Detection and Classification of Bypass (SIMbox) Fraud using Deep Learning: The Case of Ethio Telecom

Gadisa Adamu Mitiku

A Thesis Submitted to the Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master of Science in computer science and Engineering

Office of Graduate Studies

Adama Science and Technology University

June 2023

Adama, Ethiopia

Detection and Classification of Bypass (SIMbox) Fraud using Deep Learning: The Case of Ethio Telecom

Gadisa Adamu Mitiku

Advisor: Ketema Adere Gemeda (PhD)

A Thesis Submitted to the Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master of Science in computer science and Engineering

Office of Graduate Studies

Adama Science and Technology University

June 2023

Adama, Ethiopia

Declaration

I hereby declare that this Master Thesis entitled "**Detection and Classification of Bypass (SIMbox) Fraud using Deep Learning: The Case of Ethio Telecom"** is my original work. That is, it has not been submitted for the award of any academic degree, diploma, or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation

Gadisa Adamu Mitiku	_____	_____
Name of the student	Signature	Date

Advisor Recommendations

I/we, the advisor(s) of this thesis, hereby certify that I/we have read the revised version of the thesis entitled "**Detection and Classification of Bypass (SIMbox) Fraud using Deep Learning: The Case of Ethio Telecom**" prepared under my/our guidance by **Gadisa Adamu Mitiku** submitted in partial fulfillment of the requirements for the degree of Masters of Science in computer science and Engineering. Therefore, I/we recommend the submission of the revised version of the thesis to the department following the applicable procedures.

Ketema Adere Gemeda (PhD) _____ _____

Advisor Signature Date

Approval page

I/we, the advisors of the thesis entitled "**Detection and Classification of Bypass (SIMbox) Fraud using Deep Learning: The Case of Ethio Telecom**" and developed by **Gadisa Adamu Mitiku** hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Ketema Adere Gemeda (PhD) _____ _____

Advisor Signature Date

We, the undersigned, members of the Board of Examiners of the thesis by **Gadisa Adamu Mitiku** have read and evaluated the thesis entitled **Detection and Classification of Bypass (SIMbox) Fraud using Deep Learning: The Case of Ethio Telecom**" and examined the candidate during the open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in computer science and Engineering.

_____ _____ _____

Chairperson Signature Date

_____ _____ _____

Internal Examiner Signature Date

_____ _____ _____

External Examiner Signature Date

Final approval and acceptance of the thesis are contingent upon the submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

_____ _____ _____

Department Head Signature Date

_____ _____ _____

School Dean Signature Date

_____ _____ _____

Office of Postgraduate Studies, Dean Signature Date

ACKNOWLEDGMENTS

I admit that none of my achievements would have been possible without God's blessing. Every step I've taken has revealed God's hand, illuminated my path, and given me strength in the face of difficulty. His supernatural wisdom has guided my judgments, allowing me to confidently and trustingly traverse life's uncertainties. Thank you, Jesus, for being my everything, Holly God's son!

Next, I'd like to express my heartfelt gratitude and appreciation to everyone who has helped me through this thesis. First and foremost, I would like to thank my adviser, **Ketema Adere (PhD**), for their advice, tolerance, and invaluable insights during the entire study process. Their knowledge and consistent assistance were critical to the effective completion of this thesis.

Finally, I'd like to thank my parents for instilling in me a passion for learning and for always being my rock. Your sacrifices and guidance have formed me into the person I am today. This research would not have been feasible without my family's love, understanding, and continuous support. I am very privileged to have such an incredible and supportive family, and I will be eternally grateful for their presence in my life.

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

ANN	Artificial Neural Network
CDR	Customer Data Record
CFCA	Communication Fraud Control Associations
CLI	Calling Line Identification
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DCNN	Deep Convolutional Neural Network
DL	Deep Learning
DNN	Deep Neural Network
ETB	Ethiopian Birr
FMS	Fraud Management System
FN	False Negative
FP	False Positive
FPR	False Positive Rate
GB	Giga Byte
GBC	Gradient Boosting Classifier
GDP	Gross Domestic Product
GPGPU	General Purpose Graphic Processing Unit
GSM	Global System for Mobile Communication
HBS	Human Behavior Simulation
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
IRSF	International Revenue Share Fund
ISP	Internet Service Provider
IUC	International Interconnect Usage Charge
KNN	K-Nearest Neighbor
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error

ML	Machine Learning
MLP	Multilayer Perceptron
MO	Mobile Originator
MSC	Mobile Switching Centre
MSE	Mean Square Error
MT	Mobile Terminator
PBX	Public Branch Exchange
RAM	Read Access Memory
RF	Random Forest
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic Curve
RQ	Research Question
SDC	Standard Distribution Control
SIM	Subscriber Identity Module
SVM	Support Vector Machine
SW	Sliding Window
TB	Tera Byte
TCG	Test Call Generators
TN	True Negative
TP	True Positive
TPR	True Positive Rate
TV	Television
VOIP	Voice Over Internet Protocol

