Addis Ababa Institute of Technology (AAiT)

MASTER THESIS

Performance Evaluation of Supervised Machine Learning Algorithms to Detect IP Spoofing Attack: The Case of Ethio telecom LTE Network

Author:

Surafel.F

Advisor:

Dr. Yalemzewd.N

A thesis submitted to

School of Electrical and Computer Engineering

Addis Ababa Institute of Technology

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Telecommunication Network Engineering

February 23, 2020

Declaration

I, the undersigned, declare that the thesis comprises my work in compliance with internationally accepted practices; I have fully acknowledged and referred to all materials used in this thesis work.

Author Name	Signature

Addis Ababa University
Addis Ababa Institute of Technology
School of Electrical and Computer Engineering
Graduate program in Telecommunication Engineering

This is to certify that the thesis prepared by **Surafel Fikre**, entitled *Performance Evaluation of Supervised Machine Learning Algorithms to Detect IP Spoofing Attack: The Case of Ethio telecom LTE Network*. And submitted in partial fulfillment of the requirements for the degree of Master of Science Telecommunication Engineering complies with the regulations of the University and meets the accepted standards concerning originality and quality.

Signed by the Examining Committee:

Internal Examiner	Signature	Date
External Examiner	Signature	Date
Advisor	Signature	Date
Co-Advisor	Signature	Date

Dean, School of Electrical and Computer Engineering

Abstract

The mobile communication system revolutionized the way people communicate, entertain, doing their business and educate. This results in the need and demand for mobile and Internet users to increasing every day. Ethio telecom is a discoverer market in Eastern Africa with 66.8 million mobile connections as of August 2018. According to Growth and Transformation Plan 2 (GTP2) of the Federal Democratic Republic of Ethiopia (FDRE), the total mobile subscriber is expected to reach 103 million and the mobile broadband share will be estimated to 35 million subscribers by 2020. Based on the company marketing report, 85.9% of the revenue is generating from mobile services.

GPRS Tunneling Protocol (GTP) is the pivotal protocol used in Long Term Evolution (LTE) to assign the Internet Protocol (IP) addresses to mobile terminals and manages the data communication path in a mobile data network. IP spoofing attack is one of the most significant attacks in the IP based communication system and it is used as a stepping stone for most of the attacks. Ethio telecom deployed LTE since 2014, in 2018 there were 300,000 subscribers. This technology is starting to attract the intention of users as well as the company and it is expected to be the next mobile communication technology. Dong W. Kang et al. conducted a detection approach of IP spoofing attacks in a 3G network and several studies are conducted in machine learning-based network anomalies detection methodologies. However, to the best of researches knowledge, there is no specific research that is conducted on machine learning-based IP spoofing attack detection on the LTE network.

This study analyzes a machine learning-based IP spoofing attack detection system. Three supervised machine-learning classifiers namely: Logistic Regression (LR), K- Nearest Neighbor (KNN) and Gaussian Nave Bayes (GNB) are evaluated.The evaluation is based on best-suited metrics such as; sensitivity, specificity, precision, False Positive Rate (FPR) and computational time rather than stick on generic metrics like accuracy. Even though GNB scores the heights sensitivity of 99.93%, considering the other metrics KNN is reasonably considered as the best classifier with a sensitivity of 99.89%, a specificity of 99.96%, precision of 99.93%, FPR of 0.03% and accuracy of 99.94%. However, in most cases of a real situation, KNN is not preferred for practical implementation, since KNN is computationally intensive. As a result, considering computational time metrics as key metric for practical implementation, LR is reasonably recommended as the best classifier with a sensitivity of 99.82%, specificity of 87.56%, precision of 79.87%, FPR of 12.43%, accuracy of 91.62%, training and testing time of 0.506sec and 0.005sec respectively.

Keywords:- LTE, GTP, IP Spoofing, Security, Threats, Attacks, Machine Learning, LR, KNN, GNB

Dedication

This research work is dedicated first to my beloved mother in heaven (GOD) and Savior, Lord Jesus Christ and his mother St. Mary. My spouse; Mrs. Alem Kifle, My little king son Sador Surafel and Nieces and Nephews, Friends, Colleagues, and good and well-wishers.

Acknowledgment

First and foremost, I would like to express my enormous thankfulness to the Almighty God for his grace, mercy and most importantly his everlasting blessings with a wonderful family in place of my mother and favor towards the success of this research. My deep gratefulness goes to my non-alternative/optional mother Mrs. Fikerte W/Mariam, for her incomparable love, kindness, goodwill, and scarification of life for the happiness and accomplishment of my thesis. My profound appreciation goes to my spouse Mrs. Alem Kifle and my little king son Sador Surafel, for their encouragement, support and being key to my life and source of strength. Other thanks go out to my aunt's Mrs. Berhane W/Mariam and Tigist Shiferaw, and all my first cousin's for their support and concern in achieving this research. This thesis would not have been finished without support from my advisor; Dr. Yalemzewd Negash, Dean of Electrical and Computer engineering at Addis Ababa Institute of Technology (AAiT), for giving a significant amount of his time to guiding my work tirelessly, going through all my draft, providing valuable suggestions and constructive criticism, for the improvement of this research effort.

Another gratitude goes to my examiners Dr. Ephrem and Dr. Murad to providing valuable suggestions and constructive criticism, for the improvement of this research. Another gratitude goes to Dr. Birhanu and Dr. Beneyam for always supporting me and so much more. Also, my deepest appreciation extended to Dr. Ing. Dereje Hailemariam for his outstanding academic performance and knowledge sharing throughout our teaching period, especially his support on the development of well-organized research methods and proposals. I would like to extend my gratitude to the entire Ethio telecom and Huawei mobile network engineers, security expertise, administrative staffs and specialists for their consistently supportive approach to mine, provide relevant information and data. Special thanks to Ethio telecom CEO Ms. Frehiwot Tamru for her positive-minded approach and approval of access privilege requests on the mobile core network that makes life easy to work in this research area. Lastly, a big thanks to AAiT, friends, and colleagues for their, directly and indirectly, providing the emotional support I needed to carry this thesis work out to the end.

Table of Contents

List of Figures

List of Tables

List of Acronyms

1G	First Generation
2G	Second Genetation
3G	Third Generation
3GPP	Third Generation Partnership Project
4G	Forth Generation
AAiT	Addis Ababa Institute of Technology
AS	Access Stratum
AUC	Area Under the Curve
BN	Bayes Network
DDoS	Distributed Denial of Service
DoS	Denial of Service
DRC	Democratic Republic Congo
EPC	Evolved Packet Core
E-UTRAN	Evolved Universal Terrestrial Radio Access Network
FDRE	Federal Democratic Republic of Ethiopia
FMS	Fraud Management System
FP	False Positive
FPR	False Positive Rate
GGSN	GPRS (General Packet Radio Service) Serving Node
GNB	Gaussian Nave Bayes

GPRS	General Packet Radio Service
GTP	GPRS Tunneling Protocol
GTP2	Growth and Transformation Plan 2
GTP-C	GTP Control Plane
GTP-U	GTP User Plane
GTPV2	GTP Control Plane Version 2
HSS	Home Subscriber Server
ICT	Information Communication Technology
IoT	Internet of Things
IP	Internet Protocol
IPS	Intrusion Prevention System
ISP	Internet Service Provider
KNN	K- Nearest Neighbor
LMT	Local Maintenance Terminal
LR	Logistic Regression
LTE	Long Term Evolution
LTE-A	LTE-Advanced
MAC	Medium Access Control
MITM	Man in The Middle
ML	Machine Learning
MLP	Multi-Layer Perceptron
MME	Mobility Management Entity
MSISDN	Mobile Station Integrated Service Digital Number
NAS	Non Access Stratum
OTT	Over The Top
PDN	Packet Data Convergence Protocol
P-GW	PDN Gateway
QoS	Quality of Service
RAN	Radio Access Network
RLC	Radio Link Control

ROC	Receiver Operating Characteristics
RRC	Radio Resource Control
S-GW	Serving GPRS Support Node
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
TPR	True Positive Rate
UDP	User Data Protocol
UEs	User Equipment's
UMTS	Universal Mobile Telecommunication System

Chapter 1

Introduction

Mobile communication systems revolutionized the way people communicate, entertain, doing their business and educate. The evolution of this technology is about to keep providing user demands and generate revenue [1]. Nowadays, mobile communication plays a significant role in our day-to-day social and economic activities, which cause the need and demand for mobile and Internet users to increasing every day. At the end of 2016, there were 420 million unique mobile subscribers in Sub-Saharan Africa and will rise to nearly 1 billion by 2020 [2, 3]. Four of the most populated markets in this region are namely; Democratic Republic Congo (DRC), Ethiopia, Nigeria, and Tanzania, which will account for nearly half the 115 million new subscribers expected by 2020.

Ethiopia is a discoverer market in Eastern Africa with one telecom operator called Ethio telecom with 66.8 million mobile connections as of August 2018 [4]. Figure 1.1 and Figure 1.2 illustrate the demand growth trends of mobile communication technologies and the revenue gained from various telecom services in Ethio telecom respectively. According to GTP2 of the Federal Democratic Republic of Ethiopia (FDRE), the total mobile subscriber of the company will be expected to reach 103 million and the mobile broadband share is estimated to be about 35 million by 2020 [5]. The government of Ethiopia believed that this technology is the most enabler for Information Communication Technology (ICT) to facilitating rural developments such as rural finance, health, smaller enterprises, creating job opportunities, and towards the industrialization goal of GTP2. Therefore, mobile communication technology is one of the groundbreaking technology for the economic development of the country. In 2016, the contribution of total mobile revenue to local products was 1.78% and the contribution of total revenue to the local products was 2.2% [5].

Ethio telecom deployed Second Generation (2G) and Third Generation (3G) countrywide and LTE mobile network only in Addis Ababa since 2014.

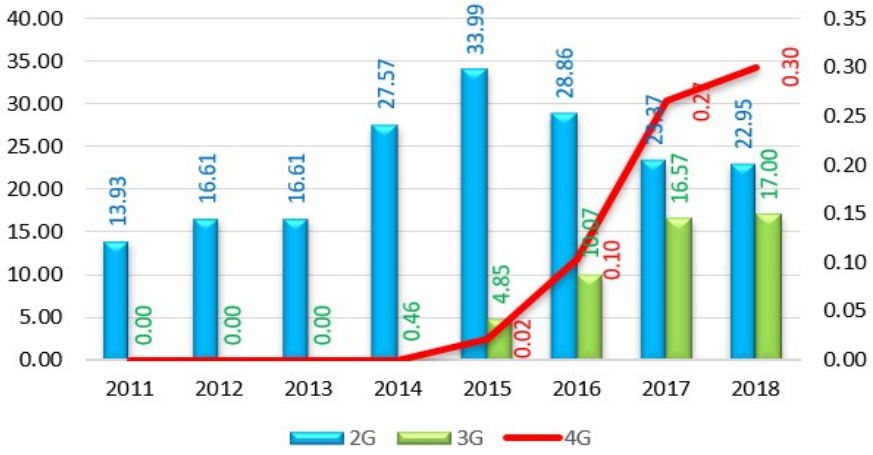


Figure 1.1: Ethio telecom customer growth trend in million

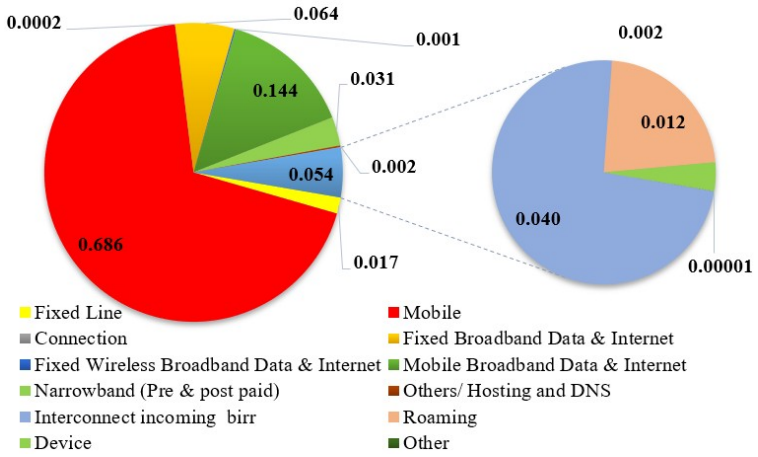


Figure 1.2: Ethio telecom revenue by service type

Despite the advantages of mobile communication network technology, there are hot security issues, due to inherent technological vulnerabilities that can be exploited by attackers to perform malicious activities. This can affect the normal and smooth operation of the networking ecosystem. As the use of personal computers moves to mobile devices with the emergence of Internet of Things (IoT), the scope and definition of mobile network security are changed from time to time [6]. The vulnerability of mobile network even become more critical as the technology is advanced towards ALL IP network. LTE is the first ALL IP network architecture that is managed to meet the growing

need for mobile data. As Ethio telecom adopted this technology, an attacker can inject threats to the network, which significantly affects the company's survival as well as the country. Even though the country believed that mobile communication technology is the enabler of facilitating the goal towards GTP2, there is less focus given to the security-related issues of mobile network technologies [3].

1.1 Statement of the Problem

Mobile network vulnerabilities will lead to threats that enable the attacker to misuse the network through an IP spoofing attack. These vulnerabilities might differ as the generation is evolved. A vulnerability in the ALL IP mobile network technologies (4G/LTE and above) becomes more important as the computer IP network threats and attacks can be applicable and flows in the mobile network. This attack results in a decrease in revenue and even can lead to economic and political crises for the company as well as the country.

Ethio telecom deployed LTE in 2014 and there are 300,000 subscribers as of 2018 [4]. LTE is becoming the next mobile technology that supposed to meet the need and demands of its customers. Thus, like any other telecom operators that implemented LTE mobile technology suffered from this attack, Ethio telecom will also be attacked through IP spoofing as it adopts and deployed LTE. Even though the mobile network is believed to plays a crucial role in achieving GTP2 and 85.9% of the company revenue is gained from mobile network services, Existing Ethio telecom mobile network security architecture is built for detecting and filtering of abnormal mobile packet traffics injected from external networks. However, an IP spoofing attack can arise from the internal User Equipments (UEs). Therefore, as the mobile network is considered as a backbone network and uses its own Evolved Packet Core (EPC) protocol called GTP, which is specific to it, there must be an advanced security measure.

1.2 Objective of the Study

1.2.1 General Objective

The general objective of this study is to analyze the performance of supervised machine learning-based IP spoofing attack detection mechanisms by examining GTP packets that are collected from real LTE mobile core networks.

1.2.2 Specific Objectives

The specific aims of the research are:

- Generate and collect both the benign and IP spoofed GTP packets using Ethio telecom real LTE network.
- Prepare the required LTE dataset.
- Compare three Machine Learning (ML) classifier algorithms namely; LR, KNN, and GNB based on classification metrics.
- Present the most better detection mechanism.

