is analyzed when false positives and false negatives are important [77].

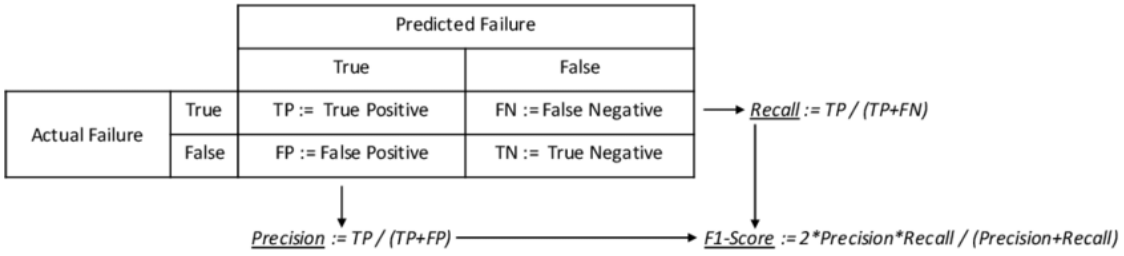

Figure 4.4: Confusion matrix, illustrating the calculation of precision, recall,and F1-score [83].

- **ROC AUC:** AUC - ROC curve is a performance measurement for the classification problems at various threshold settings.ROC is a probability curve and AUC represents the degree or measure of separability.It tells how much the model is capable of distinguishing between classes.Higher the AUC,the better the model is at predicting 0 classes as 0 and 1 classes as 1 [80]. the BoT-IoT real dataset is highly imbalance with 733598 Botnet traffic and 107 Normal traffic in 733705 total data.Classifier always predicting each traffic as botnet traffic will still have more than 90 percentage accuracy. So,for effective evaluation of model, analysis of ROC AUC curve is prefered. The ROC curve is plotted with TPR against the FPR where TPR is on the y-axis and FPR is on the x-axis.

In order to control the model's performance, avoid overfitting and to have a gener-alizable estimation of the quality of the model obtained we used k-Fold Cross Validation.Cross-validation is a technique of analyzing independent data set and is used to predict accurately the predictive model. Generally, cross validation works by partitioning a data into complementary subsets and performing analysis on the subset. In our research, we perform 5 folds cross validation in order to obtain the optimum result from the dataset.With 5 fold cross-validation, the data was randomly partitioned into 5 subsamples.From the 5 subsamples, one subsample was retained for validation and testing; while the remaining 4 subsamples were used for training data.Then, the cross validation process were repeated 5 times (folds), with each of the 5 subsamples are used once as the validation data. The 5 results from the folds are then averaged to produce a single estimation. The advantage of using this technique is,it will repeat the random sub.sampling i.e. all observations were used for both training and testing and each observation was used exactly once for validation.

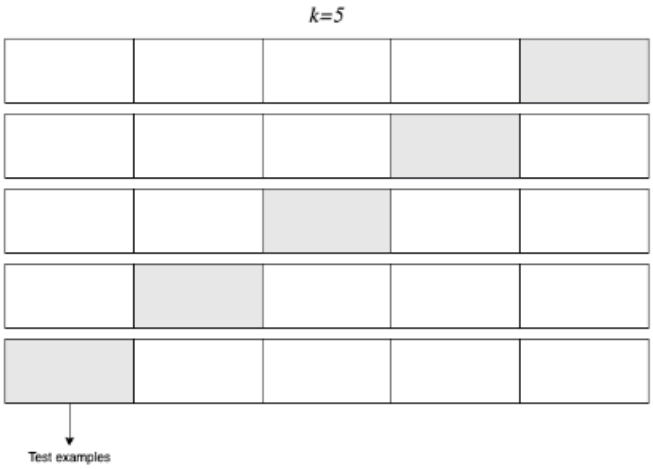

Figure 4.5: K-fold cross validation with 5 folds [81].

4.4 Experiment and Discussion

To benchmark the performance of MLAs,we implemented classifier algorithms on two set of BoT-IoT the real dataset "imbalanced" dataset and balanced dataset created by using SMOTE technology in real dataset.The dataset is divided into two parts,normal traffic and Ddos botnet traffic Fig. 4.6 shows that the size of the dataset is 92.3 MB ,Fig. 4.7 show the dataset consists of 19 columns and 733705 rows.Fig. 4.8 shows that the dataset is highly imbalance. Then data preprocessing was done to get reliable data.In order to get this I drop the rows containing null values using dropna() function of pandas.

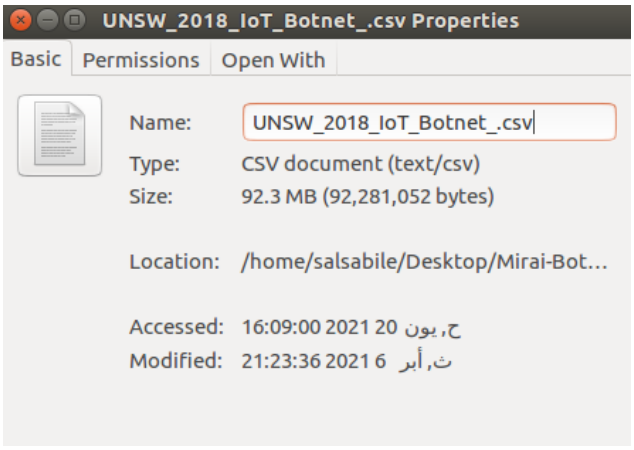

Figure 4.6: Size of dataset.

```
size of the dataset
Rows : 733705 columns: 19
```

Figure 4.7: Size of dataset.

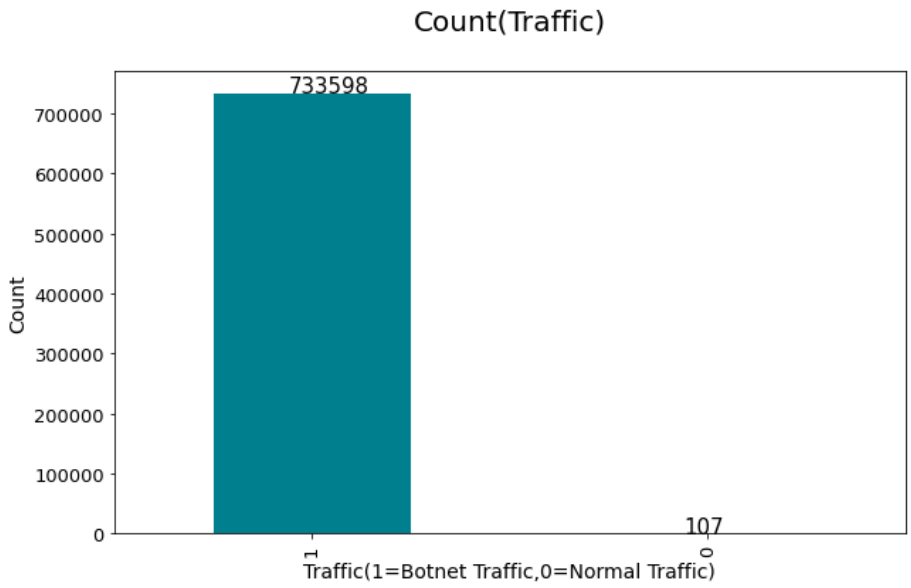

Figure 4.8: Bar graph showing class imbalance in data-set.

4.4.1 Transformation

In BoT-IoT dataset we use there are many categorical features containing non-numeric data,We assign numeric values to protocol names and address IP. For subcategory feature, we assign numbers starting from 0 to 6. Fig. 4.9 and 4.10 shows data before data transformation and Normalization.

	proto	saddr	stddev	min	state_number	mean	max
0	udp	192.168.100.150	0.226784	4.100436	4	4.457383	4.719438
1	tcp	192.168.100.148	0.451998	3.439257	1	3.806172	4.442930
2	udp	192.168.100.149	1.931553	0.000000	4	2.731204	4.138455
3	tcp	192.168.100.148	0.428798	3.271411	1	3.626428	4.229700
4	tcp	192.168.100.149	2.058381	0.000000	3	1.188407	4.753628
5	tcp	192.168.100.149	2.177835	0.000000	3	1.539962	4.619887
6	udp	192.168.100.147	1.368196	1.975180	4	3.910081	4.885159

Figure 4.9: First 7 rows of dataset before transformation and Normalization.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 733705 entries, 0 to 733704
Data columns (total 19 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   pkSeqID            733705 non-null  int64
 1   proto              733705 non-null  object
 2   saddr              733705 non-null  object
 3   sport              733705 non-null  object
 4   daddr              733705 non-null  object
 5   dport              733705 non-null  object
 6   seq                733705 non-null  int64
 7   stddev             733705 non-null  float64
 8   N_IN_Conn_P_SrcIP  733705 non-null  int64
 9   min                733705 non-null  float64
 10  state_number       733705 non-null  int64
 11  mean               733705 non-null  float64
 12  N_IN_Conn_P_DstIP  733705 non-null  int64
 13  drate              733705 non-null  float64
 14  srate              733705 non-null  float64
 15  max                733705 non-null  float64
 16  attack             733705 non-null  int64
 17  category           733705 non-null  object
 18  subcategory        733705 non-null  object
dtypes: float64(6), int64(6), object(7)
```

Figure 4.10: Before Data Transformation.

Fig. 5.11 and 5.12 shows data After data transformation.

```
Data Transformation:
   proto  saddr    stddev       min  state_number      mean       max
0      0      0  0.226784  4.100436             4  4.457383  4.719438
1      1      1  0.451998  3.439257             1  3.806172  4.442930
2      0      2  1.931553  0.000000             4  2.731204  4.138455
3      1      1  0.428798  3.271411             1  3.626428  4.229700
4      1      2  2.058381  0.000000             3  1.188407  4.753628
5      1      2  2.177835  0.000000             3  1.539962  4.619887
6      0      3  1.368196  1.975180             4  3.910081  4.885159
```

Figure 4.11: First 7 rows of dataset after transformation.

```
Data columns (total 19 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   pkSeqID            733705 non-null  int64
 1   proto              733705 non-null  int64
 2   saddr              733705 non-null  int64
 3   sport              733705 non-null  int64
 4   daddr              733705 non-null  int64
 5   dport              733705 non-null  int64
 6   seq                733705 non-null  int64
 7   stddev             733705 non-null  float64
 8   N_IN_Conn_P_SrcIP  733705 non-null  int64
 9   min                733705 non-null  float64
 10  state_number       733705 non-null  int64
 11  mean               733705 non-null  float64
 12  N_IN_Conn_P_DstIP  733705 non-null  int64
 13  drate              733705 non-null  float64
 14  srate              733705 non-null  float64
 15  max                733705 non-null  float64
 16  attack             733705 non-null  int64
 17  category           733705 non-null  int64
 18  subcategory        733705 non-null  int64
dtypes: float64(6), int64(13)
```

Figure 4.12: After Data Transformation.

Fig 4.13 shows First 7 rows of dataset after Normalization.