Abstract

Telecom frauds are a major problem that affects operators and telecoms organizations all around the world. Fraudsters have used technological advancements in the organization to their advantage to commit fraud. SIMBox/Bypass fraud is one of the most common types of telecom fraud, and it involves the use of Voice over IP (VoIP) technologies to avoid access charges and profit on international calls. Due to the dynamic nature of this fraud, it can quickly defeat the Test Call Generators (TCG) and FMS (Fraud Management System). In addition to TCG and FMS near real-time machine learning approach has been designed to detect and classify SIMbox/Bypass Fraud which required UpToDate data a problem and supervised machine learning algorithms show limitations to capture the dynamic nature of Fraud. To detect and classify SIMBox/Bypass fraud, deep learning/deep neural network classifier algorithms were used in this study, namely MLP (multilayer perceptron), RNN (recurrent neural Network), and LSTM (long short-term memory), with the two validation techniques 10-fold cross-validation and separate train test. After collecting Call Detail Record (CDR) data from Ethio Telecom, relevant features were chosen and preparation operations such as data cleaning, integrating, and aggregating were completed. The experimental results demonstrate that MLP (multilayer perceptron) classifier with a separate test data validation technique obtains a higher classification accuracy (99.17%), LSTM (long short-term memory) is ranked second with an overall performance accuracy of 98.90%, and RNN (recurrent neural Network) is ranked third with an accuracy of 98.10%. Detecting and classifying SIMBox/Bypass fraud has numerous advantages for telecom companies, including raising awareness to mitigate risk, preventing financial loss, increasing revenue assurance, improving operational efficiency, improving customer satisfaction, protecting brand reputation, ensuring regulatory compliance, and establishing industry leadership.

KEYWORDS: *SIMBox/Bypass fraud, Deep Learning, Deep Neural Network, multilayer perceptron, long short-term memory, Recurrent neural networks*

CHAPTER ONE

INTRODUCTION

1.1 Background of the study

Assume you receive a call from an unknown local number, pick up the phone, and it's from a family member or friend living abroad; it does seem strange to receive an international call from a local number; this is Bypass/SIMbox Fraud. International calls are routed through the Internet to a cellular device, which injects them back into the cellular network via SIMbox containing several inexpensive Prepaid SIM cards. As a result, the calls become local at the destination network, and the fraudsters who set up these boxes pay mobile operators only local rates after charging international rates at the origin. The person calling will pay the entire call termination fee, but it will not be collected by his or her local operator; this fee will go to the fraudster in charge of the SIMbox and other companies in charge of directing the calls from source to destination. Most people associate fraudsters with hackers, but this is not the case always. Most of the time, they are business people who see fraudulent activity as a business opportunity in which they are willing to invest time and money to benefit. As a result, we believe that fraudsters can be defeated simply by putting them out of business (Ighneiwa & Mohamed, 2017a).

Following the tremendous growth in telecommunication networks in recent years, which has become such an important part of our lives, service providers and telecommunication companies are now facing a new major challenge of various types of fraud. With millions of dollars in profits, this quickly spread around the world. Fraud in telecommunications networks is generally defined by fraud circumstances, which effectively explain how the fraudster got illegal network access (Ogwueleka, 2010).

Telecommunications fraud is defined as any deliberate act of cheating and embezzlement in the use of telecommunications facilities committed by persons or organizations to avoid the cost of services or tracking recorded conversations. In other words, it is the unauthorized use of a telco operator's infrastructure and resources by a third party with no intention of compensating them. It is a significant issue because it has a financial impact on the company (Karunathilaka, 2020).

In Ethiopia, one of the telecom companies, Ethio Telecom, is losing money in foreign currency as well as revenue. Telecom companies develop FMS to protect themselves from fraudulent activity.

One of the most popular prevalent types of interconnect bypass fraud that affects telecom operators is SIMbox/Bypass fraud. To combat SIMbox/Bypass fraud, telecom companies primarily employ three strategies. TCG telecom operators use the international gateway to make international calls to their network. FMS is a rule-based fraud detection technique, and controlling SIM card distribution is the practice of restricting the number of SIM cards given to customers (F. Tesfaye, 2020).

1.2 Motivation

Over the previous two decades, the telecom industry has evolved, and modes of communication have changed and progressed to satisfy the needs of customers and enterprises. With the advancement of technology, telecom fraud has become an important research area. Telecommunications fraud is a problem that affects operators all over the world, and one of the most well-known Telecom frauds is SIMbox/Bypass fraud, which is illegally used in bypassing international voice traffic to a SIMbox device to avoid carrier charges by causing opportunity loss of international interconnect usage charge (IUC) to operators. As a result, cellular service providers around the world lose billions of dollars each year (Airn, 2018). In addition to this, telecom fraud causes unintended losses for mobile operators such as decreased service quality, denial of service, network congestion, Customer churn, and dissatisfaction are major challenges that arise as a result of telecom fraud. This Fraud could have serious ramifications. In addition to harming the country's economy, there are numerous consequences for the customer. Even though fraud is a major problem, Traditional approaches to SIMBox fraud detection usually rely on rule-based systems or heuristic methods, which are limited in their effectiveness due to fraudsters' constantly evolving strategies. Deep learning, a subset of machine learning techniques, has proven to be extremely effective in detecting anomalies. Deep learning systems for detecting and classifying SIMBox/Bypass fraud have great promise for significantly enhancing telecommunication security.