1.3 Research Questions

- How to prepare an LTE dataset to detect an IP spoofing attack?
- How to detect the label of this dataset given the selected features using machine learning?
- Which machine learning classifier fits best in the situation of IP spoofing attack detection?

When all these research questions are determined, detecting IP spoofing using machine learning becomes a much more manageable task. So, an effort is made to answer those questions along with this study.

1.4 Scope and Limitations of the Study

This work is considering ML-based IP spoofing attack detection on the LTE mobile network as the main subject. Amongst the various LTE attack origin points namely Internal, External, and Attacks from mobile devices, this study consider only on IP spoofing attacks that originated from mobile devices (UEs). This study considers and performs the following fundamental tasks to accomplish the desired outcome.

- Collect packets from the real LTE network for both the benign and IP spoofed GTP packets.
- In particular, this study is collecting and analyzing only uplink GTP User Plane (GTP-U) for user packets whereas it uses both the uplink and downlink GTP Control Plane Version 2 (GTPV2) packets for the control signal.
- Prepare the required LTE dataset.
- Evaluate the performance of LR, KNN and GNB Machine Learning classification algorithms.
- Recommend a better ML-based IP spoofing attack detection mechanism.

1.5 Contribution of the Study

The outcome of this research is intended to be of importance for both the industry as well as academia. Since the new knowledge gained from this research could be applied in various telecom industries as well as solving problems within the field of computer science and engineering. More specifically, the following list of contributions are considered in this research:

- It provides insight into the vulnerability of LTE network security infrastructure.
- It enables the telecom industries to the approach of machine learning-based IP spoofing attack detection mechanisms that initiated from UEs. .
- It offers an insight to researchers to study for a better approach.
- The labeled dataset can be used as an input for other related studies.

1.6 Related Works

Although research in [7] performs a study in the detection of IP spoofing attacks on a 3G network, there is a lack of studies on the detection of machine learning-based IP spoofing attacks in the LTE network. However, an effort is made to select more relevant articles and cited as a basis for answering the main research questions. Since, this study is prepared and uses the LTE dataset, reviewing literature that is working on state of the art dataset preparation is important besides reviewing articles working on the detection methodology. As a result, related studies that are cited on this study is categorized into two clusters; the first cluster is focusing on the importance of GTP protocol for the detection of mobile network anomalies like IP spoofing attack and the second focused on methods of LTE dataset preparation and ML-based mobile network anomalies detection approaches. These literature are being used to determine the appropriate techniques for selecting and evaluating ML classifiers.

1.6.1 Importance of GTP protocol for the Detection of IP Spoofing Attack on Mobile Network

GTP is an important LTE core network tunneling protocol, the tunnel created by GTP is the transport core of the LTE network between the user and the data service [8]. The tunnels provide the ability to manage each independent user and their QoS. This protocol is providing very important information on the detection process of IP spoofing attack detection as it is presented in [7]. During their study, they observed that security threats to the mobile network itself have not attracted much attention. Their study was focused on the GTP protocol to analyzed security threats likely to occur in mobile data networks, in particular, security threats using IP spoofing likely to affect the charging system and deteriorate service quality for users of mobile service, and proved in actual commercial service networks and proposed a countermeasure. The proposed countermeasure was based on analyzing GTP protocols. Similarly, in [9, 10, 11] it is explained that the information extracted from GTP protocol is crucial to detect GTP-in-GTP based DoS and Signaling DoS attack, and showed that this protocol is considered as a pivotal protocol in detecting most attacks injected into the mobile data network.

1.6.2 State of the Art Data Collection and Dataset Preparation for Mobile Network Anomalies Detection Using Machine Learning

Limei et al. [12] proposed a procedure on evaluation criterion for data collection and analysis to perform efficient security threats in the LTE or LTE/LTE-Advanced (LTE-A) mobile network based on the bases of previous studies. The main objective of the authors was, most of the existing work focuses on data collection and analysis for a certain type of LTE/LTE-A attacks, little work is done in terms of comprehensive data collection and analysis for security measurement in LTE/LTE-A. The authors analyzed a list of evaluation criteria to indicate the requirements of data collection and data analysis algorithms for attack detection. Finally, the authors come across an evaluation criterion of efficiency, privacy and resource consumption evaluation metrics for the data collection task and detection accuracy, time delay, computational complexity, handling capacity, robustness, additivity, and effect in-service performance evaluation metrics for data analysis task.

Research in [13, 14] presented a machine learning-based network anomalies detection methodology. In [13] the detection of Mobile Station Integrated Service Digital Number (MSISDN) based Over The Top (OTT) bypass fraud is explained. The authors evaluate the performance of three machine learning algorithms namely; AdaBoost + J48, RIPPER, and Support Vector Machine (SVM) in detecting MSISDN-based OTT voice call packets for OTT bypass fraud detection using their dataset, which was prepared on a controlled lab environment and set up. Based on the finding of the authors, AdaBoost with J48 was the best performer compared to RIPPER and SVM on both ten cross-fold and separate test data validation. Similarly, in [14] the effectiveness of anomaly-based IDSs in mobile malware detection using network traffic is conducted. The study assessed five classifiers, namely: the decision tree (J48), Bayes Network (BN), multi-layer perceptron (MLP), K-nearest Neighbors (KNN) and random forest. Based on the results, a multi-layer perceptron classifier produced a higher False Positive Rate (FPR) result with 5.17% compared to the random forest with 0.04%. The random forest also attained an elevated value for precision, recall, and f-measure of one. BN and random forest obtained 99.97%, which was the highest True Positive Rate (TPR) value in malware detection.

To the best of researches knowledge, there are no specific studies that work on machine learning-

based IP spoofing attack detection mechanisms in the LTE network. Even though the study in [7] highlights exactly the detection approach of IP spoofing attacks on mobile IP data networks like 3G technology, the approach was only discussed the way to detect this attack statistically, not a machine learning-based approach.

1.7 Methodology

The research workflow depicted in Figure 1.3 is divided into four phases that can highlight the complete procedures towards the required research outcome. The first phase is a literature review, which consists of studying previous related works, and the LTE standardization provided by Third Generation Partnership Project (3GPP). The second phase is concerned about data collection; the relevant data is collected from the real Ethio telecom LTE mobile core network using the following procedure:

- Install Huawei LMT (Local Maintenance Terminal) client application software and configured it to be able to login to the LTE mobile core management system.
- Connect to Ethio telecom Intranet and remotely logged in to Huawei LTE core network management system (UGW) via LMT.
- Using three LTE mobiles and one dongle/modem generates normal data traffic.
- Identify the IP addresses assigned to the LTE devices.
- Using the LTE dongle/modem, generated and send an IP spoofed Internet Control Message Protocol (ICMP) data traffic from a PC with a python script to five well-known servers. Then capture this specific traffic on the core network in .tmf file format which is Huawei proprietary.

The third phase consists of feature extraction and selection. The following procedure is used to accomplish the task of this phase:

- Convert .tmf packet to .pcap file format for the sake of easy preprocessing.
- Filter undesired packets using Wireshark (remove all packets except packets with GTP protocol).

- Prepare and store the required raw LTE dataset.

The collected packets are analyzed with the aid of Tshark/Wireshark/Python network packet analyzing software using a Linux based script to extract the required features. The fourth phase is concerned about evaluating the performance of three ML classifier algorithms to detect IP spoofed user packets.

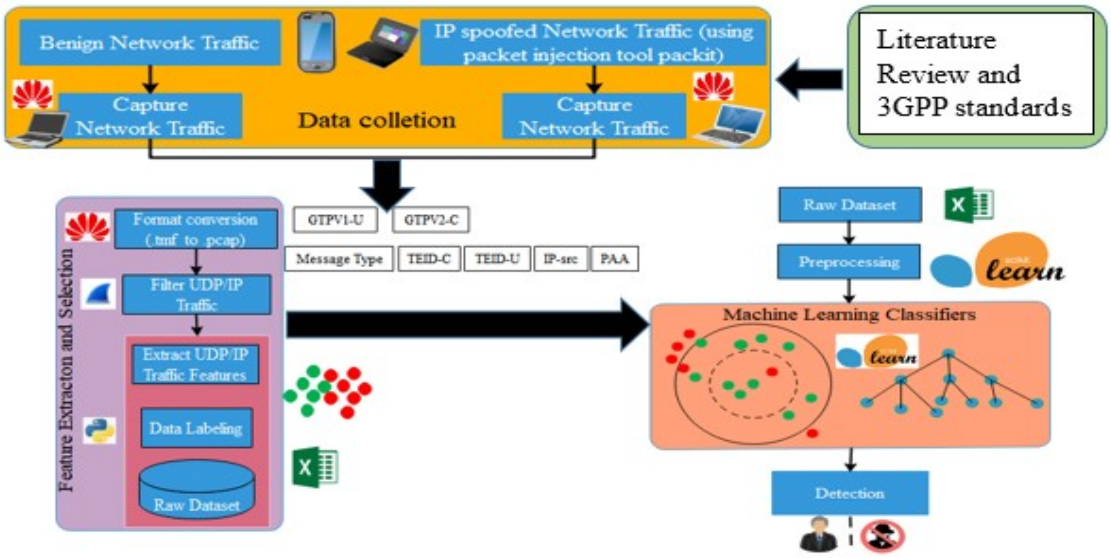


Figure 1.3: Methodology used to address IP spoofing attack in LTE

1.8 Document Structure

The chapters of this thesis reflect the different aspects of the research approach. Chapter 2 focuses on the technological background of the mobile network on the perspective of security concerns. Furthermore, the network, protocol, and security architecture are discussed and pinpoint the possible attack entry points on the LTE network. Chapter 3 outlines different machine learning techniques that are applied to this thesis. The topic of Chapter 4 presents the problem formulation of the study, furthermore, the tactic of feature engineering, experimental evaluation metrics, training, and testing dataset, as well as the research method are the point of discussion. The detailed of the results of the research are discussed in Chapter 5 and accompanied by an analysis of the collected LTE packet. In Chapter 6, general conclusions are drawn, based on the previously mentioned chapters, and possible future works are discussing.

Chapter 2

IP Spoofing Attack in LTE Network

LTE is the latest standard in mobile network technology and is supported by most smartphones. 3GPP developed the standard intending to increase downlink and uplink peak data rates, create scalable carrier bandwidths, and make a purely Internet Protocol (IP) based network architecture [15]. In addition to the significant functionality improvements, its security and privacy improved a lot compared to its predecessors. As of today, LTE is the fastest developing mobile network technology of all time and was commercially launched in more than 70% of the world [16]. In the next subsection, the LTE network architecture and protocol stacks are described in detail.

2.1 LTE Network Architecture Overview

The LTE network architecture is roughly divided into three parts: the access part called the Evolved Universal Terrestrial Radio Access Network (E-UTRAN), the core part called the EPC, and the UE. Furthermore, the E-UTRAN and EPC are divided into several network components, each playing an important role in the complete LTE network architecture [17, 18]. Figure 2.1 on page 13 illustrates the LTE network architecture, is shows the relationship between UE, E-UTRAN, EPC and their corresponding network components as well as the type of interface used between network elements. On the other hand, Table 2.1 and Table 2.2 summarized the function and interface of each network elements respectively.

Table 2.1: Description of the most common LTE network elements

Network Element	Description
UE	A UE connects to an eNB over the LTE-Uu interface.
eNB	AAn eNB provides users with the radio interfaces and performs Radio Resource Management (RMM) functions such as scheduler, eNB measurement configuration, admission control, mobility control, and Radio Bearer (RB) control.
MME	An MME is the main control entity for the E-UTRAN, it communicates with an HSS for user authentication and user profile download. Also, it handles mobility and EPS bearer management.
S-GW	An S-GW terminates the interface towards an E-UTRAN. It serves as the local mobility anchor point of data connection for Inter-eNB handover and Inter-3GPP handover.
P-GW	A P-GW provides a UE with access to a PDN by assigning an IP address from the address space of the PDN. The P-GW serves as the mobility anchor point for handover between 3GPP and non-3GPP.
HSS	An HSS is the central DB where the user profile is stored. It provides user authentication information and user profile to the MME.
PCRF	A PCRF is the policy and charging control. It makes a policy decision for QoS and charging rules to P-GW.
ePDG	The ePDG is responsible for interworking between the EPC and untrusted non-3GPP networks that require secure access, such as a WiFi, LTE metro, and femtocell access networks.

Table 2.2: Description of interfaces used in LTE network

Interface	Description
LTE-Uu	An interface for the control and user planes between a UE and an eNB.
X2	An interface for the control and user planes between two eNBs. It is used during handover and/or for Self-Organizing Network (SON) related function.
S1-U	An interface for the user plane between an E-UTRAN (eNB) and an S-GW. It provides a GTP tunnel per bearer.
S1-MME	An interface for the control plane between an E-UTRAN (eNB) and an MME.
S11	An interface for the control plane between an MME and an S-GW. It provides a GTP tunnel per user.
S5/S8	An interface defined between an S-GW and a P-GW for the control plane and user plane. The S5 interface provides a GTP tunnel per bearer for the user plane and GTP tunnel management (creation, modification, and deletion) per user for the control plane. For Inter-PLMN, an S8 interface is used instead.
S6a	An interface for the control plane between an HSS and an MME. It exchanges user subscription and authentication information.
S7	An interface to provide the transfer of QoS PCRF to PCEF in the P-GW.
S4	An interface to provides the user plane with related control and mobility support between SGSN and the S-GW.
Wn*	An interface for reference point between the untrusted Non-3GPP IP Access and the ePDG.
SGi	An interface for the control plane and user plane between a P-GW and a PDN.
S2a	An interface to provides the user plane with related control and mobility support between trusted non-3GPP IP access and the PGW.
S2a	An interface to provides the user plane with related control and mobility support between the ePDG and P-GW.

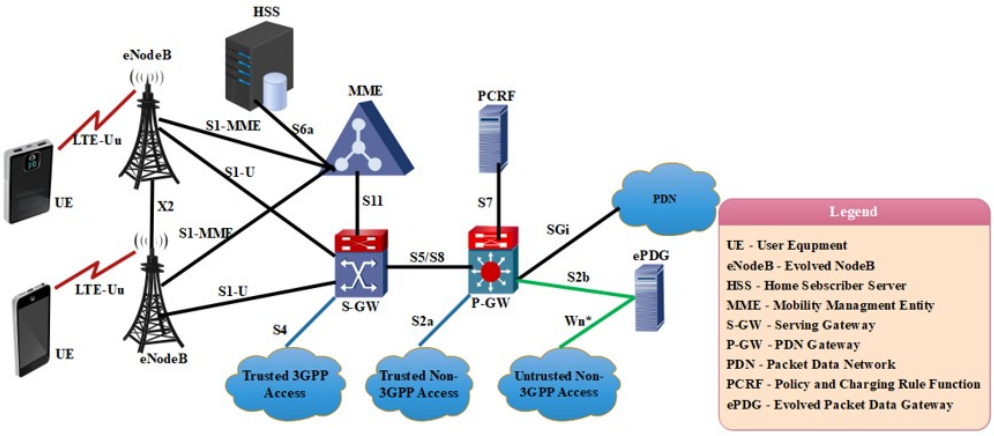


Figure 2.1: LTE network architecture

2.2 Protocol Architecture

LTE uses multiple protocols for the communication between the UE and the EPC. Each protocol performs operations on the user plane and/or the control plane. The control plane is used to route signaling traffic between the UE and the Mobility Management Entity (MME) while the user plane is used to carry user data between the UE and the S-GW [19].