```
Normalization:
   proto     saddr    stddev       min  state_number      mean       max
0   0.00  0.000000  0.090831  0.823303           0.3  0.894736  0.943888
1   0.25  0.066667  0.181034  0.690549           0.0  0.764018  0.888586
2   0.00  0.133333  0.773624  0.000000           0.3  0.548238  0.827691
3   0.25  0.066667  0.171742  0.656848           0.0  0.727937  0.845940
4   0.25  0.133333  0.824422  0.000000           0.2  0.238550  0.950726
5   0.25  0.133333  0.872265  0.000000           0.2  0.309119  0.923978
6   0.00  0.200000  0.547989  0.396585           0.3  0.784876  0.977032
```

Figure 4.13: First 7 rows of dataset after Normalization.

4.4.2 Oversampling minority class

The Fig 4.14 shows the implementation of SMOTE technique used to oversample minority class.

Real time BoT-IoT dataset contains 733705 data, out of which 107 belong to Normal traffic and 733598 belong to Botnet traffic.

After processing data through SMOTE technique, we get 1467196 data containing equal number of botnet and Normal traffic, that is 733598.This makes the dataset class balanced.

```
Before using SMOTE Technology
(733705,16) (733705,)
0 107
1 733598
After using SMOTE Technology
(1467196, 16) (1467196,)
0 733598
1 733598
```

Figure 4.14: Size of dataset before and after SMOTE technique.

4.4.3 Feature Score

After analyzing the dataset,we used feature engineering to select appropriate features. Fig 4.15 shows the most 10 important feature based on ExtraTreesClassifier.

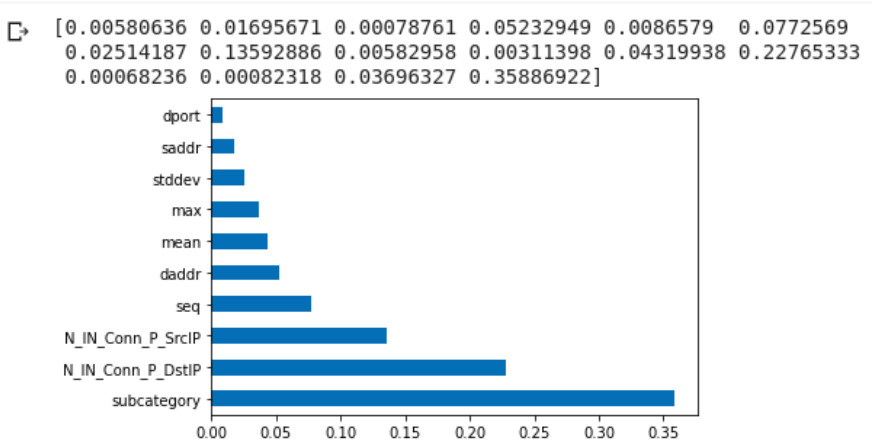

Figure 4.15: Top 10 respective feature score.

Feature	Description
subcategory	Traffic subcategory
N IN Conn P DstIP	Number of inbound connections per destination IP.
N IN Conn P SrcIP	Number of inbound connections per source IP.
seq	Argus sequence number
daddr	Destination IP address
mean	Average duration of aggregated records
max	Maximum duration of aggregated records
stddev	Standard deviation of aggregated records
saddr	Source IP address
dport	Destination port number

Figure 4.16: Feature Description.

4.4.4 Train-Test Split

The dataset was splitted into train and test to evaluate the performance of model.70 percent of data were used for training and 30 percent for test.

4.4.5 Comparison of performance of Machine learning algorithms

- **Results with Bernoulli Naive Bayes:** As seen in Fig. 4.17 for real BoT-IoT dataset, with BernoulliNB algorithm, we observed 99.98% accuracy but Fig 4.22 shows 50% ROC-AUC, and Fig. 4.18 shows very low value in recall and f1-score.This evidently showed that accuracy is generally not helpful in imbalanced data. As we have more than 99 percent botnet traffic and less than 1 percent normal traffic in this dataset this classifier may possibly classify all samples as 1. Thus, we get high accuracy but low ROC AUC . After integrating SMOTE technology,Fig. 4.19,4.21,4.23 showed that we got better value in precision,recall,f1-score,and ROC AUC. This denotes that after using smote technique the BernoulliNB algorithm is effective to distinguish botnet and normal traffic.

```
naive bayes Without SMOTE technique (10 Feature)
confusion_matrix:


[[     0     29]
 [     0 220083]]
accuracy score:


0.9998682488914734
```

Figure 4.17: BernoulliNB model performance without using SMOTE technique.

	precision	recall	f1-score	support
0	0.00	0.00	0.00	29
1	1.00	1.00	1.00	220083
accuracy			1.00	220112
macro avg	0.50	0.50	0.50	220112
weighted avg	1.00	1.00	1.00	220112

Figure 4.18: BernoulliNB model performance without using SMOTE technique.

```
[[206152  14000]
 [ 19252 200755]]
0.9244545720978101
              precision    recall  f1-score   support

           0       0.91      0.94      0.93    220152
           1       0.93      0.91      0.92    220007

    accuracy                           0.92    440159
   macro avg       0.92      0.92      0.92    440159
weighted avg       0.92      0.92      0.92    440159
```

Figure 4.19: BernoulliNB model performance using SMOTE technique.

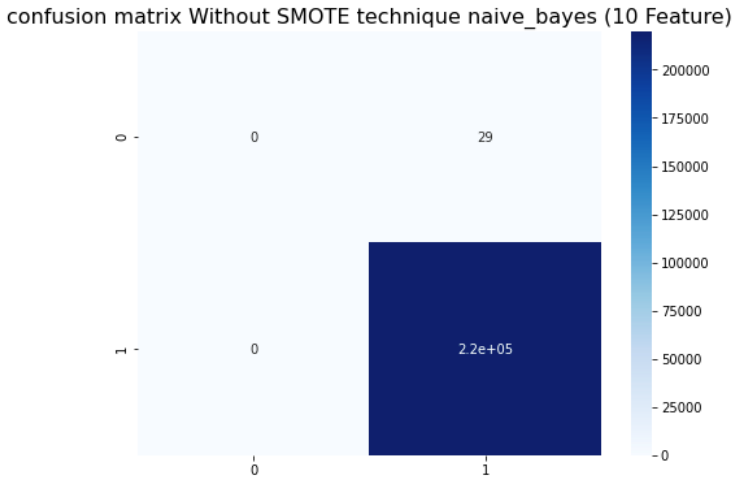

Figure 4.20: matrix confusion for BernoulliNB without using SMOTE technique.

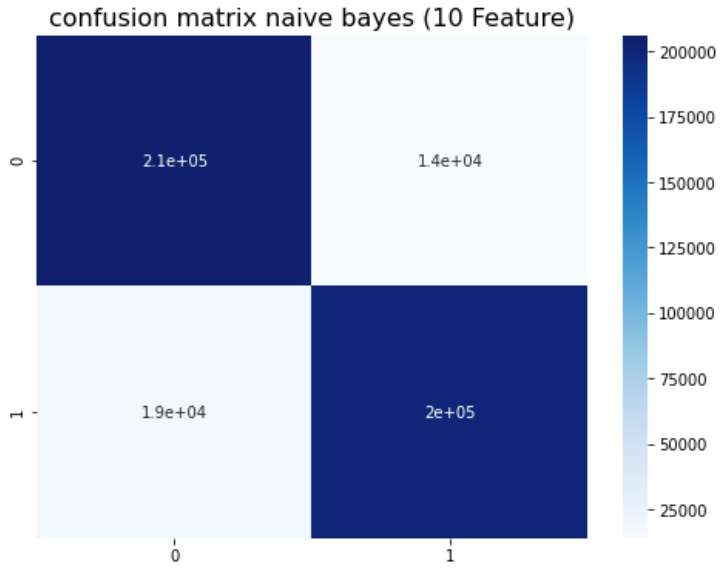

Figure 4.21: matrix confusion for BernoulliNB using SMOTE technique.

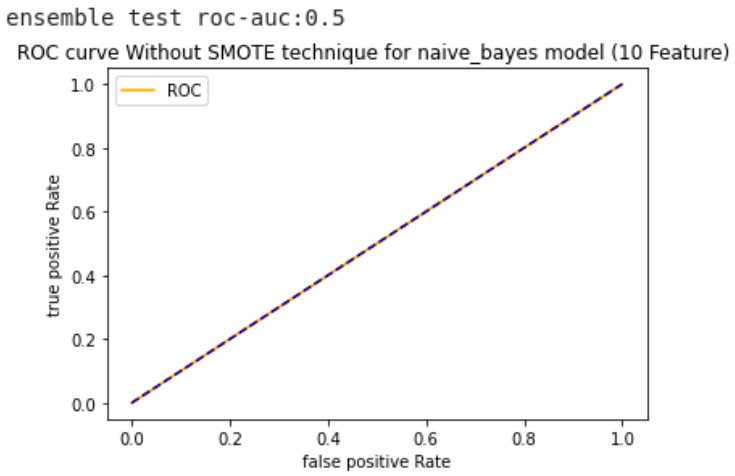

Figure 4.22: ROC AUC graph from BernoulliNB model without using SMOTE Technique.

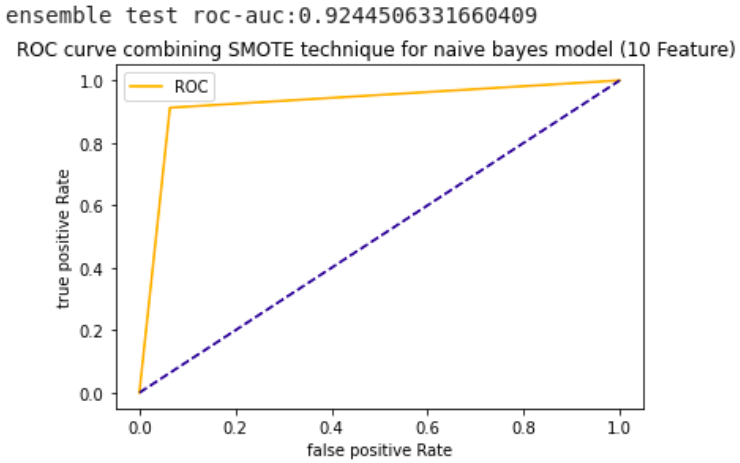

Figure 4.23: ROC AUC graph from BernoulliNB model using SMOTE Technique.

Fig. 4.24 shows BernoulliNB model performance on realtime Class balance dataset using top 8 feature the accuracy and ROC_AUC get better then using top 10 feature 92.65%,92.64%(Fig. 4.26) respectively.