As a result, we are motivated to solve this problem utilizing deep learning/deep neural networks, which have the potential to detect and classify SIMbox/Bypass Fraud while overcoming the dynamic nature of Fraud. The proposed method trains deep neural networks using large-scale data sets, such as call detail records and Cell IDs. Using a deep learning approach, we are aimed to detect and classify SIMbox/Bypass fraud, solving a major challenge for telecommunications service providers.

The practical consequences of this research for telecommunications service providers, regulatory organizations, and law enforcement agencies are enormous. The suggested approach reduces revenue losses and protects the integrity of telecommunication networks by increasing the detection and classification of SIMbox/Bypass fraud.

1.3 Problem Statement

SIMbox (Bypass) fraud is a common problem in the global telecom sector, particularly in some Asian and African countries such as Ethiopia, where international termination rates are significantly higher than local (retail) termination rates, allowing fraudsters to bypass an international call and make a local call using a SIMbox device. SIMbox/Bypass fraud is a major source of loss in the telecom operator sector on multiple fronts. One of the key issues that is regularly reported in numerous research studies is the dynamic nature of fraud. This form of fraud is not defined by a specific event or style (Raghavan & Gayar, 2019). In addition, concept drift and data limitation are prominent issues raised in related works. Even though it attempted to detect and Classify this bypass fraud previously using an FMS (Fraud Management system) approach, which requires massive computational power for pattern matching and rules that must be specially derived for the working domain, near real-time machine learning algorithm which requires up-to-date data, and other Supervised Machine Learning Algorithms such as KNN, and Random forest show a limitation on detecting and classifying fraud because of their dynamic nature(Anthony & Joy, 2019; Djomadji et al., 2023; F. Tesfaye, 2020; Ighneiwa & Mohamed, 2017a; Shing Lim et al., 2021; Terzi et al., 2021; Veloso et al., 2020).

Because of the rigorous regulations of the traditional detecting method, fraudsters can adapt to the existing FMS (Fraud management system) detection rules and policies. Following that, they raise the company's revenue losses while also hurting the company's brand and undermining subscribers' trust ties with the company. This FMS (Fraud management system) is incapable of overcoming fraudsters' new behavioral change.

To overcome the stated problem, we proposed a Deep Learning methodology that could give a solution. This approach extracts hidden knowledge or patterns from the CDR data of subscribers. This data-driven detection type looks for behavioral changes in the subscriber's data history

Without the use of established rules, such a method allows for the detection and classification of fraudsters by overcoming their dynamic nature.

1.4 Research Questions

RQ1. What are the features that can be used to detect and classify SIMbox/Bypass fraud?

RQ2. What are the available deep learning algorithms for detecting and classifying SIMbox/Bypass fraud?

RQ3. Which deep learning algorithm will efficiently detect and classify SIMbox/Bypass fraud?

1.5 Objectives

1.5.1 General Objective

The overall goal of the research is to detect and classify SIMbox/Bypass fraud for Ethio Telecom using Deep Learning on user CDR data.

1.5.2 Specific Objectives

The specific objectives are:

- To identify the features which are used to detect and classify SIMBox/Bypass fraud.
- To prepare the data for the deep learning model. (Feature Extraction and Engineering).
- To investigate the appropriate deep learning algorithm for detecting and classifying SIMbox/Bypass fraud.
- To create and train the model using preprocessed data.
- To compare and evaluate the models' performance.

1.6 Significance

The role of fraud detection and classification in telecommunications companies is critical in preventing fraudsters from causing financial loss, and reputation damage, and invading their customers' private information, thereby increasing customer satisfaction. Aside from that, it will contribute to the telecom industry in the following ways:

- Discovers uncovered hidden fraudulent behavior.
- Creating awareness and assisting telecom operators in detecting and classifying SIMbox/Bypass fraudsters.

- Provide practical suggestions that may assist anti-fraud managers and employees in gaining a better understanding of SIMbox/Bypass fraud.
- Provide insight into the company's traditional fraud management system and its shortcomings.
- Proposes suitable deep learning algorithms for SIMbox/Bypass fraud detection.

1.7 Scope and Limitations

Today, there are numerous types of telecom fraud and data available. However, this research only focused on detecting and classifying SIMbox/Bypass fraud. A variety of methods and tools may be employed to detect SIMBox/Bypass fraud, even though we have applied deep learning/deep neural network algorithms to detect and Classify SIMBox/Bypass fraud. CDR data from Ethio Telcom company were used as a data source for this thesis work.