2.2.1 User Plane

IP packets destined for a UE are encapsulated in an EPC-specific tunneling protocol and transported from the PDN Gateway (P-GW) through the Serving Gateway (S-GW) to the eNodeB, where the packet is transmitted to the UE over the air. As observed in Figure 2.2, the user data is encapsulated in the GTP during transportation from the eNodeB to the P-GW. The E-UTRAN user plane protocol stack is composed of the Packet Data Convergence Protocols (PDCPs), Radio Link Control (RLC) and Medium Access Control (MAC) and each protocol performs a specific function [17].

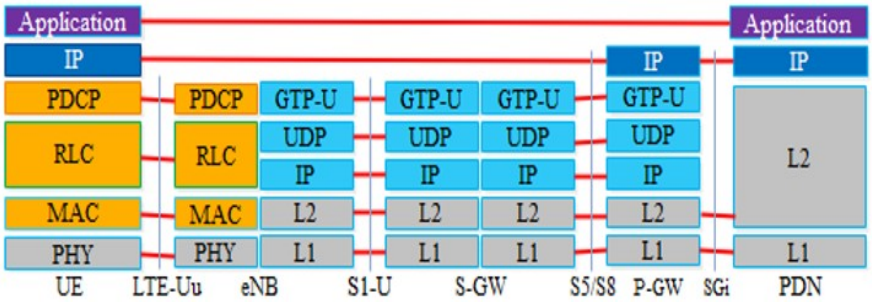


Figure 2.2: User plane protocol stack

2.2.2 Control Plane

The control plane includes functionality such as paging, broadcasting system information, UE measurement reporting, authentication, and EPC bearer management [17]. Figure 2.3 illustrates the protocol stack for the control plane between the UE and the EPC. Non Access Stratum (NAS) is the network layer communication between the UE and the MME, while the Access Stratum (AS) protocols used for communication between the UE and the eNodeB. The control plane contains the same protocols as in the user plane protocol stack in the E-UTRAN part except the control plane includes the Radio Resource Control (RRC) protocol [18].

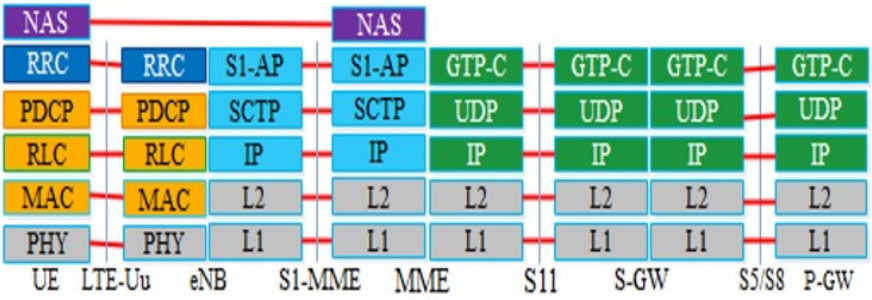


Figure 2.3: Control plane protocol stack

2.3 GTP Protocol Overview

Mobile network operators use GTP Protocol on various interfaces in roaming and RAN deployments, and within the packet core in 3G and 4G networks, to carry General Packet Radio Service (GPRS) packets. GTP allows mobile subscribers to use their phones (user equipment) to maintain

a connection to a PDN for Internet access while on the move [20]. 3GPP outlines eGTP of LTE in two separate specifications, GTPv2-C and GTPv1-U [8, 21, 22]. GTPv2-C is responsible for creating, maintaining, and deleting tunnels on S*x* interfaces. It is also responsible for forwarding relocation messages, and the creation of forwarding tunnels during inter-LTE handovers. GTPv1-U is used for transferring user data using GTP tunnels.

2.4 LTE Security

The security architecture of the EPC is mainly based on the Universal Mobile Telecommunication System (UMTS) architecture. However, new extensions and improvements have been implemented to increase the security of LTE. Consequently, LTE provides mutual authentication between the UE and the EPC making attacks such as Man in The Middle (MITM) difficult to perform and strong encryption algorithms make content hard to obtain. Although LTE has several solid built-in security mechanisms, still it is vulnerable to IP spoofing attacks [7].

To understand the security procedure in LTE, a general description of the EPC security concepts are discussed for an initial attach procedure of LTE user. Once the user power on the mobile device, it submits the subscriber identity to the EPC via the eNodeB; consequently, the MME queries the Home Subscriber Server (HSS) if the UE is allowed access to the network. Additionally, the MME requests the HSS for authentication data and initiates the authentication procedure if the UE identity is known. After completion of the authentication procedure, both the UE and the MME share the same master key K*ASME*. Subsequently, K*ASME* adopts further keys, used to ensure confidentiality and integrity protection of signaling messages and traffic between them and the UE [23]. K*ASME* derives three keys: K*NASenc*, K*NASint* and K*eNB*. K*NASenc*, is used for confidentiality protection, and K*NASint* is used for integrity protection. K*eNB* is used to ensure User Plane (UP) confidentiality between the eNodeB and the UE.

2.5 Origin of Security Threats in the LTE Network

Attack scenarios on the LTE core network can be launched at different sections of the network as depicted in Figure 2.4 [24]. Hence, all the security threats on an LTE network can be categorized into three types based on the origin of the attack. The first type of attack is originated from mobile

devices. The S1 interface connects and authenticates thousands of eNodeBs to the core network. An attacker can use this S1 interface to attack core network elements. The second type of attack is an internal attack, which is initiated by an insider/employee of the mobile network operator. In such an attack, a person who has access to back-haul network elements such as eNodeBs, aggregation routers, and repeaters will abuse his/her administrator's rights to perform these types of attacks. The third type of attack is originated at roaming networks. The S8 interface is used to interconnect external operator networks to support roaming customers. An attacker can use a device in the untrusted external operator network to attack the LTE core network through the S8 interface.

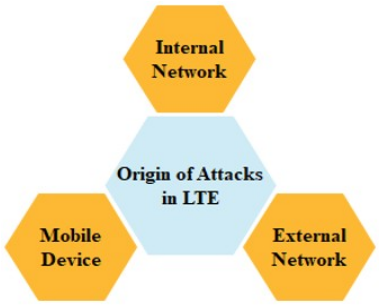


Figure 2.4: Origin points of attack in an LTE network

2.6 Source IP Spoofing attack in LTE

IP address spoofing is the forging of a source IP address field in IP packets to hide the identity of the sender or impersonate another computing system. Fundamentally, source IP spoofing is possible because Internet global routing is based on the destination IP address [25]. More precisely, an Internet router forwards packets from one interface to another looking up only the destination IP address. An application with sufficient privileges can modify the source IP address field of an IP packet to any syntactically correct value, and in most cases, the packet will be sent through the network interface and in many cases, will reach the destination.

Of course, an incorrect source IP address may delay normal operation of communications responses from the destination application or intermediary nodes and (e.g. ICMP responses) will not reach the sender. But attacks mounted using the spoofing technique do not rely on properly set up communication flows. On the contrary, they abuse this feature, directing traffic flow of responses to the target identified by the forged source IP address.

Chapter 3

Basics of Machine Learning

Machine Learning (ML) is automated learning with little or no human intervention. It involves programming computers so that they learn from the available inputs [26, 27]. The main purpose of machine learning is to explore and construct algorithms that can learn from the previous data and make predictions on new input data. Learning is the process of converting experience into expertise or knowledge.

The input to a learning algorithm is training data, representing experience, and the output is any expertise, which usually takes the form of another algorithm that can perform a task. The input data to a machine learning system can be numerical, textual, audio, visual, or multimedia. The corresponding output data of the system can be a floating-point number or an integer representing a category or a class. In this chapter, the three categories of machine learning and the common algorithms used in engineering and sciences are discussed. There are three categories of machine learning algorithms as listed below. However, the most commonly used ones are supervised and unsupervised learning [27].

- Supervised learning algorithm
- Unsupervised learning algorithm
- Reinforcement learning algorithm

3.1 Supervised Learning

This algorithm consists of a target or outcome or dependent variable which is predicted from a given set of predictor or independent variables. Using these sets of variables, generate a function that maps input variables to desired output variables. The training process continues until the model achieves a desired level of accuracy on the training data. Some examples of supervised learnings are Regression, Decision Tree, Random Forest, KNN, Logistic Regression, Naïve Bayes, etc.

Supervised learning is commonly used in real-world applications, such as face and speech recognition, products or movie recommendations, network anomalous detection, and sales forecasting. Supervised learning can be further classified into two types: Regression and Classification.

Regression: trains on and predicts a continuous-valued response.
Classification: attempts to find the appropriate class label, such as analyzing positive/negative opinions, for instance, benign and malignant network packets.

In supervised learning, learning data comes with description, labels, targets or desired outputs and the objective is to find a general rule that maps inputs to outputs. This kind of learning data is called labeled data. The learned rule is then used to label new data with unknown outputs.

Supervised learning involves building a machine learning model that is based on labeled samples. Supervised learning deals with learning a function from available training data. Here, a learning algorithm analyzes the training data and produces a derived function that can be used for mapping new observations. There are many supervised learning algorithms such as Logistic Regression, Neural networks, Support Vector Machine (SVM), KNN and Naive Bayes classifiers. Common examples of supervised learning include classifying e-mails into spam and not spam categories, labeling webpages based on their content, and voice recognition.

3.2 Unsupervised Learning

In this algorithm, there is no target or outcome or dependent variable to predict or estimate. It is used for clustering a given dataset into different groups, which is widely used for segmenting customers into different groups for specific intervention. K-means is the most common examples

of unsupervised learning. Unsupervised learning is used to detect anomalies, outliers, such as fraud or defective equipment, or to group customers with similar behaviors for a sales campaign. It is the opposite of supervised learning as there is no labeled data here.

When learning data contains only some indications without any description or labels, it is up to the coder or the algorithm to find the structure of the underlying data, to discover hidden patterns, or to determine how to describe the data. This kind of learning data is called unlabeled data. Suppose that there are many data points, and want to classify them into several groups. And may not exactly know what the criteria of classification would be. Therefore, the unsupervised learning algorithm tries to classify the given dataset into a certain number of groups in an optimum way.

Unsupervised learning algorithms are extremely powerful tools for analyzing data and identifying patterns and trends. They are most commonly used for clustering similar input into logical groups. Unsupervised learning algorithms include K-means, Random Forests, Hierarchical clustering and so on.

3.3 Reinforcement Learning

Using this algorithm, the machine is trained to make specific decisions. Here, the algorithm trains itself continually by using trial and error methods and feedback methods. This machine learns from past experiences and tries to capture the best possible knowledge to make accurate business decisions. Markov decision process is an example of reinforcement learning. Here learning data gives feedback so that the system adjusts to dynamic conditions to achieve a certain objective. The system evaluates its performance based on the feedback responses and reacts accordingly. The best-known instances include self-driving cars.

3.4 List of Common Machine Learning Algorithms

In this subsection, a brief of common and widely used machine learning algorithms that can be applied to almost any data problems are presented:

3.4.1 Linear Regression

Linear regression is used to estimate real values like the cost of houses, the number of calls, total sales, etc. based on a continuous variable(s) [26]. Here, the algorithm establishes a relationship between dependent and independent variables by fitting the best line. This line of best fit is known as the regression line and is represented by a linear equation.

3.4.2 Logistic Regression

Logistic regression is another technique borrowed by machine learning from statistics. It is the preferred method for binary classification problems, that is, problems with two class values [27]. It is a classification algorithm and not a regression algorithm as the name says. It is used to estimate discrete values or values like 0/1, Y/N, T/F based on the given set of independent variables. It predicts the probability of the occurrence of an event by fitting data.

3.4.3 Decision Tree Algorithm

A decision tree creates regression models and classification models just like a tree structure [28]. This tree works with the same concept as the if-then rule set that is mutually exclusive and exhaustive for classification. Rules are learned sequentially by applying the training data one at a time. Every time a rule is learned, the tuples that the rules handle are deleted. This process is repeated on the training set until a meeting termination condition is attained.

3.4.4 Support Vector Machines (SVM)

Support vector machines, also known as SVM, are well-known supervised classification algorithms that separate different categories of data [26, 27]. These vectors are classified by optimizing the line so that the closest point in each of the groups will be the farthest away from each other. This vector is by default linear and is also often visualized as being linear. However, the vector can also take a nonlinear form as well if the kernel type is changed from the default type of Gaussian or linear.

3.4.5 Naïve Bayes

The Naïve Bayes algorithm is a probabilistic classifier, which is driven by the Bayes theorem [28]. This is based on a simple assumption where attributes are conditionally independent. This assumption always reduces the computational cost. Although the assumption fails many times because the properties are dependent. Despite this, the Naïve Bayes has continued to work so well. This is a simple algorithm to implement and improve outcomes that are generated in most instances. It can be scaled into massive datasets because it assumes a linear time.

3.4.6 KNN (K-Nearest Neighbors)

K-Nearest Neighbors, KNN for short, is a supervised learning algorithm specialized in classification [26, 27]. It is a simple algorithm, which stores all available cases and classifies new cases by a majority vote of its k neighbors. The case is assigned to the class is the most common among its K nearest neighbors measured by a distance function. These distance functions can be Euclidean. If K = 1, then the case is simply assigned to the class of its nearest neighbor. Hence, choosing K turns out to be a challenge while performing KNN modeling.

The algorithm looks at different centroids and compares distance using some sort of function (usually Euclidean), then analyzes those results and assigns each point to the group so that it is optimized to be placed with all the closest points to it. KNN can be used for both classification and regression problems. However, it is more widely used in classification problems in the industry. KNN can easily be mapped to real lives.

3.4.7 Random Forest

Random Forest is a popular supervised ensemble-learning algorithm [26]. Ensemble means that it takes a bunch of 'weak learners' and has them work together to form one strong predictor. In this case, the weak learners are all randomly implemented decision trees that are brought together to form a strong predictor called a random forest. The goal of ensemble methods is to combine the predictions of several base estimators built with a given learning algorithm to improve generalizability/robustness over a single estimator.