```
naive bayes (8 Feature)


confusion_matrix:
[[219394    906]
 [ 31416 188443]]
accuracy score:


0.9265674449460308

              precision    recall  f1-score   support

           0       0.87      1.00      0.93    220300
           1       1.00      0.86      0.92    219859

    accuracy                           0.93    440159
   macro avg       0.93      0.93      0.93    440159
weighted avg       0.93      0.93      0.93    440159
```

Figure 4.24: BernoulliNB model performance using SMOTE technique (8 Feature).

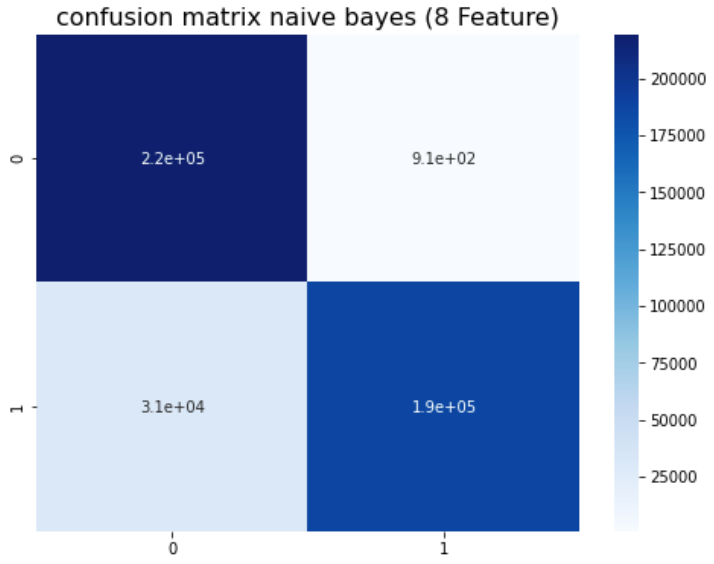

Figure 4.25: matrix confusion for BernoulliNB using SMOTE technique (8 Feature).

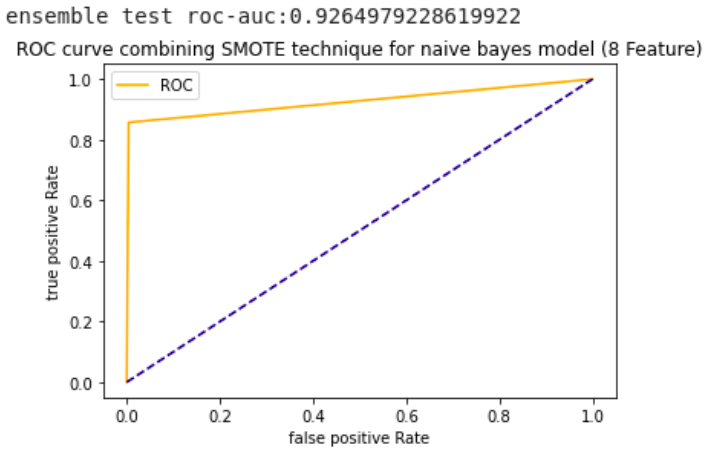

Figure 4.26: ROC AUC graph from BernoulliNB model using SMOTE Technique (8 Feature).

- **Results with KNN:** Fig. 4.27 shows KNN model performance on imablance dataset we get good accuracy 99.98% but with low precision and recall,F1-score.as we see in confusion matrix there's only 13 true positive value,and Fig. 4.31 shows that we get only 72.4% ROC_AUC ,This validate that accuracy percentage is not enough to validate the model performance. after we used SMOTE technology to balanced our dataset even tho the accuracy score get decreased a bit 99.92% (Fig. 4.28) but the ROC_AUC and F1-score,recall,precision increased (Fig 4.28,4.30,4.32) that mean our model performance got better.

```
confusion_matrix:

[[    13     16]
 [    10 220073]]

accuracy score:

0.9998818783164934

              precision    recall  f1-score   support

           0       0.57      0.45      0.50        29
           1       1.00      1.00      1.00    220083

    accuracy                           1.00    220112
   macro avg       0.78      0.72      0.75    220112
weighted avg       1.00      1.00      1.00    220112
```

Figure 4.27: KNN model performance without using SMOTE technique.

```
KNeighborsClassifier (10 Feature)
confusion_matrix:


[[220147      0]
 [   329 219683]]


accuracy score:


0.9992525428311133

              precision    recall  f1-score   support

           0       1.00      1.00      1.00    220147
           1       1.00      1.00      1.00    220012

    accuracy                           1.00    440159
   macro avg       1.00      1.00      1.00    440159
weighted avg       1.00      1.00      1.00    440159
```

Figure 4.28: KNN model performance using SMOTE technique.

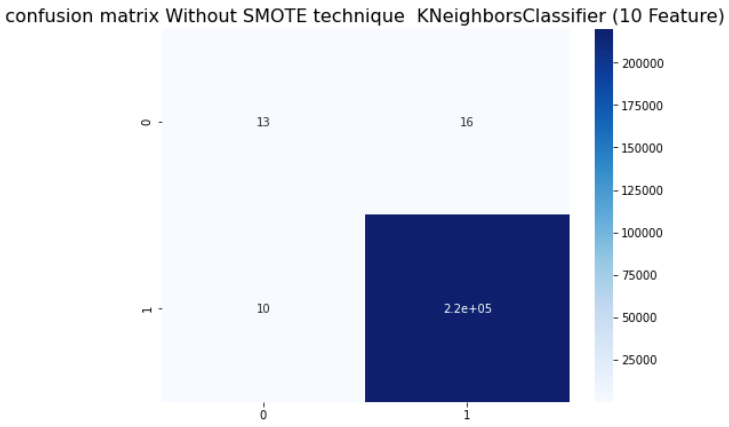

Figure 4.29: matrix confusion for KNN model without using SMOTE technique.

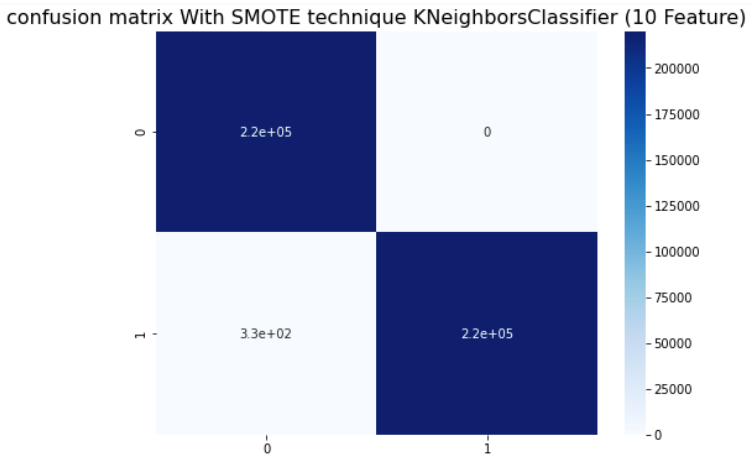

Figure 4.30: matrix confusion for KNN model using SMOTE technique.

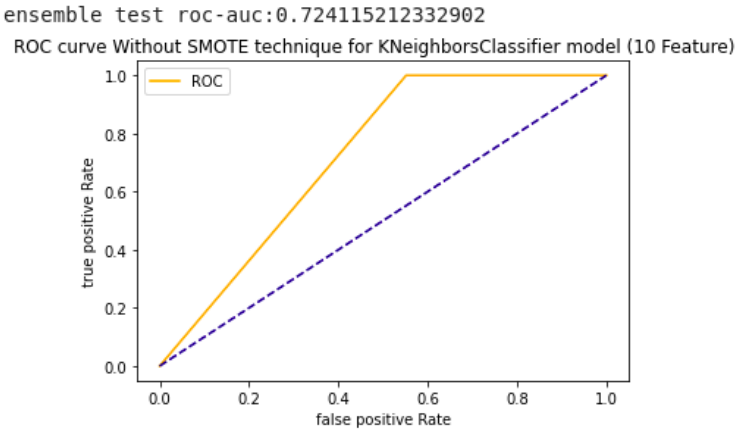

Figure 4.31: ROC AUC graph from KNN model without using SMOTE Technique.

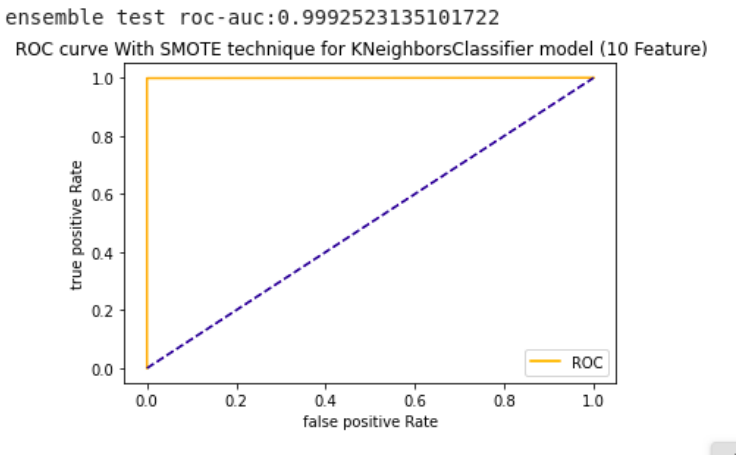

Figure 4.32: ROC AUC graph from KNN model using SMOTE Technique.

Fig. 4.33 shows KNN model performance on realtime Class balance dataset using top 8 feature the accuracy and ROC_AUC get better then using top 10 feature 99.97%,92.97%(Fig. 4.35) respectively.