1.8 Thesis Organization

The following is the overall structure of this thesis research: Chapter 2 discusses Telecom fraud, its types, and its effect on the telecom industry by demonstrating scenarios and depicting various Techniques for detecting and preventing SIMBox/Bypass fraud, SIMBox devices, deep learning algorithms, Activation functions and also related research done in this area by various researchers. The third chapter is entirely devoted to discussing the Research Methodology including problem formulation, data preparation, preprocessing, feature selection, model training, performance measuring metrics, and hardware and software tools used in this research. Chapter 4 discusses the proposed solution of the study and implementation of the proposed solution in addition to this, Tasks completed during the implementation of the proposed solution process, including Preprocessing of data and feature selection, algorithm training, and evaluations. The outcomes of the performance evaluation of the algorithms' model are discussed in Chapter 5. Chapter 6, the final chapter, contains the conclusion, recommendation, contribution, and future work.

CHAPTER TWO

LITERATURE REVIEW AND RELATED WORK

2.1 Overview of Bypass (SIMbox) Fraud

In modern society's daily activities, telecommunications are becoming increasingly significant. Pressure on telecommunications providers increases as customer demands shift from simply being connected to being connected at all times, but it also opens the door for new types of digital crime (Łuczak et al., 2021). The fraud scenario is one of the most popular and important concerns in the industry of telecommunications. The operation of telecom industries is becoming increasingly concerning due to the dynamic nature of fraud and the misuse of technological advancement. As a result, telecom companies are putting in extra effort to protect their services from fraudsters and reduce revenue leakage (Alenezi & Qureshi, n.d.). Among fraud scenarios, SIMBox/Bypass fraud has a significant impact on the telecom industry, costing companies $6 billion per year and ranking as the most expensive fraud in the world. It is an ongoing threat to network operators' revenue because it is challenging to predict its properties such as time, location, and method of attack. Fraud is quickly becoming one of the most popular serious threats to the telecom industry, and it appears to be evolving. Telecommunication fraud refers to a diverse range of illegal activities that take place on telecom operator networks. There are various types of fraud, all of which harm voice bandwidth, quality of service, network resources, and the revenue of service providers. Such frauds include Spamming with phone calls and SMS, private branch exchange fraud, subscription fraud, premium rate fraud, domestic revenue sharing deception(fraud), and SIMBox fraud (Abuhamoud et al., 2021).

SIMBoxes intercept international voice calls and redirect them via the Internet to cellular device networks, where they are re-injected. As a result, not only do calls appear local at the destination network, but the intermediate and destination network cellular operators are not paid for call routing and termination, resulting in economic losses (Alsadi & Abuhamoud, 2018). and SIMBox/Bypass fraud is a significant source of revenue loss for telecom operators which require well-regulated fraud detection technologies and standards to protect themselves and their customers from SIMBox/Bypass fraud (Abuhamoud et al., 2021). Companies in the telecoms industry, particularly in Africa, sustain significant losses as a result of fraud. One of the most

common types of fraud is SIM box bypass fraud, which entails utilizing SIM cards to reroute incoming international calls from mobile operators, causing enormous revenue losses (Djomadji et al., 2023).

2.2 Bypass Fraud

A legitimate path/route call will be bypassed, resulting in revenue loss. National and international calling rates are typically set by a country's regulators or by an individual or group of operators. Bypass fraud is common in countries where calling rates distinguish between domestic and international calls. Furthermore, Government operators have a monopoly on international gateways in some countries. The rate differential is used by the fraudsters to ensure a sufficient profit, which is the primary motivator for them to invest in acquiring the equipment and GSM connections needed to conduct a large-scale Bypass fraud. In countries with low, nonexistent, or negative international-to-national terminating charge margins, bypass fraud either does not exist or is conducted on a very small scale. It is one of the most popular, current, and serious threats to the revenue of a telecom operator. It is the unauthorized use or manipulation of a network operated by a third party. This can occur in two ways (Baskar, 2021).

- SIMBox Interconnect Fraud
- GSM Gateway Fraud

Fraudsters benefit from using such methods to avoid such high-cost interconnections and deliver expensive international calls illegally. Fraudsters employ Voice over Internet Protocol (VOIP) - Global System for Mobile Communications (VOIP-GSM) gateways, often known as "SIMBoxes," to receive incoming calls (through wired connections) and deliver them to a cellular voice network. It looks to be a local call made from the phone of a verified consumer. Not only does this practice degrade legitimate customers' network experience while also violates many countries' telecommunications laws (Baskar, 2021).