3.4.8 K-Means

K-Means clustering is based on an unsupervised ML approach [26, 27]. This method aims to discover clusters in the data, and k refers to the number of clusters to be generated by the algorithm. The method is implemented by iteratively allocating each data point to one of the k clusters according to the given features. Each cluster will contain samples with similar features. The k-means algorithm applies iterative refinement to generate an ultimate result. The inputs of the algorithm are the number of clusters (k) and dataset, which contains a set of features for each sample in the dataset. Firstly, the k centroids are estimated, and then each sample is assigned to its closest cluster centroid according to the squared Euclidean distance. Secondly, after all the data samples are assigned to a specific cluster, the cluster centroids are recalculated by computing the mean of all samples assigned to that cluster. The algorithm iterates these steps until no sample that can modify the clusters exists.

Chapter 4

Problem Formulation

This chapter focused on the experiment that is analyzed IP spoofing attack detection in the LTE network. The discussion starts with the dataset description that is used then goes through packet generation and collection procedures for both the benign and IP spoofed packets. The dataset collection and testing environment that used in this research are conducted in the Ethio telecom real LTE network. Therefore, a method such as data cleaning, features-extraction, and features engineering tactics are applied and discussed to ensure the dataset global standard. Classification algorithms selection, experimental validation approach also defined and elaborated detail in this chapter based on the overall research workflow depicted in Figure 4.1.

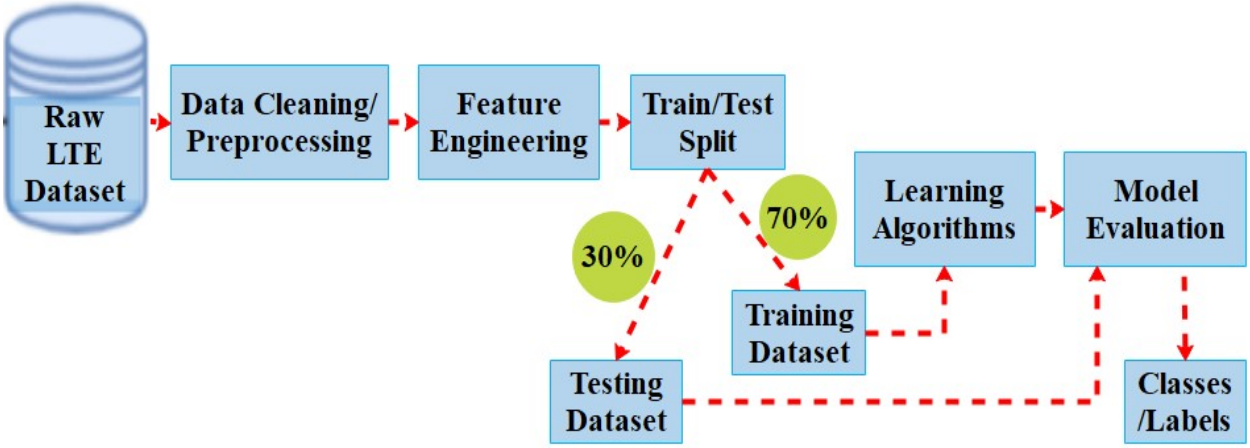


Figure 4.1: Overall research workflow

4.1 Dataset Description

The research is conducted based on the actual real LTE mobile data network. Ethio telecom provides four service numbers to use for this research purpose. Until the research finalized, the company is allowed to access the mobile core and collect packets associated only to those of approved service number/MSISDN. At the same time, this research is accompanied by a promise to act as an ethical attacker, it means that the study must not affect the normal LTE network operation. Hence, not overwhelmed the network by generating a huge amount of traffic, in particular during peak hours. To accomplish the task of this study, following network setup and procedures as illustrated in Figure 4.2 is used to generate and capture both the required packets. The materials used to collect both the benign and IP spoofed packets, as well as the implementation of machine-learning, are summarized and shown in Table 4.1.

Table 4.1: Device and tools used in this study

Devices	Quantities	Application
HP laptop (core i7)	1	Packet generation, capturing and ML implementation
Toshiba laptop (AMD)	1	Packet generation and capturing
Huawei LTE smart phones	3	Normal packet generation
Huawei LTE USB modem	1	IP spoofed packet generation
Tools		
Wireshark		Manual packet filtering and analysis
Tishark		Automatic packet filtering
python		IP spoofed packet generation
Sicikt learn		ML classification algorithm implementation
Huawei LMT		Packet capturing

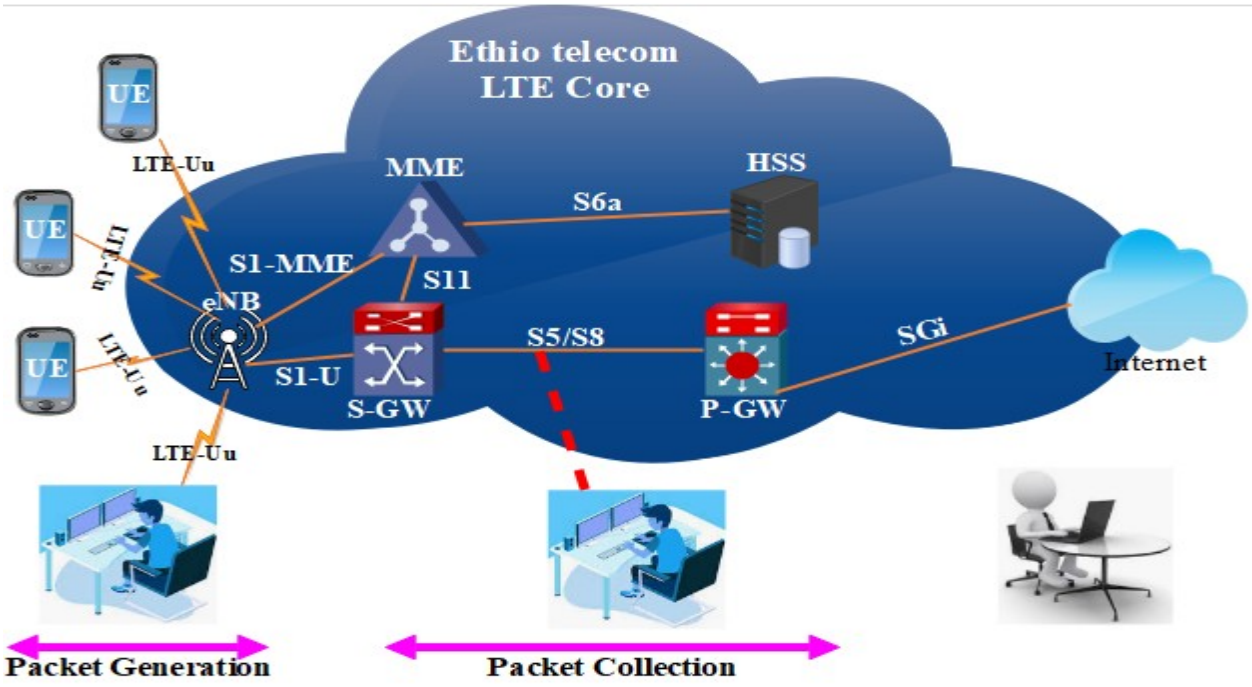


Figure 4.2: Packet generation and collection setup

4.2 Benign Packet Data Generation and Collection

The benign user packets are generated as any legitimate LTE subscribers are generating traffics. In other words, the normal data traffics are generated using the following procedures:

- Switch on the LTE mobile devices.
- Search and locked the mobile devices to access only the LTE network.
- Turn on the mobile data.
- Browse and generate data traffics.

When the user generates normal traffics, the associated packets are started to capture in the mobile core. This including the signaling traffic at the same time. The actual process followed to capture the user-specific data traffics is listed below:

- Install Huawei UGW client software (LMT) on a Pcs.
- Connect to Ethio telecom Intranet.

- Open and configure the LMT.
- Remotely log in to selected UGW by providing the username and password.
- Capture user-specific packet by filtering based on their MSISDN/IMSI/IME.
- Save the captured user packets, which is in .tmf file format(Huawei proprietary).

4.3 IP Spoofed Packet Data Generation and Collection

The IP spoofed user packets are generated using an LTE modem on Linux using a python script. The detailed procedure is explaining as follows:

- Connect a stick mode LTE modem on to a PC, if the modem is a Hi-link mode change to stick mode first. Unfortunately, the modem used in this study is Hi-link mode.
- Configure Linux so that it detects the LTE modem as a modem, by default Linux detects the LTE modem as a mass storage device.
- Identify the allocated IP address of the victim.
- Run a python script that generates and sends an IP spoofed ICMP packets to a well-known five servers (Google, Facebook, 4shared, YouTube and Yahoo).

Similarly, capture the IP spoofed packets on the real LTE core network with the same procedures as followed in the case of capturing the normal packet traffics in section 4.2. Based on the network setup and procedures discussed above a total of one hundred three thousand seven hundred six (103, 706) raw LTE packets are collected. The details of raw LTE packet data collection statistics are discussed in the next section.

4.4 Data cleaning

The collected LTE packets are usually having lots of raw data, which is not relevant for this study. Hence, clean the raw data before going to the next step is important. To start data cleaning, understand what the data is, and what want to achieve is vital. Without that understanding, have no basis from which to make decisions about what data is relevant when clean and prepare the required

dataset. The packets are in .tmf format that is not easy to handle any data filtering and analysis. Therefore, first converting the packets from .tmf to .pcap file format using Huawei proprietary tool TranExpert is performing. Then based on an extensive literature reviewing, reading 3GPP LTE standards and a detailed understanding of the normal and IP spoofing attack flow nature in the LTE network, the packets are mainly filtered upon GTP and GTPV2 protocol. The number of packets generated and collected statistics are briefly presented in Table 4.2.

Table 4.2: Packet collection statistics

Number of Raw Packets Collected		Total	Number of Filtered Packets		Total LTE Raw Dataset
Normal	46,914	103,706	Normal	18,441	27,840
IP Spoofed	56,792		IP Spoofed	9,399	

4.5 Feature Extraction

As this study is focusing on a machine learning-based detection system, feature extraction is important. Machine learning algorithms give the best results only when providing it the best possible features that structured the underlying form of the problem, which was trying to address. Often these features have to be manually created by spending a lot of time with actual raw data and trying to understand its relationship with all other data that was collected to addressed this problem [27]. The purpose of feature extraction is to decide which of the initial (possibly large number) features to include in the required LTE dataset and which feature to ignore. The collected packets have more than 100 features, identifying the most relevant features directly amongst those of all captured features are essential.

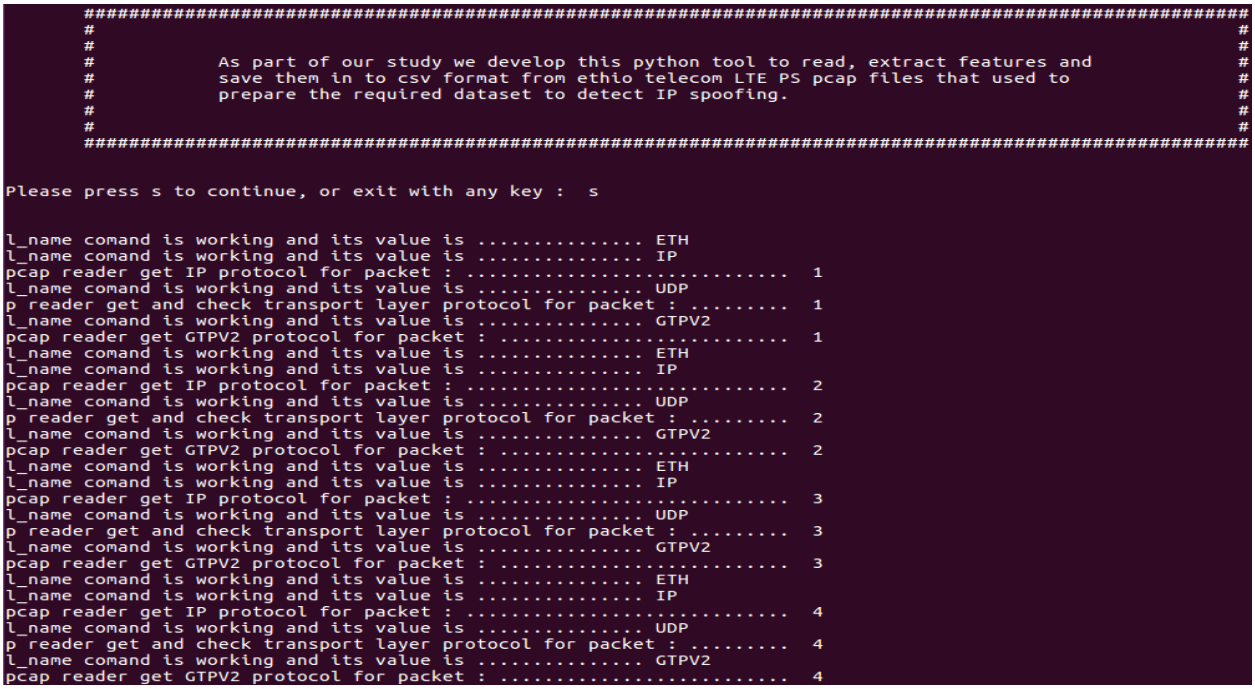


Figure 4.3: Python tool that automatically extract the required features from packets and generate a row dataset in CSV

The basic principle in feature extraction is fewer features make the IP spoofing detection system faster and the more features make the IP spoofing detection system more accurate. In general, the performance of the IP spoofing detection system depends on the features extracted. The packets contained features that are not important for this particular study, so reducing the number of features improves the overall performance and computational time of the detection system. The method of feature extraction followed in this research is grounded on an extensive literature reviewing, reading 3GPP LTE standards and a drill-down understanding of the flow of normal and IP spoofed attack natures in the LTE network as described in the data cleaning section 4.4. To automate the feature extraction process, and as part of this study, a python tool is developed that reads the collected packets in pcap format and extract the required features. As Figure 4.3 illustrates, the tool automatically processes the packets and extracted the required features amongst hundreds of features and output the result in CSV format as shown in Table 4.4. Upon the feature extraction methodology, the tool is extracted fifteen (15) features that can able to captured the characteristics of an IP spoofed LTE packets. Table 4.3 list and explain those selected features in which they can capture the possible dynamics of the attacking nature assumed in this study.