```
KNeighborsClassifier (8 Feature)
confusion_matrix:

[[219760      0]
 [   116 220283]]

accuracy score:

0.9997364588705445

              precision    recall  f1-score   support

           0       1.00      1.00      1.00    219760
           1       1.00      1.00      1.00    220399

    accuracy                           1.00    440159
   macro avg       1.00      1.00      1.00    440159
weighted avg       1.00      1.00      1.00    440159
```

Figure 4.33: KNN model performance using SMOTE technique (8 Feature).

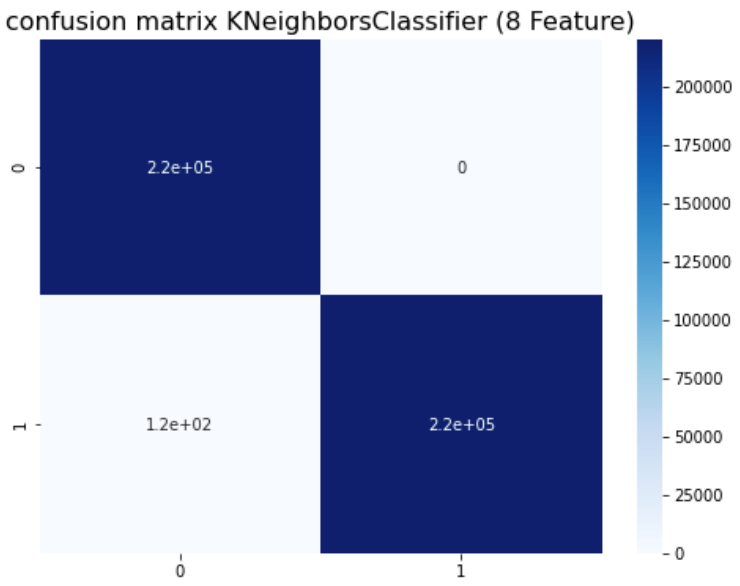

Figure 4.34: matrix confusion for KNN using SMOTE technique (8 Feature).

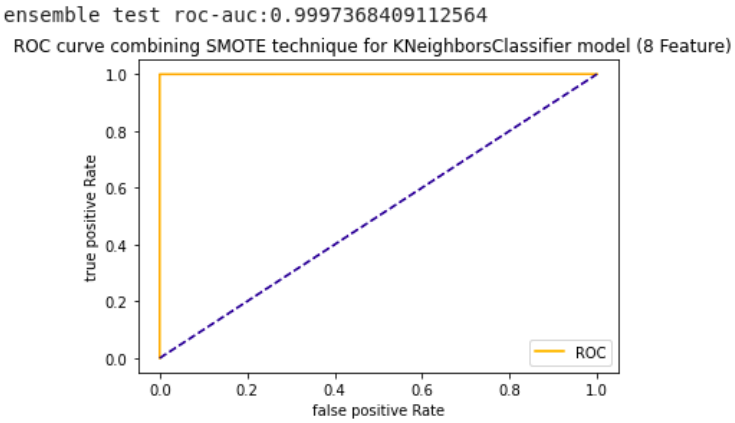

Figure 4.35: ROC AUC graph from KNN model using SMOTE Technique (8 Feature).

- **Results with DecisionTreeClassifier:** Fig 4.36 shows DecisionTreeClassifier model performance on imablance dataset we got good accuracy 99.99% ,the precision and recall and F1-score are good as well(Fig 4.36,4.38),and 96.55%(Fig. 4.40) in our ROC_AUC . After we used SMOTE technology we got the best result in all of algorithms we used, accuracy 99.99%(Fig. 4.37) ,recall and F1-score and precision all was 1(Fig. 4.37,4.4.39),with 99.99% ROC_AUC(Fig 4.41).

```
DecisionTreeClassifier Without SMOTE technique (10 Feature)
confusion_matrix:


[[    27      2]
 [     0 220083]]
accuracy score:


0.9999909137166534


              precision    recall  f1-score   support

           0       1.00      0.93      0.96        29
           1       1.00      1.00      1.00    220083

    accuracy                           1.00    220112
   macro avg       1.00      0.97      0.98    220112
weighted avg       1.00      1.00      1.00    220112
```

Figure 4.36: DecisionTreeClassifier model performance without using SMOTE technique.

```
DecisionTreeClassifier (10 Feature)
confusion_matrix:


[[220147      0]
 [     1 220011]]
accuracy score:


0.9999977280937116


              precision    recall  f1-score   support

           0       1.00      1.00      1.00    220147
           1       1.00      1.00      1.00    220012

    accuracy                           1.00    440159
   macro avg       1.00      1.00      1.00    440159
weighted avg       1.00      1.00      1.00    440159
```

Figure 4.37: DecisionTreeClassifier model performance using SMOTE technique.

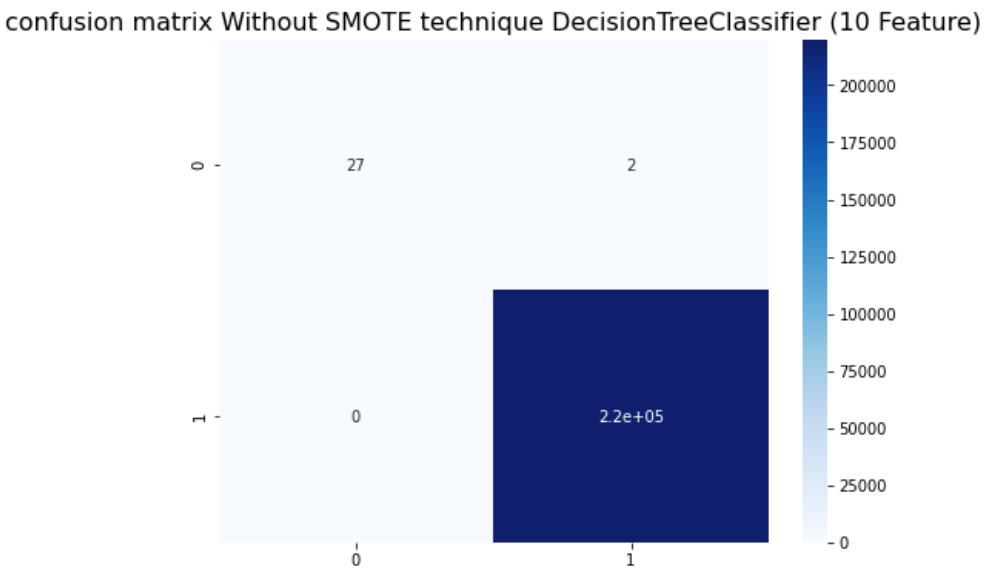

Figure 4.38: matrix confusion for DecisionTreeClassifier model without using SMOTE technique.

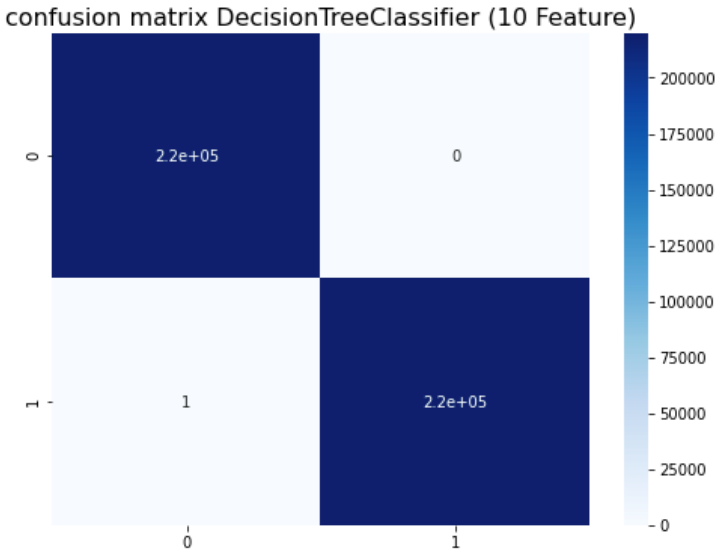

Figure 4.39: matrix confusion for DecisionTreeClassifier model using SMOTE technique.

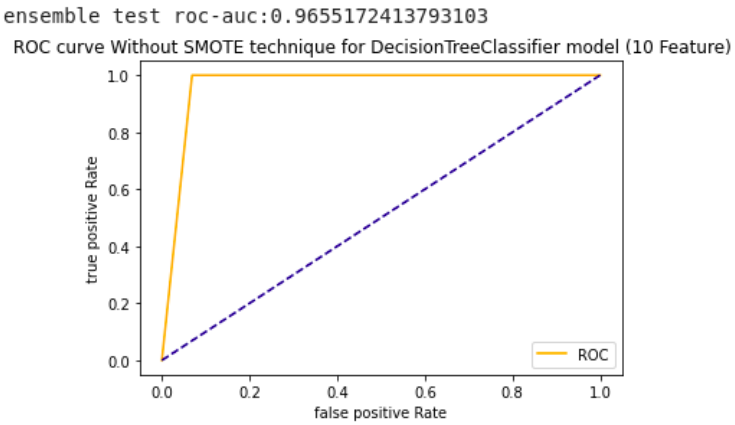

Figure 4.40: ROC AUC graph from DecisionTreeClassifier model without using SMOTE Technique.

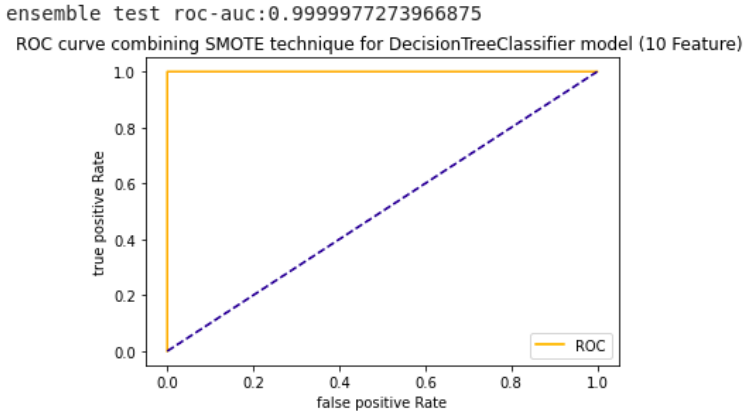

Figure 4.41: ROC AUC graph from DecisionTreeClassifier model using SMOTE Technique.

Fig. 4.42 shows DecisionTreeClassifier model performance on realtime Class balance dataset using top 8 feature,we got the same accuracy and ROC_AUC as we used top 10 feature,99.99%,92.99%(Fig. 4.44) respectively.