To understand bypass fraud, we will first describe the legal way to make international calls. Assume callers A and B are from separate countries. Caller A uses the mobile operator to dial caller B's phone number. The call is answered by Country A's mobile operator and routed through his international gate to a transient operator. The transient operator then pays a toll and routes the call

to the country B mobile operator via voice-over-IP (VoIP). Following that, country B's mobile operator terminates the call to caller B via his network. Figure **2.1** depicts the legal route (Ighneiwa & Mohamed, 2017b)

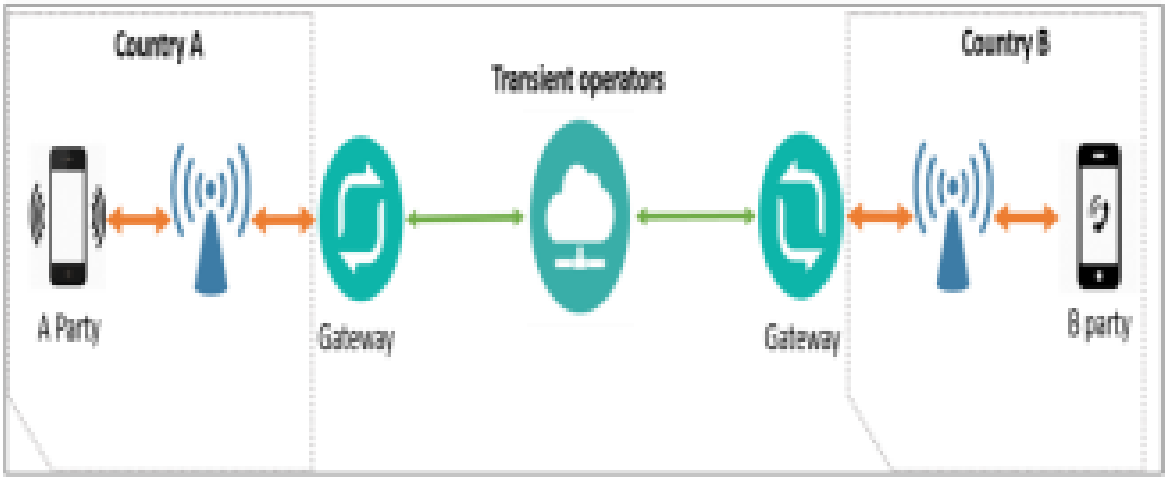

Figure 2.1 The Legitimate Route of International Call (Ighneiwa & Mohamed, 2017b)

In bypass fraud, the transient operator routes the call via VoIP through a SIMbox in country B, and the SIMbox reroutes the call via the country B mobile operator, paying only for the local call. Figure **2.2** depicts the bypass fraud route (Ighneiwa & Mohamed, 2017b).

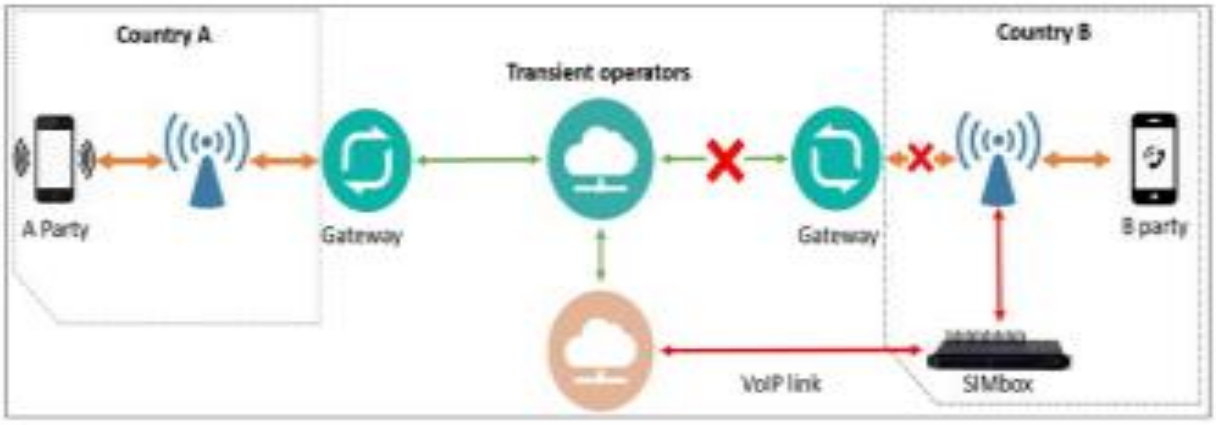

Figure 2.2 The Bypass Fraud Route of International Call (Ighneiwa & Mohamed, 2017b)

The benefit, in this case, is that the toll charge levied by country B's mobile operator is waived. Because the bypass fee is significantly higher than the local call fee, it is financially viable.