Table 4.3: The extracted features and their description

Features	Descriptions
highest_layer	Capture either the packet is control siginal or user data
transport_layer	Capture the type of transport layer protocol
message_type	Capture the type of LTE control messages
message_length	Capture the length of the data
src_ip	Capture the IP address of the user equipment (UE)
dst_ip	Capture the destination IP address of the user packets
src_port	Capture the source port of the user equipment
dst_port	Capture the destination port of the UE packets
seq_num	Capture the sequence number of the control packets
teid	Capture the UL tunnel ID of the user packets
f_teid_gre_key	Capture the assigned UL tunnel ID by the network
PAA	Capture the assigned IP address to UE by the network
UE_msisdn	Capture the MSISDN number of the user
UE_imsi	Capture the IMSI number of the user SIM card
UE_mei	Capture IMEI number of the user equipment (UE)

Table 4.4: Sample of the required raw LTE dataset in csv format

highest layer	transport layer	message type	message length	src ip	dst ip	src port	dst port	seq num	teid	f_teid gre_key	PAA	UE msisdn	UE imsi	UE mei	class
GTP	UDP	255	28	10.57. 220.146	31.13. 90.36	2152	2152	1	66226 2558	1	1	1	1	1	IP spoofed
GTP	UDP	255	28	10.57. 220.146	31.13. 90.36	2152	2152	1	66226 2558	1	1	1	1	1	IP spoofed
GTP	UDP	255	28	10.57. 220.146	31.13. 90.36	2152	2152	1	66226 2558	1	1	1	1	1	IP spoofed
GTP	UDP	255	28	10.57. 220.146	31.13. 90.36	2152	2152	1	66226 2558	1	1	1	1	1	IP spoofed
GTP	UDP	18	76	[redacted]	[redacted]	2123	2123	1	68136 0651	1	1	1	1	1	Normal
GTP	UDP	19	48	[redacted]	[redacted]	2123	2123	1	31410 24304	1	1	1	1	1	Normal
GTP V2	UDP	32	193	[redacted]	10.202. 0.1	2123	2123	1595 862	0	25146 17904	10.57. 220.146	1	63601 30000 00000	86114 10000 00000	Normal
GTP V2	UDP	32	193	[redacted]	10.202. 0.1	2123	2123	1595 786	0	25146 17904	10.57. 125.199	1	63601 30000 00000	86114 10000 00000	Normal
GTP V2	UDP	33	134	10.202. 0.1	[redacted]	2123	2123	1595 786	25146 17904	68109 8507	1	1	1	1	Normal

4.6 Feature Engineering

Raw data rarely comes in the form and shape that is necessary for the optimal performance of a learning algorithm. Thus, the method of preprocessing is the most crucial step in any machine learning application [29]. The detection quality of any machine learning algorithm depends predominantly on the quality of input is being passed. The process of creating appropriate data features by applying the business context is called feature engineering, and it is one of the most important aspects of building efficient machine learning systems [27]. So, understand the fundamental of feature engineering before proceeded to different types of machine learning algorithms is key. Some of the common practices that are part of feature engineering and adopted in this research are discussed in the following subsection.

4.6.1 Dealing with Missing Data

Missed data can mislead or create problem during analyzing the data. To avoid any such issues, it is needed to assign a value to missed data. Fortunately, for this particular study, the required LTE dataset is engineered manually, not suffered on such issues as ensured in Figure 4.4. A feature name with zero value means there is no missed value, whereas one value means there is a missed value that needs to avoid it.

```
Out[10]: highest_layer      0
         transport_layer    0
         message_type       0
         message_length     0
         src_ip             0
         dst_ip             0
         src_port           0
         dst_port           0
         teid               0
         seq_num            0
         PAA                0
         f_teid_gre_key     0
         UE_msisdn          0
         UE_imsi            0
         UE_mei             0
         class              0
         dtype: int64
```

Figure 4.4: Confirming that there is no missing data in our LTE dataset

4.6.2 Handling Text Data

Most machine learning algorithms are designed to work well with numerical variables rather than nominal variables. So, nominal variables in their original form of text description can't be directly used for model building. This study used Microsoft Excel to handle the conversion of text variables to numerical variables keeping the integrity of the data is not affected. That means applying the same approach for each observation. As seen in Table 4.5, the encoding is done based on the application of the protocol and numerical representation from their original collected packet.

Table 4.5: Handling of nominal features

Features	Encoding
GTP	1
GTPV2	2
UDP	17

4.6.3 Encoding Class Labels

In supervised learning, the dataset mostly comes across a variety of labels, which often be in the form of words. Because many times labels need to be in a readable form. Hence, the training data is usually labeled with words. In this particular study, there are word labels namely: Normal and IP spoofed. Table 4.6 refers to the mapping of word labels into numbers so that the algorithms can understand how to work on them [30]. Many machine-learning libraries required that class labels are encoded as integer values. Although most estimators for classification in scikit-learn convert class labels to integers internally, there is no universal rule to encoding the class labels. However, it is considered as a good practice to provide class labels as an integer array to avoid technical problems. To encode the class labels, the label encoding module in the scikit-learn library is used. Remember that class labels are not ordinal, and it does not matter which integer number is assigned to a particular word label. Thus, enumerated the class labels starting at zero.

Table 4.6: Label encoding

classes	Encoding
Normal	0
IP spoofed	1

4.6.4 Feature Selection

Feature selection is the process where selects those features, which contribute most to the prediction variable. Once feature extraction performs, feature selection is one of the core concepts in machine learning that extremely impacts the performance of the classification model. The data features that are used to train the machine learning models have a huge influence on performance can achieve. Irrelevant features can negatively affect model performance. Having irrelevant features in the dataset can decrease the accuracy of the models and make the model learn based on irrelevant features.

A feature that can have a good correlation to the target variable, which examined by a certain classifier may or may not be has a good correlation when it examined with some other classifier. Hence, for feasible performance evaluation of classifiers, the classifier's independent feature selection method is being used in this research called Pearson's correlation. Correlation states how the features are related to each other or the target variable. Correlation can be positive (increase in one value of the independent feature, increases the value of the target variable) or negative (increase in one value of the independent feature, decreases the value of the target variable) or it can be zero (there is no correlation between the variables). To visualize the correlation between the dependent/target variables with the independent variables, a bar chart is used to easily understand the Pearson's correlation of the variables as depicted in Figure 4.5. Based on Pearson's correlation algorithm, the independent variable transport_layer does not correlate with the dependent/target variable. This indicates that this variable is not relevant and hence drop it as considering as a feature on the subsequent classification model building process. Thus, a total of fourteen (14) feature being used in this study.

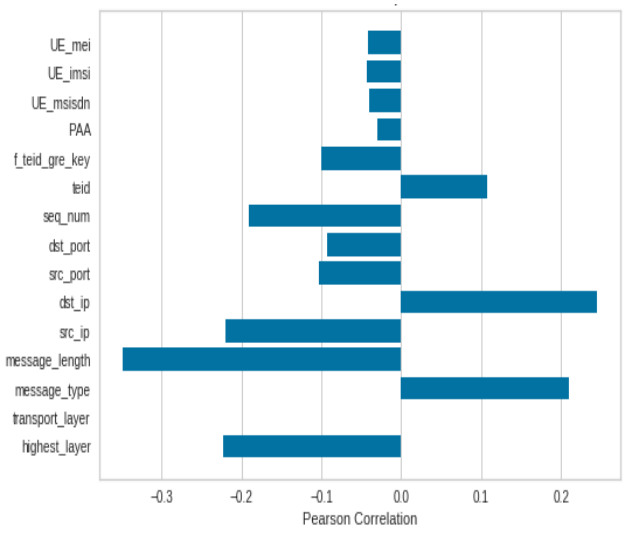


Figure 4.5: Pearson's correlation of independent variables over the dependent variables

4.6.5 Feature Scaling

Feature scaling is a crucial step in the feature engineering process that can easily be forgotten. However, the majority of machine learning and optimization algorithms behave much better if features are on the same scale. There are two common approaches to bring different features onto the same scale: normalization and standardization. Most often, normalization refers to the process of scaling individual samples to have unit norm [31], Whereas, standardization refers to the process of making the data to have Gaussian distribution with zero mean and unit variance. For this research, standardization of the dataset scaling approach is used. Table 4.7 shows the last five observations once standardization is being applied to the dataset. The mathematical formulation of data standardization implemented in scikit-learn is express as:

$$X_{std}^{i} = \frac{X^{i} - \mu_x}{\sigma_x} \tag{4.1}$$

Table 4.7: The last five sample LTE dataset after applying feature selection and standardization

hlayer	mtype	mlength	srcip	dstip	src port	dst port	seq num	teid	f_teid gre_key	PAA	UE msisdn	UE imsi	UE mei
1.0	0.67 0588	0.01 3194	0.00 0005	0.09 7024	0.03 2400	0.03 2400	1.2633 16E-01	0.84 6077	3.7628 64E-10	9.4061 42E-10	3.9687 74E-12	1.5722 78E-15	1.1545 57E-15
1.0	0.13 7255	0.03 2639	0.00 0005	0.09 7024	0.03 2400	0.03 2400	1.2652 12E-01	0.84 6077	8.1723 40E-02	9.4061 42E-10	3.9687 74E-12	1.5722 78E-15	1.1545 57E-15
1.0	0.13 7255	0.03 2639	0.00 0005	0.09 7024	0.03 2400	0.03 2400	1.2652 56E-01	0.84 6077	8.1711 37E-02	9.4061 42E-10	3.9687 74E-12	1.5722 78E-15	1.1545 57E-15
0.5	1.00 0000	0.04 0972	0.08 2786	0.00 5134	0.03 2842	0.03 2842	1.0015 59E-08	0.0 0001	3.7628 64E-10	9.4061 42E-10	3.9687 74E-12	1.5722 78E-15	1.1545 57E-15
1.0	0.67 0588	0.01 3194	0.00 0005	0.09 7024	0.03 2400	0.03 2400	1.2652 79E-07	0.84 6077	3.7628 64E-10	9.4061 42E-10	3.9687 74E-12	1.5722 78E-15	1.1545 57E-15

4.7 Classification Algorithms

The dataset used in this study is labeled manually, it provides the ability to use supervised learning classification algorithms. As Table 4.8 shows, three classifiers are selected, namely: LR, KNN, and GNB. The selection criterion for those algorithms are basically upon the characteristics of the LTE dataset as well as based on their application, computational complexity, and execution time of the algorithms.

Table 4.8: Selected supervised classification algorithms

Algo.	Pros	Cons	Applications
LR	Fast to train and detect Easy to implement Easy to understand	Cannot be applied on non-linear classification problems Proper selection of features is required	Detection of network anomalies
KNN	Simple to understand Easy to implement	Computational expensive Proper selection of features is required K should be wisely selected Proper scaling of feature is required	Detection of network intrusion and anomalies
GNB	Very fast to train and detect Easy to understand Robust to the parameters Consider as a baseline model Works well for small dataset	Expect all features to be independent Proper selection of features is required	Detection of network intrusion

4.8 Experimental Validation Method

This study applies a train-test split validation approach to evaluate and validate the performance of the selected classification algorithms. Training dataset and test dataset are two important concepts in machine learning. The ratio of the training to test dataset this research being followed is 70/30 as shown in Figure 4.1 on page 26. This means, 70% of the LTE dataset is used for training and building of the models whereas the reset 30% is used for testing the model.

4.8.1 Training Dataset

The observations in the training set create the experience that the algorithm used to learn [28]. In supervised learning problems, each observation consists of an observed target variable (Normal/IP spoofed) and fourteen (14) observed input variables/features. Seventy percent (70%) of the total

dataset, which is nineteen thousand four hundred eighty-eight (19,488) of the observations are used as a training set.

4.8.2 Testing Dataset

The testing set is a set of observations that are not used or included in the training set, which is used to evaluate and validate the performance of the model using the metrics described in the next section [28]. Thirty percent (30%) of the total dataset, which is eight thousand three hundred fifty-two (8,352) of the observations are used as a testing set and/or validation. No observations from the training set are included in the testing set. If the testing set does contain observation from the training set, it is difficult to assess whether the algorithm has learned to generalize from the training set.

4.9 Evaluation Metrics

The metrics that are mainly used to evaluate the performance of classifiers are presented in [32] and the one used in this research is explained here in the following subsection.

4.9.1 Basic terminology

True Positive (TP): the classifier correctly predicted that the packet does have IP spoofed.
True Negative (TN): the classifier correctly predicted that the packet does not have IP spoofed.
False Positive (FP): the classifier incorrectly predicted that the packet does have IP spoofed.
True Positive (TP): the classifier incorrectly predicted that the packet does not have IP spoofed.

To evaluate the general performance of different classifiers, different metrics are used. These metrics are calculated from the confusion matrix, which shows the predicted and actual classifications. Since the numbers of a class are two in this research (either positive or negative), the size of the confusion matrix is 2x2 as depicted in Figure 4.6. The confusion matrix gives a complete picture of how the classifiers are performing. It also allows computing various classification metrics, which can guide the model selection process.

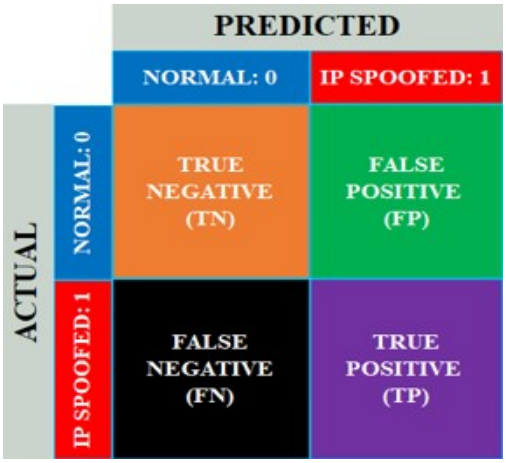


Figure 4.6: Confusion matrix for binary classification problems

4.9.2 Metrics Computed from a Confusion Matrix

The confusion matrix is useful in helping to understand the performance of the classifiers. Many metrics can be calculated from the confusion matrix and those can be directly used to choose between models. In this subsection, a few popular metrics that are choosing to consider in this research are presented.

Classification Accuracy: classification accuracy measures the performance of the model how often the classifier is correct in detecting the classes of newly observed data. Classification accuracy is calculated from the confusion matrix as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.2}$$

Classification Error: classification error measures the performance of the model how often the classifier is incorrect in detecting the classes of newly observed data. Classification error is also known as "Misclassification Rate", and calculated as:

$$Error = 1 - Accuracy \quad OR \quad \frac{FP + FN}{TP + TN + FP + FN} \tag{4.3}$$

Sensitivity: when the actual value is positive, sensitivity measures the performance of the model

how often the prediction is correct in detecting IP spoofed attack from out of sample or newly observed data. Sensitivity also has known as True Positive Rate or Recall. It can also be defined as how sensitive is the classifier to detecting IP spoofed attack. Based on the confusion matrix, sensitivity is calculated as:

$$Sensitivity(TPR/Recall) = \frac{TP}{TP + FN} \tag{4.4}$$

Specificity: when the actual value is negative, specificity measures the performance of the model how often the prediction is correct in detecting normal packets from newly observed data. Specificity can also be defined as how specific or selective the classifier is in the prediction of negative instances. Just like sensitivity, specificity wants to maximize. In terms of the confusion matrix, specificity is calculated as:

$$Specificity = \frac{TN}{TN + FP} \tag{4.5}$$

For both sensitivity and specificity, the best possible value is one, so that can describe the classifier as highly specific and highly sensitive.