```
DecisionTreeClassifier (8 Feature)
confusion_matrix:


[[219616      4]
 [     1 220538]]
accuracy score:


0.999988640468558

              precision    recall  f1-score   support

           0       1.00      1.00      1.00    219620
           1       1.00      1.00      1.00    220539

    accuracy                           1.00    440159
   macro avg       1.00      1.00      1.00    440159
weighted avg       1.00      1.00      1.00    440159
```

Figure 4.42: DecisionTreeClassifier model performance using SMOTE technique (8 Feature).

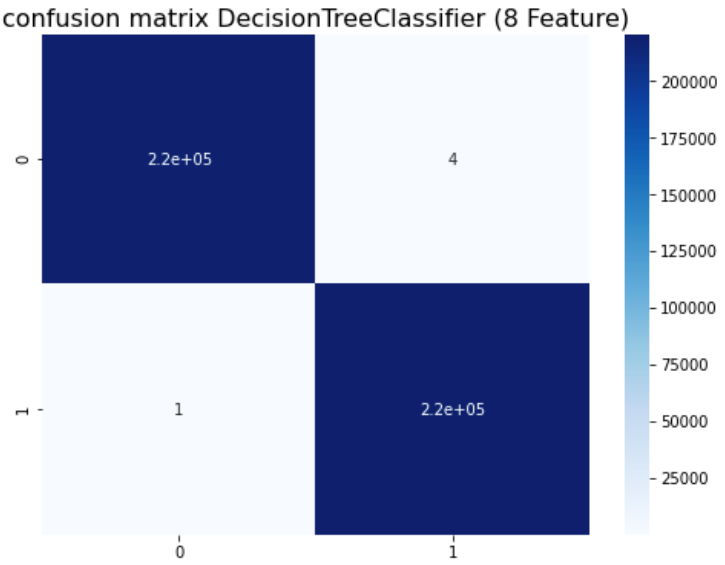

Figure 4.43: matrix confusion for DecisionTreeClassifier using SMOTE technique (8 Feature).

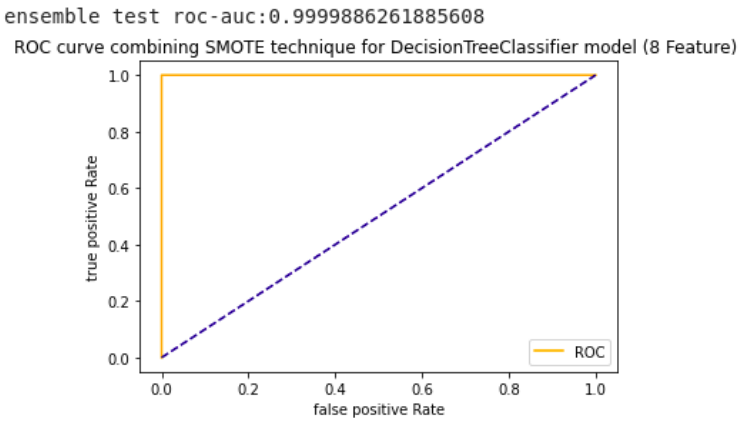

Figure 4.44: ROC AUC graph from DecisionTreeClassifier model using SMOTE Technique (8 Feature).

- **Results with LogisticRegression:** Fig. 4.45 shows LogisticRegression model performance on imablance dataset we got good accuracy 99.99% ,precision and recall and F1-score good as well(Fig. 4.45,4.47),but we only got 93.1%(Fig. 4.49) in our ROC_AUC .

  After we used SMOTE technology we got better result,accuracy 99.99%(Fig. 4.46) ,recall and F1-score and precision all was 1(Fig. 4.46,4.48),with 99.99% ROC_AUC(Fig. 4.50).

```
Logistic Regression Without SMOTE technique (10 Feature)
confusion_matrix:


[[    25      4]
 [    10 220073]]
accuracy score:


0.9999363960165734


              precision    recall  f1-score   support

           0       0.71      0.86      0.78        29
           1       1.00      1.00      1.00    220083

    accuracy                           1.00    220112
   macro avg       0.86      0.93      0.89    220112
weighted avg       1.00      1.00      1.00    220112
```

Figure 4.45: LogisticRegression model performance without using SMOTE technique.

```
Logistic Regression With SMOTE technique (10 Feature)
confusion_matrix:


[[220152      0]
 [    43 219964]]
accuracy score:


0.9999023080295983

              precision    recall  f1-score   support

           0       1.00      1.00      1.00    220152
           1       1.00      1.00      1.00    220007

    accuracy                           1.00    440159
   macro avg       1.00      1.00      1.00    440159
weighted avg       1.00      1.00      1.00    440159
```

Figure 4.46: LogisticRegression model performance using SMOTE technique.

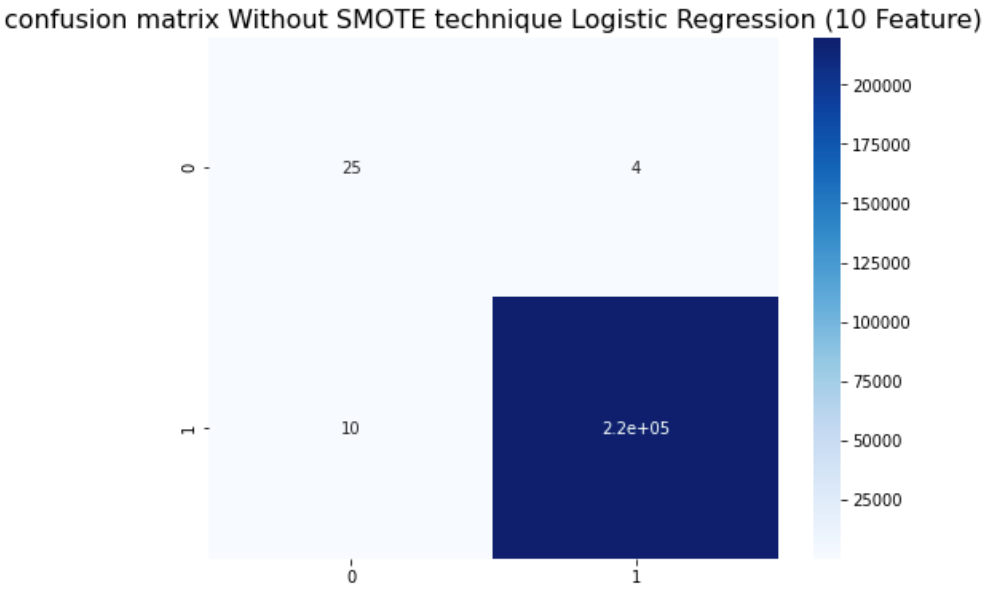

Figure 4.47: matrix confusion for LogisticRegression model without using SMOTE technique.

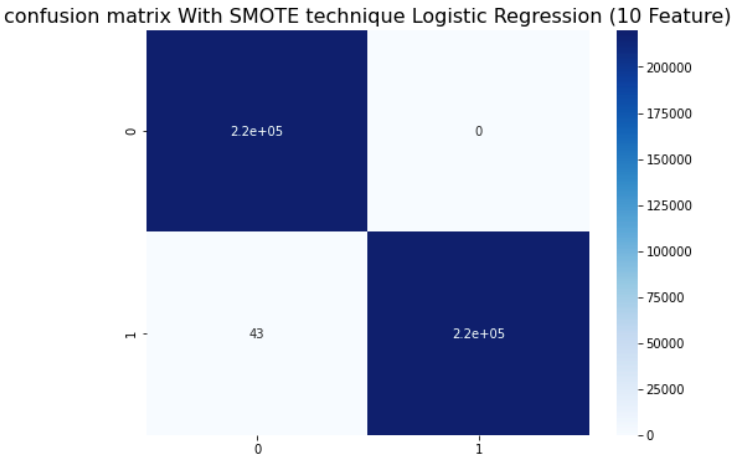

Figure 4.48: matrix confusion for LogisticRegression model using SMOTE technique.

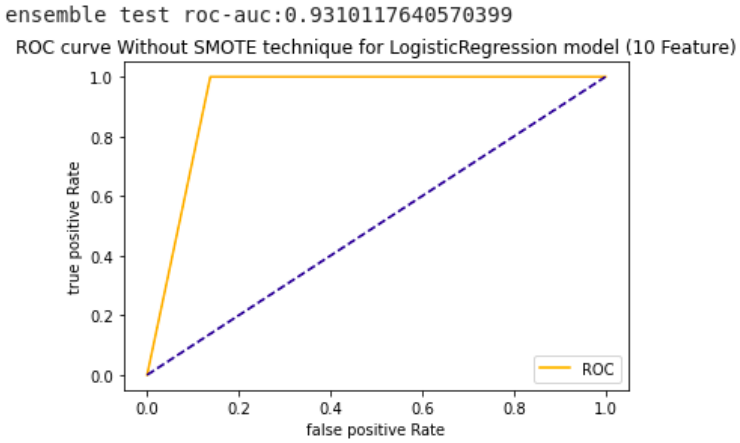

Figure 4.49: ROC AUC graph from LogisticRegression model without using SMOTE Technique.

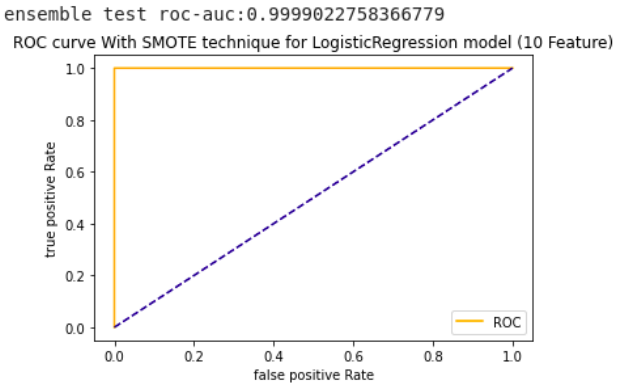

Figure 4.50: ROC AUC graph from LogisticRegression model using SMOTE Technique.

Fig. 4.51 shows LogisticRegression model performance on realtime Class balance dataset using top 8 feature the accuracy and ROC_AUC decreased then using top 10 feature,99.96%,99.96%(Fig. 4.53) respectively.