2.3 SIMBox

In the underground network of large-scale frauds that steal billions of dollars from individual victims and mobile network providers throughout the world, SIMboxes have been a key component (Oh et al., 2023). The SIMBox converts international traffic to local traffic, which is then reflected on the destination network (at MT Level). This fraudster pays only local charges or, on occasion, uses low-cost local tariffs and cards to terminate traffic. The calling party pays international call rates for connecting calls, but carriers and destination operators are unable to collect international call carrier charges and termination fees for the call. The fraudster earns money by illegally routing calls and reaping the benefits of international carriers and termination fees, but in the end, the fraudster must pay only local termination fees. SIMBox Causes economic loss by degrading the operator's services, often call quality, and congesting the network, resulting in dissatisfied customers and churn. Detecting SIMBox across enormous international voice traffic is analogous to looking for a few needles in a haystack full of little items that look like needles. While operators of intermediate and destination networks have strong financial incentives to solve the problem, they lack the data to analyze the lost international calls because the traffic terminating MSC is only local traffic (Airn, 2018).

When fraudsters install SIMboxes with multiple low-cost, prepaid SIM cards, they commit bypass fraud. SIM slots and antennas are included in SIMbox equipment. An Ethernet port connects the SIMbox to the internet. A SIMbox and its components are depicted in Figure **2.3** (Ighneiwa & Mohamed, 2017a).

Figure 2.3 A SIMbox and its Components (Ighneiwa & Mohamed, 2017a)

2.4 Types of Telecom Fraud

There are various types of telecommunication fraud, and previous works have classified them in various ways; however, almost all of them have classified telecommunications fraud based on the methods by which fraudsters gain unauthorized access (Marah, 2015; Zheng et al., 2018) made a very broad classification of fraud, dividing it into subscription fraud and superimposed fraud. Subscription fraud occurs when a fraudster obtains an account for which they do not intend to pay for the services (high debt fraud is also under this category). The account is entirely genuine, and fraud occurs while it is active. Another type of subscription fraud is registering under a false name. It is worth noting that there are two types of users: domestic and commercial, with the latter incurring a higher cost due to commercial users' higher level of usage (Saravanan et al., 2014).

Subscription fraud occurs when a commercial user registers as a residential user to save money on communication costs. The authors of (Estévez et al., 2006) have classified Subscription fraud into two types based on intent: (a) for-profit and (b) for personal use. Subscription fraud is likely the most difficult sort of telecommunications fraud to identify, and it can result in significant revenue loss for enterprises (Kau & Kogeda, 2019). Superimposed fraud occurs when fraudsters take control of a legitimate customer's account. Examining call records on bills is a common method for detecting superimposed fraud (Subuddhi & Panigrahi, 2015). Similarly, (Kou et al., 2004) used the same approach to categorize fraud as superimposed fraud or subscription fraud. The authors further classified superimposed fraud as phone cloning, ghosting, insider, and tumbling, while insolvent cases were classified as a subcategory of subscription fraud. Another classification proposed by (Lopes et al., 2011) divided fraud types into internal and external fraud according to their origin and nature. External fraud conceals the names of the perpetrators due to the nature of the source, which is from outside the organization and frequently has no geographical limits. An internal attack, in contrast, originates within the organization, making the investigation process simpler. As stated by (Ighneiwa & Mohamed, 2017a) The top three types that cause a significant loss are:

A. International Revenue Share Fraud (IRSF)

According to CFCA, IRSF is the most significant contributor to overall fraud losses. When a fraudster strikes a deal with a local carrier in a high-cost destination to split profits for increased

traffic, the fraudster gains illegal access to the public branch exchange (PBX) of an organization) and generates calls. The fraudster then generates high-traffic calls to expensive destinations to profit from the sharing agreements (Elmi and colleagues (2013)

B. Premium Rate Service Fraud

You've probably received a text message informing you that you've won a large prize and that all you have to do is call a specific number. If you dialed that phone number, you were duped into using a premium rate service. A premium rate service agreement exists between some telecom companies and service providers to share revenue generated by traffic to the premium service number; it is used in television shows, contests, and entertainment services. The fraudsters attempt to stimulate customers by leaving a missed call or message, then profit from a share of the callback revenues, and the third telecom fraud is bypass fraud (Elmi and colleagues (2013).

2.5 Battling Bypass Fraud

A. Test Call Generation (TCGs)

Test call generation is an active method of detecting bypass fraud in which operators test various international routes to their network to determine whether calls are routed through legitimate or SIMbox routes. This technique detects fraud while minimizing false positives (that is when a normal user is assumed to be a SIMbox). This method, however, is probabilistic and costly due to the need to test a large number of international routes. Also, as we will see in the anti-spam method, fraudsters employ tricks to avoid the detection of test calls. TCG, in contrast, cannot be used as a trial-and-error method across a large set of numbers (Elmi et al., 2013).

B. Fraud Management Systems (FMS)

Measures are used by fraud management systems to detect abnormal SIM card usage. FMS analyzes Call Detail Record (CDR) data to create usage profiles that differentiate between normal users and SIMboxes (Elmi et al., 2013)

C. SIM Card Distribution Control (SDC)

SIM cards are essential in the bypass cycle, and fraudsters must keep a sufficient supply on hand to remain profitable. SIM card distribution control, on the other hand, will complicate the process.