False Positive Rate (FPR): when the actual value is negative, FPR measures the performance of the model how often the prediction is incorrect in detecting normal packets from newly observed data. In terms of the confusion matrix, FPR is calculated as:

$$FalsePositiveRate(FPR) = \frac{FP}{TN + FP} \tag{4.6}$$

Precision:when a positive value is predicted, precision measures the performance of the model how often the prediction is correct in predicting IP spoofed attacks from newly observed data. It can be thought of precision as describing how precise the classifier is when predicting a positive instance. In terms of the confusion matrix, precision is calculated as:

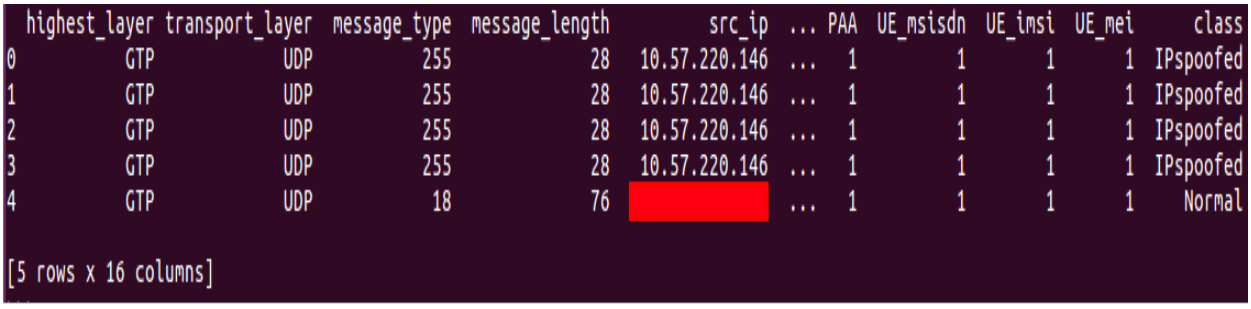
$$Precision = \frac{TP}{TP + FP} \tag{4.7}$$

Even though many other metrics can be calculated from confusion matrix such as F1 score, Mathews correlation coefficient, etc., this study is considered the metrics discussed in this section only

Chapter 5

Results and Discussions

The objective behind this section is to evaluate which of the three classification algorithms have the best performance in the detection of IP spoofing attacks on the LTE mobile network. A packet generated from the actual Ethio telecom LTE network is used for this section onward. This consisting of 14 column variables and 27,840 observations that are carried out and collected from the real LTE mobile data network. As already discussed in section 4.6, the dataset was engineered so that it handled by machine learning algorithms in scikit-learn. Thus, load the LTE dataset into scikit-learn using pandas read_csv function. Then, print the first five data frames using the built-in head method in scikit-learn as illustrated in Figure 5.1 to visualize that the proper loading of the dataset into scikit-learn. As illustrated in this Figure, each row represented a single LTE packet and the class column indicates the label of the packets (if the label is normal, it is a normal packet or else if it is IP spoofed, it is an IP spoofed packet).



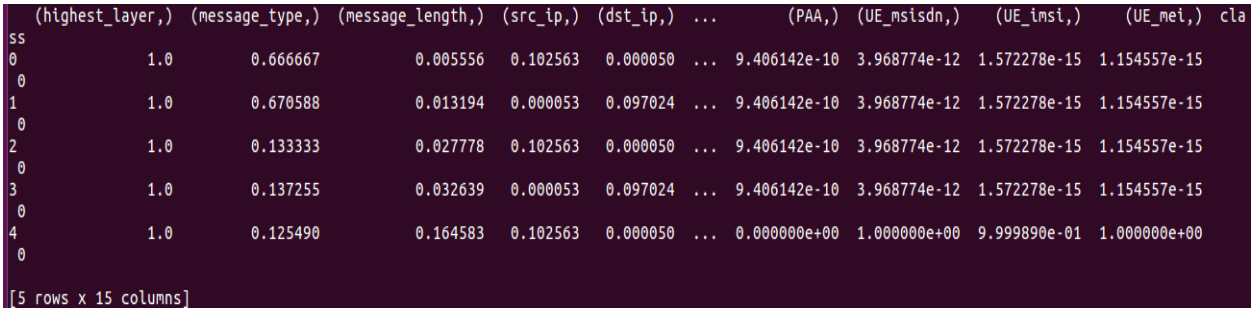
	highest_layer	transport_layer	message_type	message_length	src_ip	...	PAA	UE_msisdn	UE_imsi	UE_mei	class
0	GTP	UDP	255	28	10.57.220.146	...	1	1	1	1	IPspoofed
1	GTP	UDP	255	28	10.57.220.146	...	1	1	1	1	IPspoofed
2	GTP	UDP	255	28	10.57.220.146	...	1	1	1	1	IPspoofed
3	GTP	UDP	255	28	10.57.220.146	...	1	1	1	1	IPspoofed
4	GTP	UDP	18	76	[redacted]	...	1	1	1	1	Normal

[5 rows x 16 columns]

Figure 5.1: Our first five datasets loaded in scikit learn

Once a proper loading of the dataset into scikit-learn is confirmed, a feature engineering method is applied to come across the final LTE dataset that is used in this research as depicted in Figure5.2

the first five datasets. Then, start to use the machine learning processes by defining a variable of feature matrix **X** and the response variable vector **y** to proceed and use on the rest of the evaluation process in this chapter.



	(highest_layer,)	(message_type,)	(message_length,)	(src_ip,)	(dst_ip,)	...	(PAA,)	(UE_msisdn,)	(UE_imsi,)	(UE_mei,)	cla
ss											
0	1.0	0.666667	0.005556	0.102563	0.000050	...	9.406142e-10	3.968774e-12	1.572278e-15	1.154557e-15	
0											
1	1.0	0.670588	0.013194	0.000053	0.097024	...	9.406142e-10	3.968774e-12	1.572278e-15	1.154557e-15	
0											
2	1.0	0.133333	0.027778	0.102563	0.000050	...	9.406142e-10	3.968774e-12	1.572278e-15	1.154557e-15	
0											
3	1.0	0.137255	0.032639	0.000053	0.097024	...	9.406142e-10	3.968774e-12	1.572278e-15	1.154557e-15	
0											
4	1.0	0.125490	0.164583	0.102563	0.000050	...	0.000000e+00	1.000000e+00	9.999890e-01	1.000000e+00	
0											

[5 rows x 15 columns]

Figure 5.2: Sample of actual dataset used in this research

Using panda's data frame, extract out all the features and stores them into the variable **X**. Similarly, extract the classes of each packet and stored them into the variable **y**. Then, the selected classifiers are configured in scikit-learn with the parameters shows in table 5.1 for optimal performance of the classifiers rather than using the default parameters.

Table 5.1: Classifier algorithms parameter configuration

Algorithm	Configuration
LR	random_state=1 solver='liblinear'
KNN	n_neighbors=18
GNB	NA

The parameter selection and configuration of the classifiers are grounded on considering the nature of the LTE dataset. The problem that is going to address in this study has a linear character, thus a linear solver is used for the LR classifier. The most important and challenging parameter for the KNN classifier is finding the optimal value for *K*. Because, *K* directly depends on the nature of the dataset used. For this particular study, K=18 is observed as an optimal value as depicted in Figure 5.3.

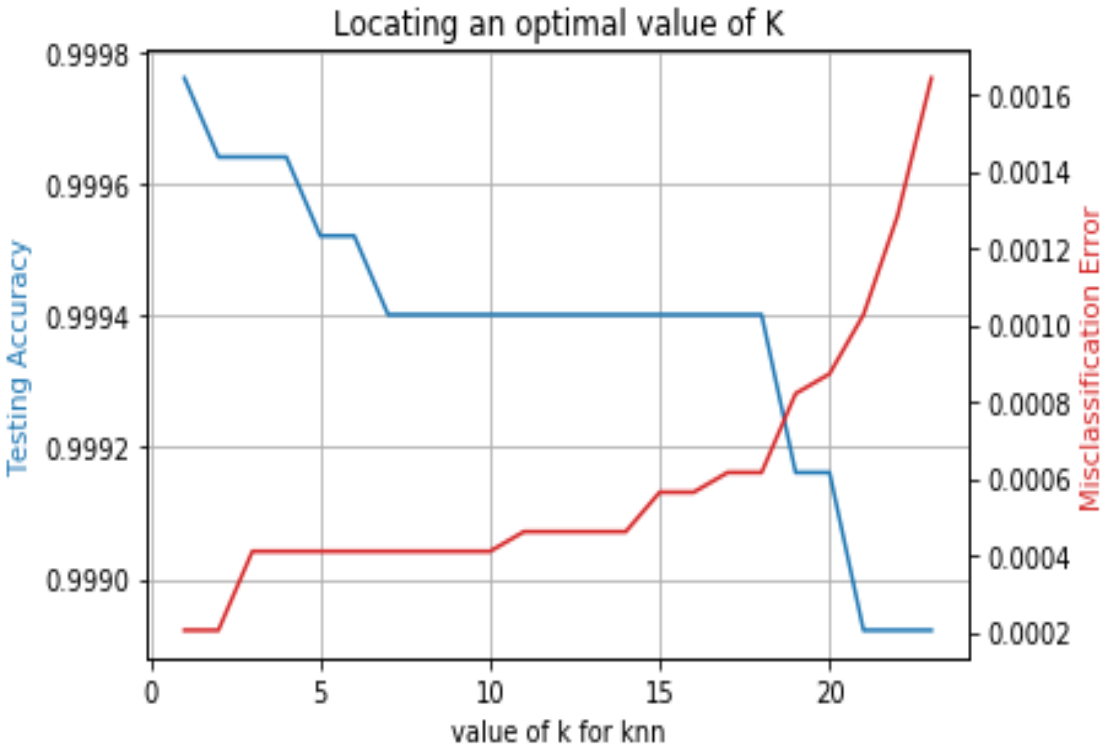
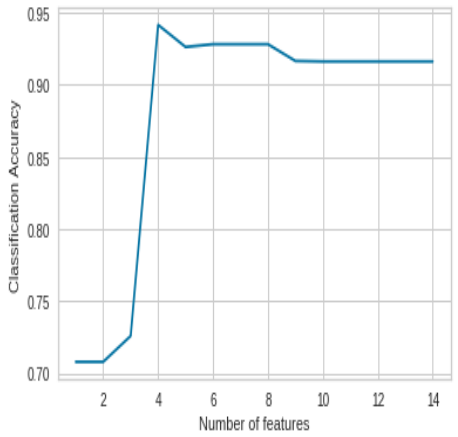


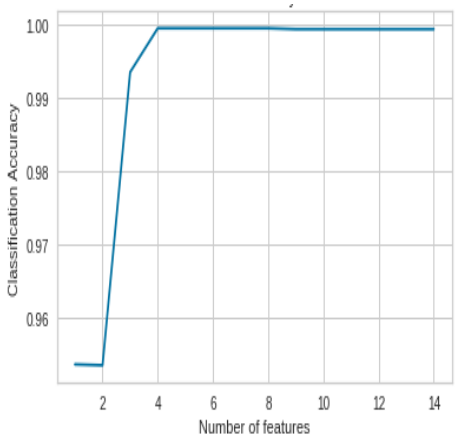
Figure 5.3: Finding and locating an optimal value of K for KNN classifier

Model complexity is decreasing as the value of K is increasing, and the model can generalize the detection of new data. Thus, K=18 is reasonably selected for this particular research.

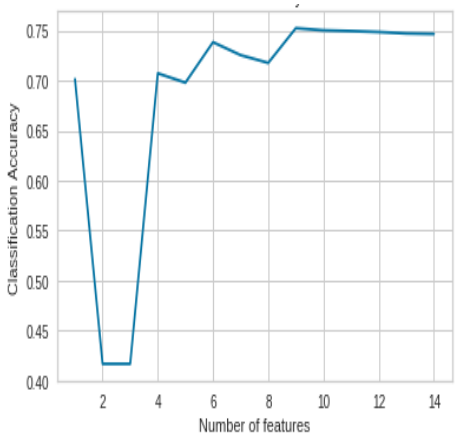
After setting and configuring the required parameter of the classifiers, feature selection methodology is applied to optimized the detection performance of the classifiers. Of course, the feature selection merely depends on the type of classifier that is working on. Each classifier scores different accuracy for the same input features. This behavior is challenging to apply and evaluate the three classifiers. However, a classifier independent feature selection algorithm called Pearson's correlation is used to ensure the feasibility and consistency of the evaluation. Figure 5.4 illustrates the optimal set of features that are contributed to the best performance of all the three classifiers.



(a) Accuracy score response of LR



(b) Accuracy score response of KNN



(c) Accuracy score response of GNB

Figure 5.4: Accuracy score response of LR, KNN and GNB for selected features

In the rest of the modeling process, a train-test split approach is used to split the dataset into training and testing sets. That means, in to *X_train*, *y_train*, *X_test*, *y_test* with parameters of random_state=1 and training_size=0.7(70%). The *random_state* parameter is used to keep track of the experiment consistency regardless of the working environment as long as using the same value, in this case, it is one. Then, train the classification algorithms in the training set. One of the supervised classifier used in this research is Logistic Regression (LR), which is a classification model despite its name. During the model training step, the model has learned the relation between *X_train* and *y_train*. Finally, make a class detection using the testing set. The general procedure used to build, test and validate the selected classifiers in the scikit-learn machine learning tool is described below.

- Import all the required modules into the current workspace of scikit-learn.
- Load the dataset using the panda's data frame method.

- Preprocess and feature engineered the dataset.
- Instantiate the required classification model (LR/KNN/GNB).
- Build/train the model.
- Test/validate the model considering evaluation metrics.

Once the previous procedure is completed successfully, split the whole dataset into training and testing set. Then train the classification algorithms using the training set, pass *X_test*, the feature matrix for the testing set to the predict method that already implemented in scikit-learn. This output the class prediction as one or zero for every observed packet in the testing set, which is stored on the python object variable called *pred_class*. Then, the classifier is predicting for the testing sets. To calculate the accuracy, first import the metrics module from scikit-learn and pass *y_test* and *pred_class* to the accuracy score function. Because the *y_test* contains the true response value for the testing set, the accuracy score function is telling what percentage of the prediction in *pred_class* works correctly. The accuracy of GNB is found to 74.66%, which seems pretty good. Similarly, the accuracy of both LR and KNN is 91.62% and 99.94% respectively. However, at any time use accuracy as evaluation metrics, it is important to compare with null accuracy, which is the accuracy that could be achieved by always predicting the most frequent class in the testing set.