```
Logistic Regression combining SMOTE technique for LogisticRegression (8 Feature)
confusion_matrix:


[[220300      0]
 [   147 219712]]
accuracy score:


0.9996660297756038


              precision    recall  f1-score   support

           0       1.00      1.00      1.00    220300
           1       1.00      1.00      1.00    219859

    accuracy                           1.00    440159
   macro avg       1.00      1.00      1.00    440159
weighted avg       1.00      1.00      1.00    440159
```

Figure 4.51: LogisticRegression model performance using SMOTE technique (8 Feature).

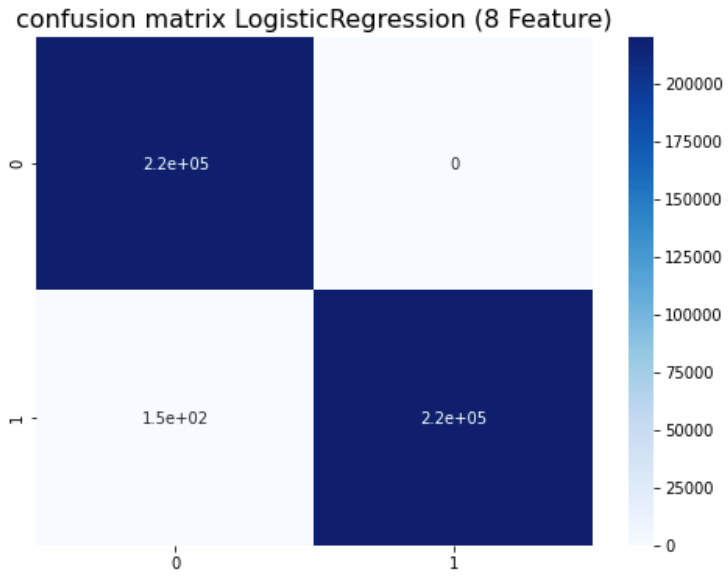

Figure 4.52: matrix confusion for LogisticRegression using SMOTE technique (8 Feature).

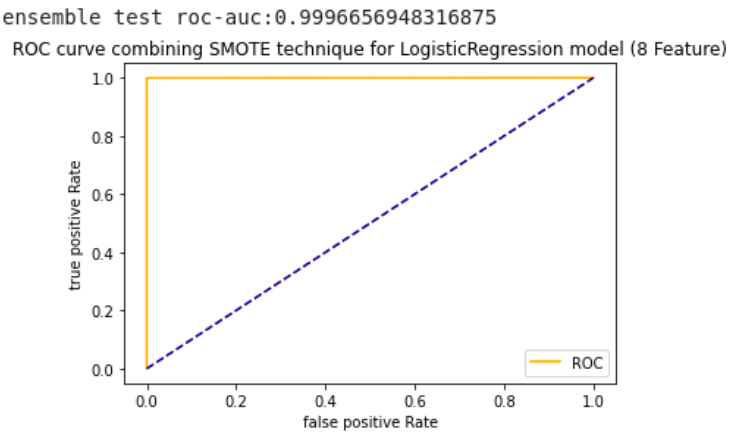

Figure 4.53: ROC AUC graph from LogisticRegression model using SMOTE Technique (8 Feature).

4.4.6 Observations:

Table below shows the results from different MLAs on : imbalance datset,balancae dataset,balance dataset using 8 top feature.using SMOTE technique improve our results,even we got good accuracy in our real data-set but the ROC_AUC was low in BernouliNB and KNN This evidently showed that accuracy is generally not helpful in imbalanced data. using different number of feature didn't give as big different,our accuracy got a bit better when we use 8 feature in most of algorithms except DecisionTreeClassifier.

DecisionTreeClassifier and LogisticRegression was the best algorithms to use in botnet detection system,they give as high accuracy and ROC_AUC in our 3 cases,BernoulliNB was the algorithm that give lower performance.

Fig. 4.56,4.57,4.58,4.59,4.60 shows performance comparison of machine learning algorithms between using SMOTE technology and real datset.

MLAs	Real time data		After SMOTE		8 Feature	
	Acurracy(%)	ROC_AUC(%)	Acurracy(%)	ROC_AUC(%)	Acurracy(%)	ROC_AUC(%)
BernoulliNB	99.98	50	92.44	92.44	92.65	92.64
KNN	99.98	72.41	99.92	99.92	99.97	99.97
DecisionTreeClassifier	99.99	96.55	99.99	99.99	99.96	99.96
LogisticRegression	99.99	93.10	99.99	99.99	99.99	99.99

Figure 4.54: Comparison of MLAs performance.

MLAs	Real time data			After SMOTE			8 feature		
	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
BernoulliNB	0.5	0.5	0.5	0.92	0.92	0.92	0.93	0.93	0.93
KNN	0.78	0.72	0.75	1	1	1	1	1	1
decisiontreeclassifier	1	0.97	0.98	1	1	1	1	1	1
logisticregression	0.86	0.93	0.89	1	1	1	1	1	1

Figure 4.55: Comparison of MLAs performance.

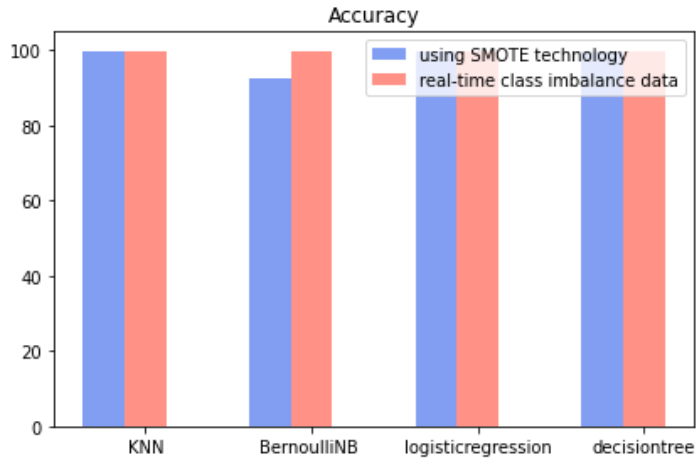

Figure 4.56: Graphical presentation of machine learning Accuracy performance comparison.

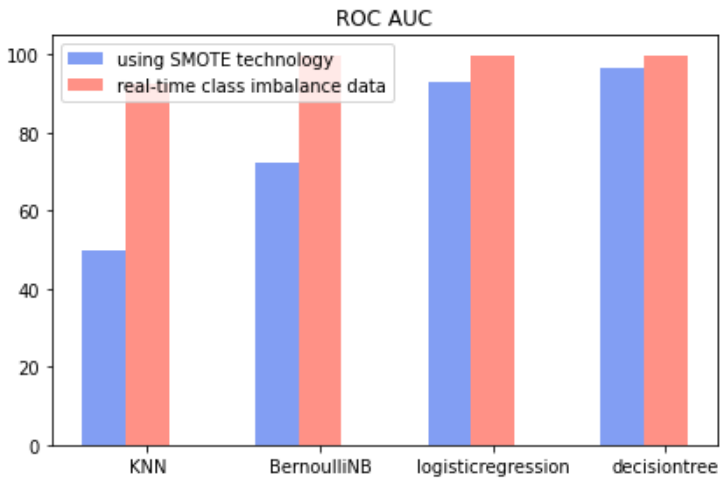

Figure 4.57: Graphical presentation of machine learning ROC_AUC performance comparison.

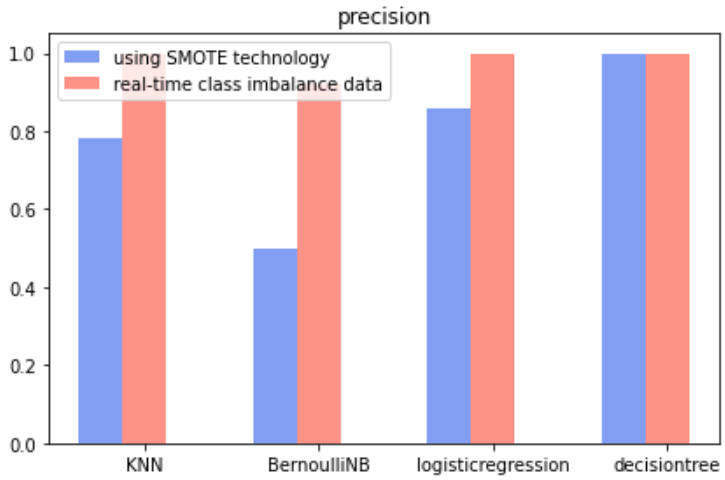

Figure 4.58: Graphical presentation of machine learning precision performance comparison.

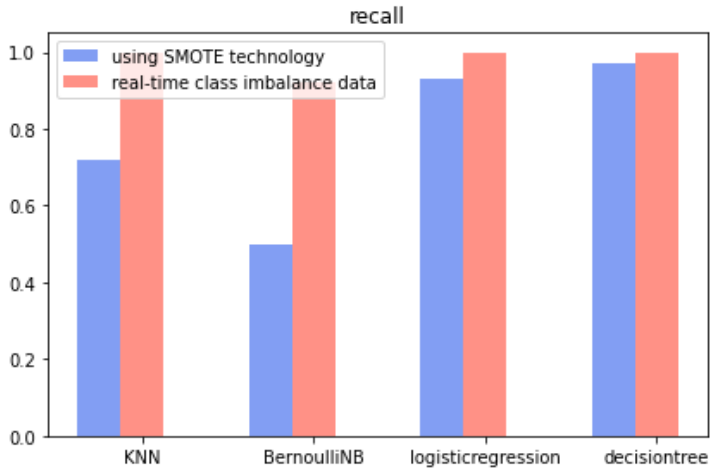

Figure 4.59: Graphical presentation of machine learning recall performance comparison.

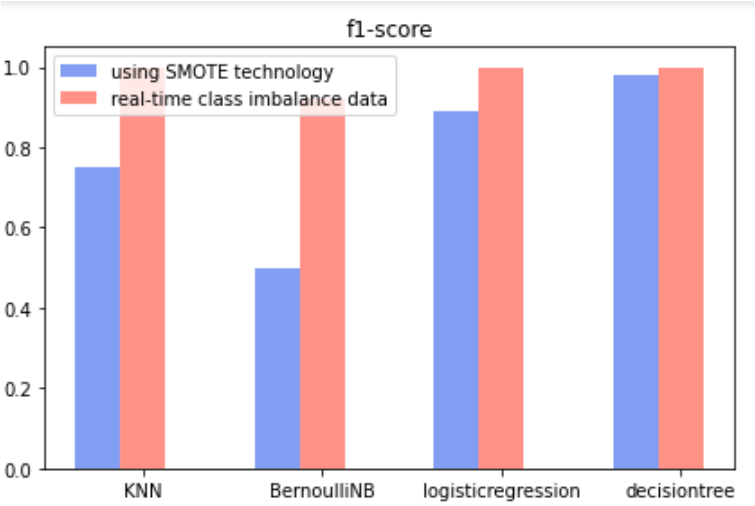

Figure 4.60: Graphical presentation of machine learning f1-score performance comparison.

Fig. 4.61,4.62,4.63,4.64,4.65 shows performance comparison of machine learning algorithms between using 8 and 10 feature.

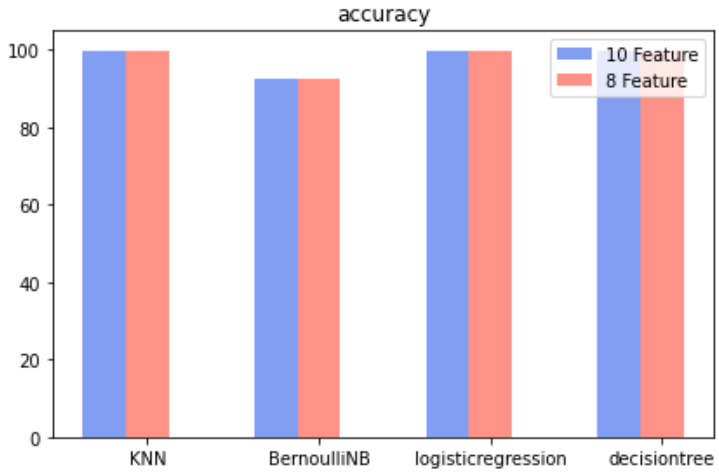

Figure 4.61: Graphical presentation of machine learning Accuracy performance comparison.

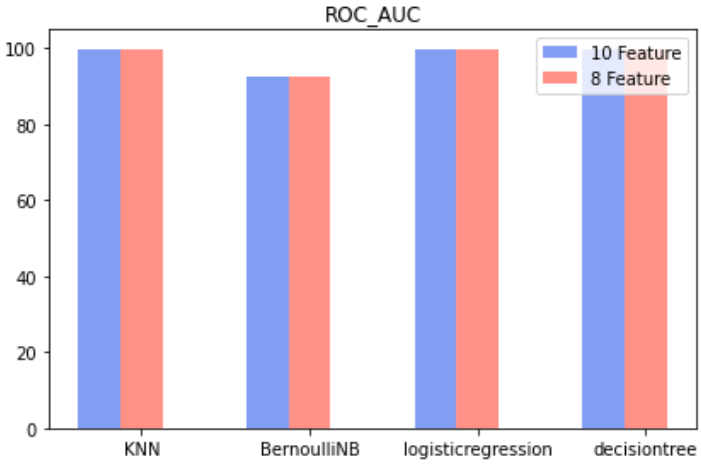

Figure 4.62: Graphical presentation of machine learning ROC_AUC performance comparison.

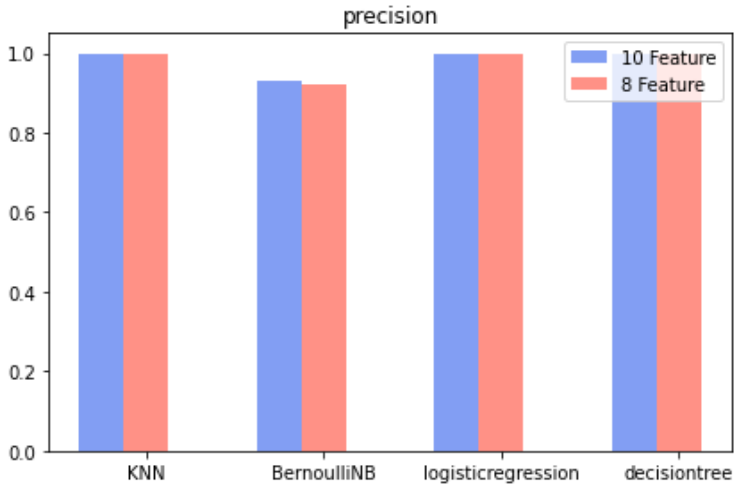

Figure 4.63: Graphical presentation of machine learning precision performance comparison.

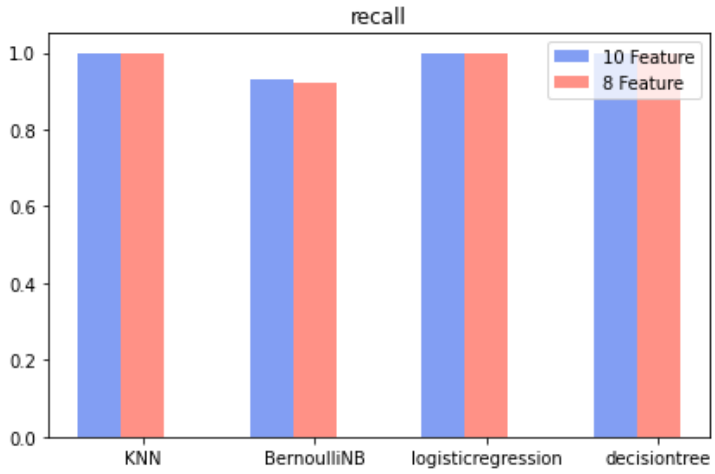

Figure 4.64: Graphical presentation of machine learning recall performance comparison.

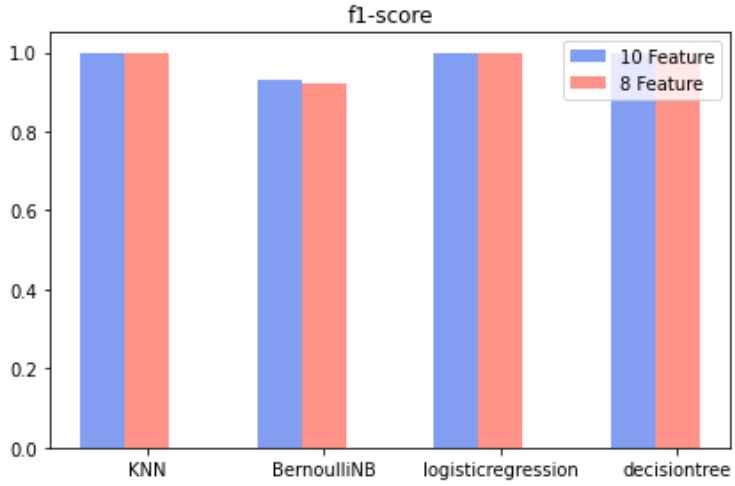

Figure 4.65: Graphical presentation of machine learning f1-score performance comparison.

4.5 Conclusion

In our project,we analyzed machine learning algorithms for Botnet DDoS attack detection.The tested algorithms are KNN,Decision Tree,Logistic Regression and BernoulliNB The evaluation was done on the BoT-IoT dataset.

Our comparison of botnet detection models on real time imbalanced dataset and balanced dataset considerably help to enrich our research.we got good accracy in KNN and BernoulliNB in imbalanced dataset but ROC_AUC and f1-score and precision and recall were low,This shows that accuracy we got from imbalanced dataset maybe illusory.Decision Tree,Logistic Regression were good in imbalanced dataset

After combining SMOTE technology, we got more stable accuracy and ROC_AUC with similar range of values on precision, recall and f1-score in all of our algorithms.This proves that with implementation of SMOTE technology we can get more reliable performance of the model.and then we used less features with the balanced dataset we got better performance in most of the algortihms.

Based on the findings, Decision Tree,Logistic Regression algorithm was found to be the most reliable in botnet detection.

General conclusion

Nowadays,the enormous number of IoT devices is usually more insecure than ordinary desktop computers.So they pose security threats to information systems.IoT botnet is one of these threats that can effect on the IoT devices.

In this thesis, we have established an approach for detecting botnets on IoT devices using KNN,Decision Tree,Logistic Regression and BernoulliNB.which have been trained in BoT-IoT data.We have done the experiment on real time data-set and balanced dataset and presented the effect of imbalance data and its impact on machine learning.After combining SMOTE technology,we got more stable accuracy and ROC AUC with similar range of values on precision,recall and f1-score.This proves that with implementation of SMOTE technology we can get more reliable performance of the model.Based on the findings,Decision Tree and Logistic Regression algorithms was found to be the most reliable in botnet detection.