Fraudsters will be prevented from obtaining a large quantity of SIM cards to install in their SIMboxes by requiring government IDs and limiting the maximum number of SIM cards permitted per ID (Elmi et al., 2013).

2.6 How Fraudsters Avoid Detection?

As detection technology progresses, fraudsters devise new tactics to avoid detection and boost profits. This section discusses the many methods used by fraudsters to avoid SIM blocking.

A. Anti-Spam (Test Call Detection)

Using different routes to known local network numbers to generate test calls (TCG) is among the most effective methods for detecting SIMs used in SIMboxes. Whether it is from a local or international number, the incoming call will be displayed; if it is from a local number, it must be connected to a SIM card that is used in a SIMbox and can be quickly processed by the fraud department. However, depending on use and other trends, the fraudsters can tell whether the calls are actual subscriber calls or originate from a TCG system by analyzing the voice call traffic that enters their SIMboxes. To escape discovery, they might either block the test calls and prevent them from ever reaching the SIMbox or divert the calls to an authorized route (Ighneiwa & Mohamed, 2017a)

B. Human Behavior Simulation (HBS)

According to the literature (Murynets et al., 2014), some features, such as 1. The SIMbox is not moving, 2. The majority of calls are incoming. 3. No use of network services such as SMS, GPRS, and others. Smart SIMboxes, by contrast, use Human Behavior Simulation to mimic the behavior of regular customers (HBS). If no sophisticated detection techniques are utilized, this strategy makes identifying fraudsters very challenging. HBS includes the following elements:

1. SIM Migration (Movability)

Fraudsters set up multiple gateways in different locations, such as one in the city center and another in a shopping mall or other crowded area, and they occasionally swap the SIM cards between the gateways to make it appear as if the user is moving. The swapping operation could be carried out manually or automatically via software.

2. **SIM Rotation**

SIMboxes are easily detected if fraudsters use their SIMs excessively during the hour, so they limit their usage by rotating the SIMs as workers shifts. This will cause SIMs to operate for a limited number of hours per day, simulating the behavior of ordinary customers.

3. **Usage of Other Network Services**

The majority of SIMboxes only use voice services, which renders them susceptible to discovery. Smart SIMboxes are communicating with one another via calls and SMS to help alleviate this problem. Additionally, they occasionally use some of the network operator's internet services.

4. **Family Lists Traditional**

SIMboxes simply reroute VoIP calls to the GSM network, allowing them to call a sizable number of different network customers. A brilliant alternative is to employ family lists, in which each SIM is allocated to redirect calls to a specific list of numbers. As a consequence, the trap of big distinct numbers detection is avoided.

To summarize, Human Behavior Simulation (HBS) makes combating bypass fraud more difficult and time-consuming. To address this issue, advanced measures must be implemented (Ighneiwa & Mohamed, 2017a).

2.7 The Impact of Bypass Fraud on The Global Telecommunication Service Providers

Telecommunication fraud, which causes significant losses, is a rapidly expanding area of criminal activity and a low-risk substitute for more conventional forms of financial crime. Due to the complex interpersonal dynamics and inherent unpredictability that the telecommunication network brings, telecommunication fraud detection has grown to be a complicated and crucial topic that has to be investigated (Ji et al., 2020). According to the Communications Fraud Control Association, mobile operators lose more than $28 billion per year due to telecom fraud and revenue loss (CFCA). In an industry where margins are being squeezed, fraud is the single biggest preventable revenue threat to mobile operators, and SIMbox is a significant contribution in some regions. When people or groups purchase hundreds of SIM cards with low-cost calling plans, they commit SIMbox/Bypass fraud. The SIM cards are then illegally used to terminate calls, diverting them

away from telecom network operators. This type of fraud is especially common in countries where the cost of terminating international calls is prohibitively expensive, resulting in an estimated $6 billion in global revenue losses per year. The global impact of the SIM box fraud on the other side, goes far beyond company balance sheets. Countries are losing revenue streams, and there are other societal ramifications. Operators have lost up to 90% of their call termination income as a result of unauthorized bypass routes. This is not only bad for the networks; many countries tax telecommunications services, so every penny lost to fraud reduces tax revenues that could have been invested in the country's infrastructure and vital services. The expansion of telecommunications services has aided the economic growth of emerging continents such as Africa. According to industry projections, the African mobile market will hit two key milestones in the next five years: 500,000 mobile subscribers in 2021 and a billion mobile connections in 2024. SIM box scammers are exploiting the region's growth and higher interconnection tariffs to avoid traditional call termination systems and route calls through local SIM gateways. This limits the ability of communications providers to invest in new technologies and meet changing customers' demands (Gent, 2021).