5.1 Null Accuracy

Null accuracy is an accuracy that could be achieved by always predicting the most frequent class. *y_test* consist of both values of once (IP spoofed) and zeros (Normal). Therefore, can count how many instances of once and zeros exist in it. In this case, zero is presented 18,441 times and one is presented 9,399 times, and this is known as the class distribution. Null accuracy answers the question if the model is predicting a predominant class 100% of the time, how often would it be correct. Because, *y_test* contains only once and zeroes, calculate the percentage of once by taking the mean of the vector *y_test*. In this case 33.76% of the values in *y_test* is one (9,399/18,441 + 9,399 =33.76%).

$$PercentageofOnce = \frac{IPspoofed}{IPspoofed + Normal} \tag{5.1}$$

$$Percentage of Once = \frac{9,399}{9,399 + 18,441} = 33.76\%$$

Since only two classes are here, the percentage of zeroes is obtained by taking 1 – the mean of *y_test*, which is 66.24%.

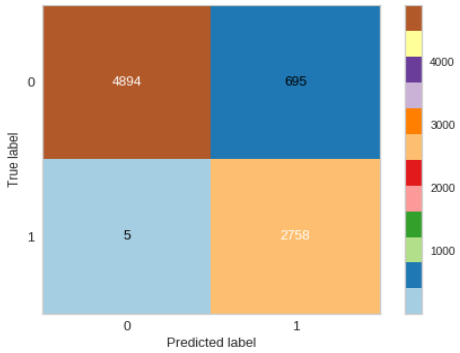
$$Percentage of Zeros = \frac{Normal}{Normal + IPspoofed} \tag{5.2}$$

$$Percentage of Once = \frac{18,441}{9,399 + 18,441} = 66.24\%$$

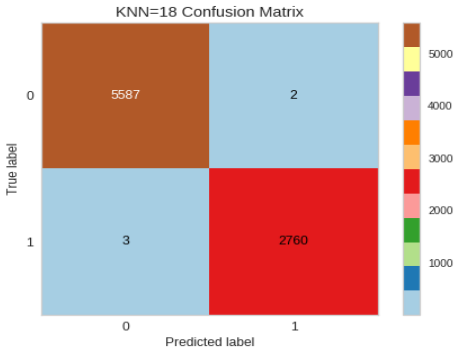
The percentage of zeroes, 66.24% is larger than the percentage of once 33.76%, thus 66.24% is the null accuracy for this problem. In other words, a model that always predicts the packet is not IP spoofed/normal will be right 66.24% of the time. This is not considering as useful metrics but provided the baseline against which might want to be observed the accuracy measure for the three selected classification algorithms. Comparing the null accuracy of 66.24% with the model accuracy of GNB, which is 74.66%, suddenly the model did not look very good. This demonstrated one weakness of accuracy as a model evaluation metrics. Therefore, the accuracy does not tell anything about the underlying distribution of the testing set.

5.2 Detail Classifier Evaluation Results

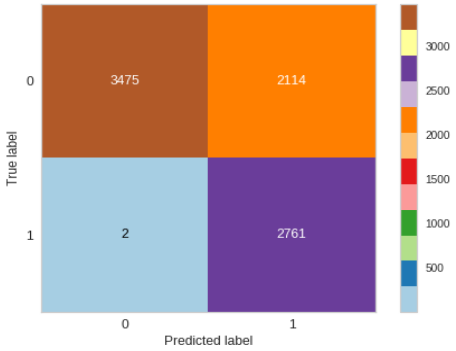
In real attack detection problems like this study, knowing the detail type of errors that the classifiers are making has a great advantage to evaluate detection performance. As discussed in section 4.9.2, several classifier comparison metrics were derived from a confusion matrix, which can capture and give an insight into the type of error in which a classifier is making. Figure 5.5, shows the actual confusion matrix for the LR, KNN, and GNB respectively. The performance evaluation of those classifiers corresponding to their confusion matrix is examined in the next section.



(a) LR confusion matrix



(b) KNN confusion matrix



(c) GNB confusion matrix

Figure 5.5: Confusion matrix of LR, KNN and GNB

Regarding the confusion matrix of the logistic regression, the classifier is performing well in detecting false negative (FN) instance of the testing set and it is only five in number as seen in Figure 5.5. However, a network attack detection problem like this one is really sensitive to FN. Normal packet transactions that are flagged as possible IP spoofed packets are more acceptable than False Negatives (FN), in which IP spoofed packet transactions are missed. Since the former can often be resolved without significantly impacting the smooth operation of the LTE network. Therefore, the ultimate focus of this study is giving attention to FN instances. It does not mean that the other confusion matrix parameters are not relevant. In general, the best classifier is the one that has zero FN and FP instances. Hence, based on the above technical reasons LR is considered a good candidate for the classifier in particular to this study. However, regarding the above core discussions, KNN with K=18 is the best performer upon the confusion matrix results obtained in this research with FN of three (3) and FP of two (2).

As described in section 4.9.2, metrics that are computed from the confusion matrix for an elaborated comparison to a reasonable judgment of classifiers are used for this particular research. The

null accuracy of the three classifies are 66.92% and the same for all, because null accuracy is not a classifier dependent metric. It is just used as a baseline reference to which the classifiers needed to score above this value to consider it as a candidate classifier. One of the generic evaluation metrics of a classifier is accuracy. Regarding this metric KNN score 99.94%, which is the highest score observed on this research relative to LR and GNB as summarized and shown in Table 5.2. On the other hand, GNB scores the least value since it is assumed all the features are independent but they are dependent. However, accuracy is not telling the exact type of error that the undelaying classifier is making. The other metrics that rely on to understand and judge the performance of the selected classifiers are sensitivity and specificity. Considering these two metrics, the three classifiers namely: LR, KNN, and GNB score very nearer values of 99.82% and 87.56%, 99.89% and 99.96%, and 99.93% and 62.27% respectively.

Even though GNB is scoring the highest value of sensitivity, KNN is reasonably be preferred, which score the second-highest value with only a 0.04% difference to GNB. This means out of 100 IP spoofed observations only 0.0004 is missed to detect while all are detected with GNB. Like sensitivity the other metric is specificity. Even though sensitivity is the most focused metrics for this study, a score of closer to one (1) value is desirable for specificity too. Therefore, KNN again scores the highest value of 99.96% and GNB is scoring the least value of 62.17% due to a similar reason mentioned in the case of sensitivity. Contrary to the previous metrics, in which a higher value of one (1) or very close to one (1) is desirable, FPR is the metric in which a lower value of zero (0) or very close to zero (0) is desirable. Based on FPR metrics, again KNN scores 0.03% of best value relative to LR and GNB on which they are score 12.43% and 37.84% respectively. The last but not the listed metric that considered in this study is precision. Like sensitivity and specificity, a value of one (1) or very close to one (1) is desirable for this metric. Therefore, again KNN surpasses by scoring the highest value of 99.93% relative to LR and GNB. Referred to all the above findings of this study KNN is become the outperformed classification algorithm in detecting IP spoofing attacks on the LTE network.

Table 5.2: Summarized classifier results for selected evaluation metrics of the three ML algorithms

Algo.	Metrics					
	Null Accuracy	Classification Accuracy	Sensitivity	Specificity	False Positive Rate(FPR)	Precision
LR	0.6692	0.9162	0.9982	0.8756	0.1243	0.7987
KNN	0.6692	0.9994	0.9989	0.9996	0.0003	0.9993
NB	0.6692	0.7466	0.9993	0.6217	0.3782	0.5663

The other very important metric considered in this research is computational complexity (execution time). Computational complexity becomes an important metric when judging the detection system from the practical implementation point of view. Figure 5.6 illustrates the comparison of the three classification models based on the computational time required for training and testing the model.

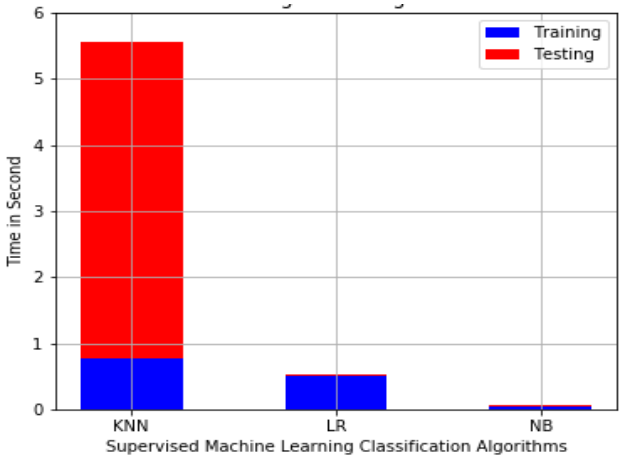


Figure 5.6: Comparison of our three classification models based on training and test time

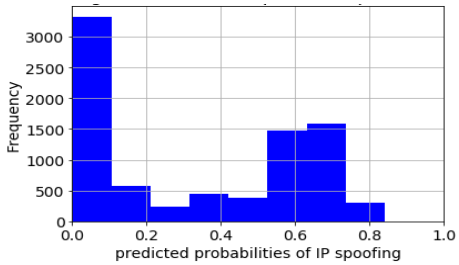
KNN is a computational incentive, as it requires a relatively very high training and testing time of 0.7792sec and 4.7821sec respectively. This is because KNN stores the training set in memory and any time a new observation is evaluated it computes the Euclidian distance amongst the nearest neighbor each time. Thus, this nature of the algorithm forces KNN to have the highest computational complexity. Contrary, GNB surpasses amazingly with the best training time of 0.0421sec. LR on the other hand, outshine with the best testing time of 0.0055sec. The reason that GNB and LR classifiers are scored the minimal time is that the GNB classifier is working based on Bayes theorem, thus apriori and conditional probabilities are learned or rather determined using a deter-

ministic set of steps. This involves trivial arithmetic operations like addition and multiplication, and further normalization is only a division by a scalar; there was no complex mathematical manipulation. Similarly, LR work based on linear mathematics. Thus, these are computed faster in modern-day computers.

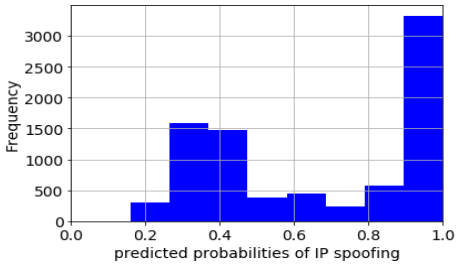
As explained in the above discussions, the sensitivity of the classifier is the key metric that careful in this study. Considering sensitivity and specificity metrics, the classifiers in scikit-learn perform a probability to decide the probability of new observation to classified as normal or IP spoofed. Thus, for every observation, the classifiers computed the probability that the observation is normal and the probability that the observation is IP spoofed, which is equivalent to one minus the probability of normal. Then, the highest probability value is considered and predicted as the class of the newly observed packet. Hence, based on this probability values a threshold is seated in which above this threshold the classifier decides that the class of the new observations. The default decision threshold in scikit-learn is 0.5, an observation that has a probability of less than 0.5 during the model is computing the probability of class prediction is labeled as zero (Normal) else labeled as one (IP spoofed). In scikit-learn adjusting the threshold is giving a headroom to play with and optimized those of the desired metrics sensitivity and specificity for better performance of the classifiers.

5.3 Adjusting the Classification Threshold

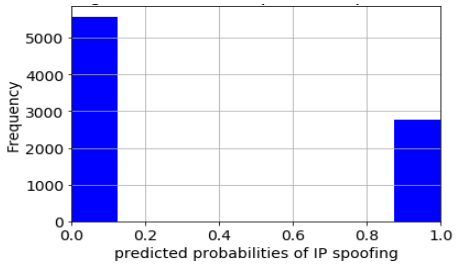
This section is discussing how to modify the performance of a classifier by adjusting the classification threshold. To make this concept clearer, graphically visualize the predicted probability for both classes is important. The effect of adjusting the classifier threshold directly affects the detection performance of a model. The histogram of the predicted probabilities of both classes for each of the three classifiers is depicted in Figure 5.7. This Figure help to demonstrated how adjusting the classification threshold can influence the performance of the model.



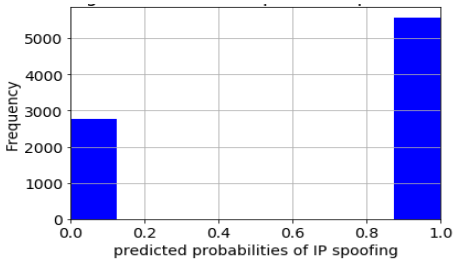
(a) Class zero predict probability of LR



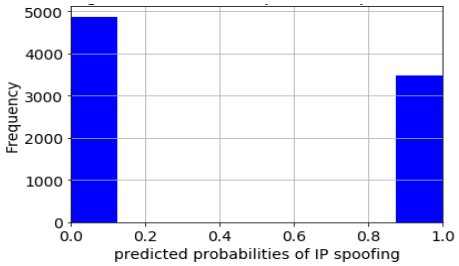
(b) Class one predict probability of LR



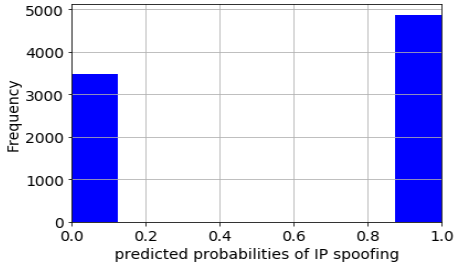
(c) Class zero predict probability of KNN



(d) Class one predict probability of KNN



(e) Class zero predict probability of GNB

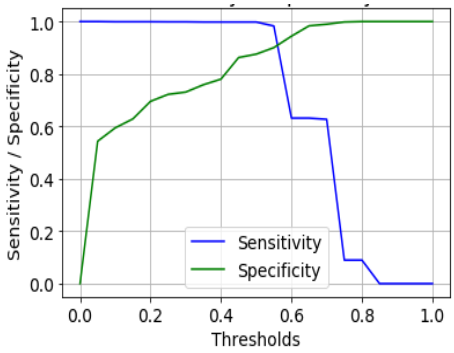


(f) Class one predict probability of GNB

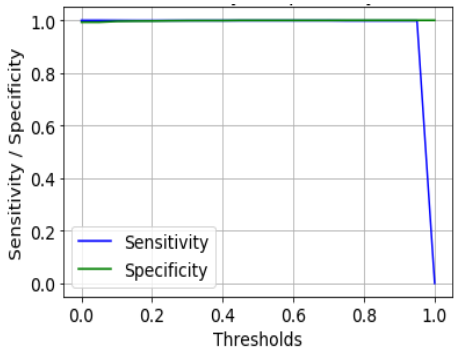
Figure 5.7: Histogram of predict probability for both class zero and one of LR, KNN and GNB

The histogram illustrated in Figure 5.7 shows the distribution of a numerical value of the observations. From LR of class one histogram, almost 1,600 of the observed packets, which are located on the second and third bar have predicted probability values between 0.3 and 0.5. Given the default 0.5 classification threshold mentioned earlier, the testing set observed packets have a predicted probability below the threshold of class one is predicted. Thus, a change of the classification threshold to a number other than 0.5, say to 0.3 can adjust both the sensitivity and specificity of a classifier by adjusting this threshold. For instance, if the threshold for predicting an IP spoofed packets of the LR is adjusting to said 0.3, it can improve the sensitivity of the classifier by 0.07% while significantly degrading the specificity of the classifier by 14.4%. This performance change is occurred just only by shifting the threshold bar from 0.5 to 0.3 such that all the observed packets with predicted probabilities above 0.3 are now predicted as class one. This increases the sensitivity because the classifier is now more sensitive to positive instances.