References

[1] Rwan Mahmoud, Tasneem Yousuf, Fadi Aloul, Imran Zualkernan,Internet of Things (IoT) Security:Current Status, Challenges and Prospective Measures,2015.

[2] Harika Devi Kotha 1,V Mnssvkr Gupta,IoT Application-A Survey,2018.

[3] Shanzhi Chen, Senior Member, IEEE, Hui Xu, Dake Liu, Senior Member, IEEE, Bo Hu, and Hucheng Wang,A Vision of IoT: Applications, Challenges,and Opportunities With China Perspective,2014.

[4] https://www.fracttal.com/en/blog/the-9-most-important-applications-of-the-internet-of-things,9/06/2021.

[5] https://www.businessinsider.com/internet-of-things-devices-examples,9/06/2021.

[6] Muhammad Burhan,Rana Asif Rehman,Bilal Khan and Byung-Seo Kim,IoT Elements, Layered Architectures and Security Issues: A Comprehensive Survey,24 August 2018.

[7] Yang Lu,Li Da Xu,Internet of Things (IoT) Cybersecurity Research: A Review of Current Research Topics.

[8] ttps://www.link-labs.com/blog/6lowpan-vs-zigbee,9/06/2021.

[9] Konstantinos Vidakis, Argyro Mavrogiorgou, Athanasios Kiourtis, Dimosthenis Kyriazis,A Comparative Study of Short-Range Wireless Communication Technologies for Health Information Exchange,12-13 June 2020.

[10] Abdul Salam,Internet of Things for Sustainable Community Development.

[11] Yasmine HARBI,Security in Internet of Things,2021.

[12] Jie Lin, Wei Yu, Nan Zhang, Xinyu Yang, Hanlin Zhang, and Wei Zhao. A survey on internet of things: Architecture, enabling technologies, security and privacy,and applications. IEEE Internet of Things Journal, 4(5):1125–1142, 2017.

[13] Andrea, I., Chrysostomou, C., Hadjichristofi, G., Feb. 2015. Internet of things: security vulnerabilities and challenges. In: Proc. IEEE Symp. Computers and Communication,Larnaca, Cyprus, pp. 180–187.

[14] Bekara, C., 2014. Security issues and challenges for the IoTbased smart grid. Procedia Comput. Sci. 34, 532–537.

[15] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, "A survey on jamming attacks and countermeasures in WSNs."Communications Surveys Tutorials, IEEE 11, no. 4 (2009): 42-56.

[16] H. Tobias, et al. "Security Challenges in the IP-based Internet of Things." Wireless Personal Communications 61, no. 3 (2011): 527-542.

[17] M.U. Farooq, M. Waseem, A. Khairi, S. Mazhar, "A Critical Analysis on the Security Concerns of Internet of Things (IoT)", International Journal of Computer Applications (0975 8887), Volume 111 - No. 7, February 2015.

[18] Li, Hong, Y. Chen, and Z. He. "The Survey of RFID Attacks and Defenses." 8th International Conference on IEEE Wireless Communications, Networking and Mobile Computing (WiCOM), 2012.

[19] I. Andrea, C. Chrysostomou and G. Hadjichristofi, "Internet of Things: Security vulnerabilities and challenges," 2015 IEEE Symposium on Computers and Communication (ISCC), pp.180-187,Larnaca, 2015.

[20] G. Han, L. Xiao, and H. V. Poor, "Two-dimensional anti-jamming communication based on deep reinforcement learning," in Proc. IEEE Int. Conf. Acoustics Speech and Signal Processing, New Orleans, LA, Mar. 2017, pp. 2087–2091.

[21] Pal Varga, Sandor Plosz, Gabor Soos,Csaba Hegedus,Security Threats and Issues in Automation IoT,2017.

[22] E. Alsaadi and A. Tubaishat, “Internet of Things : Features , Challenges , and,” vol. 4, no. 1, pp. 1–13, 2015.

[23] Belapurkar, A. (2009). Distributed systems security: Issues, processes, and solutions. Chichester, UK: John Wiley Sons

[24] Mukrimah Nawir, Amiza Amir , Naimah Yaakob , Ong Bi Lynn.Internet of Things (IoT): Taxonomy of Security Attacks,August 11-12, 2016.

[25] Lakhmi C. Jain, George A. Tsihrintzis, Valentina E. Balas, Dilip Kumar Sharma.Data Communication and Networks.

[26] Phillip Lee, Andrew Clark, Linda Bushnell, and Radha Poovendran. A passivity framework for modeling and mitigating wormhole attacks on networked control systems. IEEE Transactions on Automatic Control, 59(12):3224–3237, 2014.

[27] Mian Muhammad Ahemd,Munam Ali Shah,Abdul Wahid.IoT Security: A Layered Approach for Attacks & Defenses.2017.

[28] Evaluating Security Threats for each Layers of IoT System.Hamza,M.Junaid Arshad.13 January 2020.

[29] K.Somasundaram, Dr.K.Selvam.IOT ± Attacks and Challenges.September 2018.

[30] Debabrata Singh, Pushparaj,Manish Kumar Mishra, Anil Lamba, Sharabanee Swagatika.Security Issues In Different Layers Of IoT And Their Possible Mitigation.INTERNATIONAL JOURNAL OF SCIENTIFIC TECHNOLOGY RESEARCH VOLUME 9, ISSUE 04, APRIL 2020.

[31] Otmane El Mouaatamid, Mohammed Lahmer ,Mostafa Belkasmi.Internet of Things Security: Layered classification of attacks and possible Countermeasures. 2016.

[32] Smita Dange.IoT Botnet: The Largest Threat to the IoT Network.September 2019.

[33] N.S.Raghava,Divya Sahgal,Seema Chandna.Classification of Botnet Detection Based on Botnet Architechture.2012.

[34] Brandon Shirely, Chad D.Mano .Sub-Botnet Coordination Using Tokens in a Switched Network".

[35] Ihsan Ullah,Naveed Khan,Hatim Aboalsamh.Survey on botnet: Its architecture, detection, prevention and mitigation.April 2013.

[36] Jayasree Sengupta , Sushmita Ruj, Sipra Das Bit, A Comprehensive Survey on Attacks, Security Issues and Blockchain Solutions for IoT and IIoT.November 4, 2019.

[37] Sudhir T. Bagade,Mayuri A. Bhabad.Internet of Things: Architecture, Security Issues and Countermeasures.14, September 2015.

[38] João Marcelo Ceron, Klaus Steding-Jessen Cristine Hoepers, Lisandro Zambenedetti Granville ,Cíntia Borges Margi.Improving IoT Botnet Investigation Using an Adaptive Network Layer.2019.

[39] V. Kansal , M Dave.DDoS attack isolation using moving target defense. dec 2017.

[40] M. Dibaei, X. Zheng, K. Jiang, S. Maric, R. Abbas, S. Liu, Y. Zhang, Y. Deng, S. Wen, J. Zhang, Y. Xiang, and S. Yu. An overview of attacks and defences on intelligent connected vehicles.2019.

[41] Satish Pokhrel, Robert Abbas, Bhulok Aryal.IoT Security: Botnet detection in IoT using Machine learning.