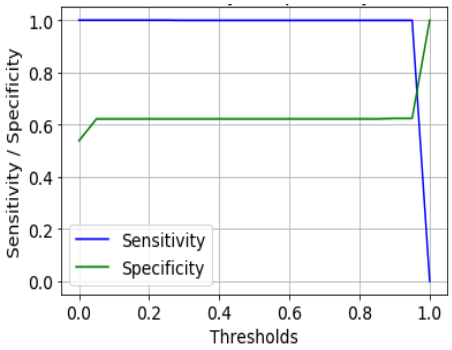
As seen above, a threshold of 0.5 is used by default to convert predicted probabilities into class prediction in scikit-learn. However, for the sake of optimal classifier performance, the threshold needs to be fine-tuned for better sensitivity and specificity as per the business objective of this research. Fortunately, the optimal threshold for all of the three classifiers is observed on the default threshold 0.5. To easily visualize the trade-off between metrics and observe which classifier even performed better by adjusting the threshold, a line graph of sensitivity versus specificity is illustrated in Figure 5.8. often, sensitivity and specificity have an inverse relationship, increasing one is decreased the other. As a result, the optimal sensitivity and specificity of LR are 0.9982 and 0.8756 with a threshold of 0.5, KNN is 0.9989 and 0.9996 with a threshold of 0.5 and GNB is 0.9993 and 0.6217 with a threshold of 0.5 respectively.



(a) LR classifier of class one



(b) KNN classifier of class one



(c) GNB classifier of class one

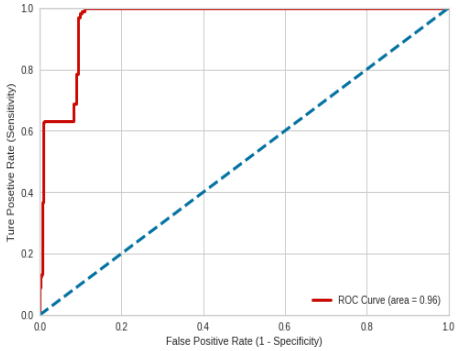
Figure 5.8: Sensitivity Vs specificity for different threshold of LR, KNN and GNB of class one

5.4 ROC Curves and Area Under the Curve (AUC)

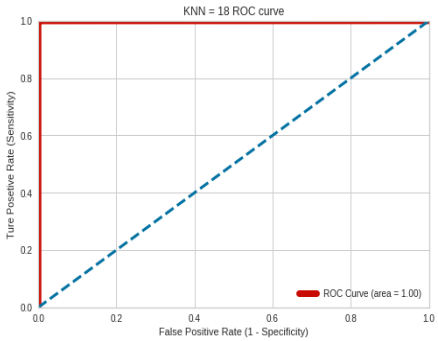
In the previous section, a methodology for finding an optimal threshold was looked at to get the best sensitivity and specificity of classifiers. Alternatively, Receiver Operating Characteristics (ROC) is another mechanism to visually finding an optimal threshold. ROC curve is the plot of

the TPR on the y-axis against the FPR in the x-axis for all possible classification thresholds, to use terminology, the y-axis is sensitivity and the x-axis is 1 – specificity. Figure 5.9 of the LR; for instance, to achieve a sensitivity of say 0.999%, have to accept specificity of around 0.930%. The optimal ROC curve lays on the upper left corner of the plot since that would represent the classifier with both high sensitivity and specificity. In summary, the ROC curve can help to visually choose the threshold that balanced sensitivity and specificity in a way that makes sense for this research problem.

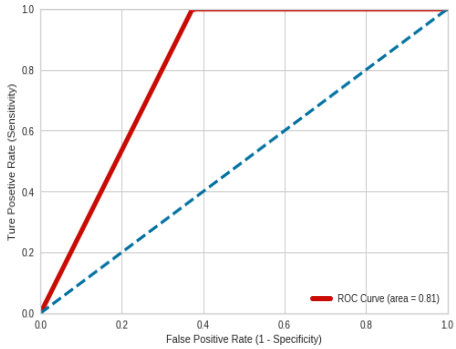
One other metric that is considered in this research known as Area Under the Curve (AUC). AUC is quite literally the area under the ROC curve, meaning the ratio of the area under the actual ROC curve to the area of the ideal ROC curve. Because an ideal classifier would lay on the upper left corner of the ROC curve plot, a higher AUC value is indicative of better overall classifier performance, as such AUC is often used as a single number summary of the performance of classifiers as an alternative to accuracy. AUC of the Logistic Regression obtained in this study is 0.96% and it is 1 and 0.81 for KNN and GNB respectively as shown in Figure 5.9, The best possible AUC for any classifier is one. AUC can also be interpreted as, choose randomly one positive observation and one negative observation from the testing set, AUC represents the likelihood that the classifier will assign a higher predicted probability to the positive observation. It makes sense that this is a useful goal. Because ultimately classifiers needed to rank positive observations higher than negative observations in terms of predicting probability. Therefore. AUC is a useful evaluation metric even when there is a high-class imbalance, meaning that one of the classes dominate. Therefore, KNN is the best performer relative to LR and GNB regarding this research.



(a) LR



(b) KNN



(c) GNB

Figure 5.9: ROC-AUC curve for LR, KNN and GNB of class one

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This study showed the detection of IP spoofing attacks in the LTE network using supervised machine learning classifier algorithms. Even though IP spoofing attack on the LTE network is discovered globally, to the best of my knowledge no study addressed the detection based on a machine learning approach. In addition to this, the nature of the network attack detection approach is merely depending on the mobile network environment such as network architecture, type of services, data usage culture of subscribers and operator's as well as vendor's strategies. Hence, a proposed detection methodology for some particular telecom operator may or may not be resolved the same problem of another operator. Therefore, in this study, a dataset, which was collected from the real Ethio telecom LTE network has been used. Based on this study, three classifiers namely: Logistic Regression (LR), K-nearest neighbor (KNN) and Gaussian Nave Bayes (GNB) are considered as the best-suited classifiers for the LTE dataset.

Accuracy, FPR, sensitivity, specificity, precision, ROC, AUC, and computational time evaluation metrics are being used to compare the best classifier. The choice of metrics ultimately depends on the business objectives, which can guide the model selection process. Based on the nature of the problem that was studied in this research, observed packets that are represented as a positives class as IP spoofed packet. In this case, Ethio telecom judged that False Positives (FP), in which normal packet transactions are flagged as possible IP spoofed packet are more acceptable than False Negatives (FN), in which IP spoofed packet transactions are missed. Since the former can

often be resolved without significantly losing revenue and QoS. Whereas, the later likely result in a significant loss of revenue and QoS. Thus, the priority is to minimize False Negatives (FN), so this study chooses to optimize a model for sensitivity. However, it does not mean that other classification metrics are not important.

According to the technical concepts discussed above, even though GNB scores the heights sensitivity of 99.93%, considering the other metrics KNN is being reasonably considered as the best classifier with a sensitivity of 99.89%, a specificity of 99.96%, precision of 99.93%, FPR of 0.03% and accuracy of 99.94%. But, in most cases of a real situation, KNN is not preferred for the practical implementation of this research. Because KNN is a computationally intensive classification algorithm. As a result, based on considering the computational complexity (testing and training time) findings of this research, KNN is scored 0.722sec and 4.712sec for the training and testing time respectively. On the other hand, GNB is scored the best training time of 0.039sec, whereas, LR is scored the best testing time of 0.005sec. Hence, considering all results of the evaluation metrics on the perspective of practical implementation point of view of this research, LR would reasonably be considered and suggested as the best classifier with sensitivity of 99.82%, specificity of 87.56%, precision of 79.87%, FPR of 12.43%, accuracy of 91.62%, training and testing time of 0.506sec and 0.005sec respectively.

6.2 Future Work

Due to a combination of time limitations, the sensitivity of the data that the problem is dealing with, as well as the study is conducted in the real LTE network, it is impossible to collect large enough packets. However, the thesis provided a detailed approach to the detection of the IP spoofing attack in the LTE network. This research is the first attempt to considered a machine learning-based IP spoofing attack detection mechanism into the complex environment of the LTE core network. While the study and investigation of the thesis have taken place within the Ethio telecom LTE network, the ideas that are resulted from this thesis are applicable in any telecommunication industries that share similar characteristics with it. Hence, based on the experiences that were gained from this research the following future works are suggested for further study.

- Using the results that are obtained in this research, studying the practical test of IP spoofing attack detection on the LTE network can be considered as future work. This can be done either in real-time or non-real time based testing approach.
- Similarly, using the finding of this research as an input, studding on the mitigation mechanisms is suggested as future work.
- This research is targeted on IP spoofing attack detection that is initiated from the user equipment. However, this attack can also be initiated by roaming partners. Hence, studies on the detection and mitigation of this attack on the mentioned entry point can be considered as another future work.
- Likewise, studying this attack detection as well as prevention, which are intentionally initiated from the Ethio telecom employee itself can be considered as another future study.

Bibliography

[1] T. Mshvidobadze, "Evolution mobile wireless communication and lte networks," in *2012 6th International Conference on Application of Information and Communication Technologies (AICT)*, Oct 2012, pp. 1–7.

[2] Ethio-telecom marketing communication. (2018, Oct) Telecom business news. [Online]. Available: http://intranet.ethiotelecom.et

[3] Ethio-telecom marketing communication. (2018, Nov) Telecom business news. [Online]. Available: http://intranet.ethiotelecom.et

[4] Ethio-telecom marketing communication, "Customer base," November 2018.

[5] *Ethio Telecom business news*, July 2016. [Online]. Available: http://intranet.ethiotelecom.et

[6] C. Vorakulpipat, E. Rattanalerdnusorn, P. Thaenkaew, and H. Dang Hai, "Recent challenges, trends, and concerns related to iot security: An evolutionary study," in *2018 20th International Conference on Advanced Communication Technology (ICACT)*, Feb 2018, pp. 405–410.

[7] C. T. I. W. S. Y. Dong W. Kang, Joo H. Oh and Y. J. Won, "A practical attack on mobile data network using ip spoofing," in *AMIS*, 2013.

[8] 3rd Generation Partnership Project Technical Specification Group Core Network and Terminals, "3gpp evolved packet system (eps); evolved general packet radio service (gprs) tunnelling protocol for control plane (gtpv2-c); stage 3," Tech. Rep., Jun 2018.

[9] F. Ahmed, M. Z. Rafique, and M. Abulaish, "A data mining framework for securing 3g core network from gtp fuzzing attacks," in *ICISS*, 2011.

[10] O. Joohyung, K. Dongwan, K. Sekwon, and I. ChaeTae, "3g wcdma mobile network dos attack and detection technology," in *2012 World Academy of Science, Engineering and Technology*, Sep 2012.

[11] P. Xuena, W. Yingyou, and Z. Hong, "Security issues and solutions in 3g core network," in *2011 ACADEMY PUBLISHER*, Feb 2011.

[12] L. He, Z. Yan, and M. Atiquzzaman, "Lte/lte-a network security data collection and analysis for security measurement: A survey," *IEEE Access*, vol. 6, pp. 4220–4242, 2018.

[13] T. Tewodros and T. Ephrem, "Network traffic classification using machine learning: A step towards over-the-top bypass fraud detection," Master's thesis, Nov 2018.

[14] F. A. Narudin, A. Feizollah, N. B. Anuar, and A. Gani, "Evaluation of machine learning classifiers for mobile malware detection," *Soft Comput.*, vol. 20, no. 1, pp. 343–357, Jan. 2016. [Online]. Available: http://dx.doi.org/10.1007/s00500-014-1511-6

[15] 3rd Generation Partnership Project;Technical Specification, "3gpp evolved packet system (eps)," Tech. Rep., April 2017.

[16] GSA, "Evolution of lte market and technology update," November 2015.

[17] 3rd Generation Partnership Project Technical Specification Group Core Network and Terminals, "Evolved universal terrestrial radio access (e-utra) and evolved universal terrestrial radio access (e-utran)," Tech. Rep., Jun 2010.

[18] L. Alcate, "The lte network architecture; a comprehensive tutorial," November 2009.

[19] C.Jeffrey, F.Joshua M, and B.Michael, "Lte architecture overview and security analysis," Tech. Rep., Jun 2016.

[20] P. A. Networks, "Mobile network infrastructure getting started," Tech. Rep., Jun 2018.

[21] D.Debjani, S.Architect, and B.Ravi Raj, "Unlocking long term evolution (lte): A protocol perspective," Tech. Rep., Oct 2010.

[22] 3rd Generation Partnership Project Technical Specification Group Core Network and Terminals, "General packet radio system (gprs) tunnelling protocol user plane (gtpv1-u) (release 15)," Tech. Rep., Jun 2018.

[23] D. Forsberg, G. Horn, W.-D. Moeller, and V. Niemi, *LTE security*. John Wiley & Sons, Jun 2012.

[24] M. Liyanage, M. Ylianttila, and A. V. Gurtov, "A case study on security issues in lte backhaul and core networks," 2016.

[25] INTERNETSOCIETY, "Addressing the challenge of ip spoofing," Sep 2015.

[26] C.Andreas and G.Sarah, *Introduction to Machine Learning with Python*. O'Reilly, Oct 2016.

[27] M. Swamynathan, *Mastering Machine Learning with Python in Six Steps*. Apress, Dec 2017.

[28] M. Swamynathan, *Machine Learning with Python: A Step-By-Step Guide to Learn and Master Python Machine Learning*. O'Reilly, Dec 2018.

[29] S. Raschka, *Unlock Deeper Insights into Machine Learning with This Vital Guide to Cutting-Edge Predictive Analytics*. Packt, Dec 2015.

[30] S. Raschka and V. Mirjalili, *Python Machine Learning Second Edition: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow*. Packt, Sept 2017.

[31] scikit learn, *scikit-learn documentation*, Sept 2018.

[32] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2210832718301546