[42] W. Richert, L. P. Coelho, "Building Machine Learning Systems with Python", Packt Publishing Ltd., ISBN 978-1-78216-140-0.

[43] Ayon Dey,"Machine Learning Algorithms: A Review",(IJCSIT) International Journal of Computer Science and Information Technologies,Vol.7(3),2016,1174-1179.

[44] Judith Hurwitz,Daniel Kirsch,"Machine Learning",2018.

[45] Li Yang,Abdallah Shami,"On hyperparameter optimization of machine learning algorithms: Theory and practice",Department of Electrical and Computer Engineering,University of Western Ontario, 1151 Richmond St,London,ON N6A 3K7,Canada.

[46] Jafar Alzubi,Anand Nayyar, Akshi Kumar,"Machine Learning from Theory to Algorithms: An Overview",2018.

[47] Minton S,Zweben M.Learning,Planning,and Scheduling: An Overview.In Machine Learning Methods for Planning 1993 (pp. 1-29).

[48] Sejnowski T.Net talk:A parallel network that learns to read aloud.Complex Systems 1987;1:145-68.

[49] Han J,Cai Y,Cercone N.Data-driven discovery of quantitative rules in relational databases,IEEE Transactions on Knowledge & Data Engineering,1993 Feb 1(1):29-40.

[50] Chen JX.The evolution of computing:AlphaGo.Computing in Science & Engineering.2016 Jul;18(4):4-7.

[51] Taiwo Oladipupo Ayodele,Machine Learning Overview,University of Portsmouth United Kingdom,2010.

[52] C.Gambella, B. Ghaddar, J. Naoum-Sawaya, Optimization Models for Machine Learning: A Survey (2019) 1–40, http://arxiv.org/abs/1901.05331.

[53] R. S. Sutton, “Introduction: The Challenge of Reinforcement Learning”, Machine Learning, 8, Page 225-227, Kluwer Academic Publishers, Boston,1992.

[54] L. P. Kaelbing, M. L. Littman, A. W. Moore, “Reinforcement Learning: A Survey”, Journal of Artificial Intelligence Research, 4,Page 237-285, 1996.

[55] S. B. Hiregoudar, K. Manjunath, K. S. Patil,“A Survey: Research Summary on Neural Networks”, International Journal of Research in Engineering and Technology, ISSN: 2319 1163, Volume 03, Special Issue 03, pages 385-389, May, 2014.

[56] S.B. Kotsiantis,“Supervised Machine Learning: A Review of Classification Techniques”, Informatica 31 (2007) 249-268.

[57] Vladimir Nasteski,An overview of the supervised machine learning methods,University "St. Kliment Ohridski" - Bitola,2018.

[58] Andrew Ng.(2012) CS229 Lecture notes Machine Learning - Supervised learning.

[59] Salima Omar,Asri Ngadi,Hamid H. Jebur,Machine Learning Techniques for Anomaly Detection:An Overview,October 2013.

[60] Osisanwo F.Y, Akinsola J.E.T, Awodele O, Hinmikaiye J. O Olakanmi O,Akinjobi J,Supervised Machine Learning Algorithms:Classification and Comparison,3 June 2017.

[61] WANG Hua,ZHOU Lijuan,MA Cuiqin,A Brief Review of Machine Learning and its Application,2009.

[62] Bernhard Schölkopf, Alexander J Smola, Francis Bachet al. Learning with kernels: support vector machines, regularization, optimiza- tion, and beyond. MIT press, 2002.

[63] J.Brownlee.Master Machine Learning Algorithms. Australia, 2017.

[64] Salah eddine Nacer.Sentiment Analysis Machine Learning Deep Learning based approach.September 15, 2020.

[65] https://www.javatpoint.com/machine-learning-life-cycle.10/06/2021.

[66] Mohamed Mehdi BOURAHLA.Machine learning for Sign Language Recognition: Application on Smart Buildings.2020.

[67] https://cloudstor.aarnet.edu.au/plus/s/umT99TnxvbpkkoE?path=%2FCSV.20/06/2021.

[68] https://research.unsw.edu.au/projects/bot-iot-dataset.20/06/2021.

[69] S.B. Kotsiantis,D. Kanellopoulos , P. E. Pintelas.Data Preprocessing for Supervised Leaning.2006.

[70] Salvador García,Julin Luengo,Francisco Herrera.Data Preprocessing in Data Mining.

[71] Mohamed Idhammad,Karim Afdel ,Mustapha Belouch.Semi-supervised machine learning approach for DDoS detection.2018.

[72] https://www.jeremyjordan.me/feature-selection/.21/06/2021.

[73] Piyasak Jeatrakul, Kok Wai Wong, and Chun Che Fung.Classification of Imbalanced Data by Combining the Complementary Neural Network and SMOTE Algorithm.

[74] https://research.google.com/colaboratory/faq.htmlresource-limits.22/06/2021.

[75] https://www.w3schools.com/python/numpy/numpy_intro.asp.22/06/2021.

[76] https://matplotlib.org/.22/06/2021.

[77] Satish Pokhrel, Robert Abbas, Bhulok Aryal.IoT Security: Botnet detection in IoT using Machine learning.

[78] Mahmudul Hasan, Md. Milon Islam, Md Ishrak Islam Zarif, M.M.A. Hashem.Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches.2019.

[79] Faisal Hussain,Syed Ghazanfar Abbas Muhammad Husnain,Ubaid U. Fayyaz Farrukh Shahzad,Ghalib A. Shah.IoT DoS and DDoS Attack Detection using ResNet.

[80] https://www.mygreatlearning.com/blog/roc-curve/.22/06/2021.

[81] Stefanos Kiourkoulis.DDoS datasets:Use of machine learning to analyse intrusion detection performance.

[82] Sanket Biswas,Navoneel Chakrabarty.Navo Minority Over-sampling Technique (NMOTe): A Consistent Performance Booster on Imbalanced Datasets.June 2020.

[83] Stephan Spiegel.Cost-Sensitive Learning for Predictive Maintenance.

[84] NarasimhaMallikarjunan, K.,Bhuvaneshwaran, A., Sundarakantham, K., Mercy Shalinie, S.(2019).DDAM: Detecting DDoS attacks using machine learning approach.

[85] L. Xiao, X. Wan, X. Lu, Y.Zhang, D. Wu. Iot security techniques based on machine learning.2018.

[86] Tao, Y., Yu, S. DDoS attack detection at local area networks using information theoretical metrics.

[87] Brun, O., Yin, Y., Gelenbe, E. (2018). Deep learning with dense random neural network for detecting attacks against IoT-connected home environments.

[88] Fok K, Zheng L, Watt K, Su L.Thing V (2018) Automated Botnet traffic detection via machine learning. In: Conference: TENCON 2018.

[89] Yan Meng, Wei Zhang, Haojin Zhu, Xuemin (Sherman) Shen.Securing Consumer IoT in the Smart Home:Architecture, Challenges, and Countermeasures.2018.

[90] Constantinos Kolias,Georgios Kambourakis,Angelos Stavrou,Jeffrey Voas.DDoS in the IoT: Mirai and Other Botnets.2017.

[91] Vladimir Nasteski.An overview of the supervised machine learning methods.December 2017.