Bypass fraud costs telecom firms $6 billion each year and is the world's second most expensive fraud. Figure **2.4** depicts the top three types of fraud and their annual losses. The figures are staggering, as major mobile operators have the potential to harm the country's GDP. SIMbox fraud is also one of the most significant retail frauds, accounting for 10-15% of total losses in the telecommunications industry globally (Airn, 2018). Unquestionably, fraud has grown to be a significant contributor to the mobile telecommunications industry's huge annual revenue losses. If each network operator does not implement enough fraud detection technology, subscribers' trust in the security of transactions made possible by the service provider would decline (Ibrahim & Saidu, 2015).

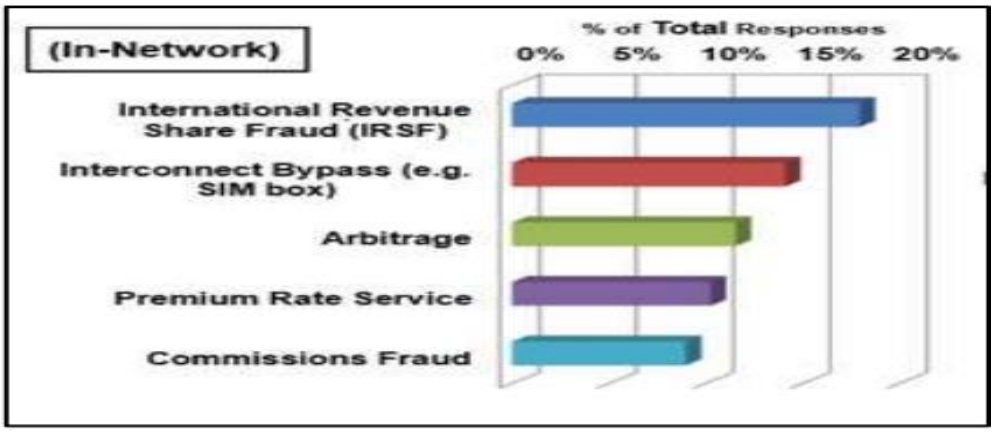

Figure 2.4 Fraud Responses as per CFCA Report (Airn, 2018)**.**

2.8 Deep Learning Algorithms

2.8.1 Introduction

Deep learning is a shining branch of machine learning that is based on learning levels of representations, which correspond to a hierarchy of features, factors, or concepts, where higher-level concepts are defined from lower-level concepts, and the same lower-level concepts can help to define many higher-level concepts (Liu et al., 2015). The field of machine learning is entering a golden age, with deep learning gradually becoming the leader in this domain. To build computational models, deep learning employs multiple layers to represent data abstractions. Deep learning algorithms like generative adversarial networks, convolutional neural networks, and model transfers have completely transformed our understanding of information processing (Pouyanfar et al., 2018).

Deep learning algorithms will be integrated into a wide range of human-assistance applications soon. They are more likely to let a computer play with its assumptions. To incorporate artificial intelligence into a computer system, most deep learning algorithms mimic the neuron connections of the human brain. This improves operational speed and accuracy for a variety of critical tasks. Computers are now programmed to perform specific tasks. In recent years, researchers have attempted to imbue a computer machine with artificial intelligence so that it can analyze and work on its own. Many robotic applications use artificial intelligence to solve problems. For example, a line-following robot is programmed to recognize and analyze a line drawn in front of it. Similarly,

several medical application robots have been trained to recognize scanned images from regular images (G, 2021).

The deep learning algorithm requires input before it can be analyzed. Images, like human vision, are used in several ways to analyze applications. Similarly, numerical data, voice, digital signals, and sensor outputs are all considered inputs for deep learning algorithms. However, because the inputs are in a different format, the deep learning algorithm requires a proper preprocessing step to remove the noise. As a result of this, the algorithm's precision increases. The preprocessing step is used to mark the area of interest as well as remove noise. then, the work is concentrated on a single area with no interruptions (G, 2021). To effectively address the various cyber security issues, a variety of techniques can be used, including multilayer perceptron's, convolutional neural networks, recurrent neural networks or long short-term memory, self-organizing maps, auto-encoders, restricted Boltzmann machines, deep belief networks, generative adversarial networks, deep transfer learning, and deep reinforcement learning, as well as their ensembles and hybrid approaches (Ghillani, 2022).

2.9 Deep Neural Network

A deep Neural Network is made up of several (deep) layers of highly optimized algorithms and architectures. There are many Deep neural network algorithms, among them Convolutional Neural Network which has more application Image, and Recurrent Neural Network, much of the time it is used sequential and time series data. and Multilayer Perceptron which has multiple hidden neurons to learn data deeply by using its hidden neurons. A multilayer perceptron model is used to learn representations from data sets without the need for manually designed feature extractors. As the name implies, deep learning has more or deeper processing layers than shallow learning, which has fewer unit layers. The shift from shallow to deep learning architectures has enabled the mapping of more complex and non-linear functions that were previously inefficiently mapped with shallow architectures (Shrestha & Mahmood, 2019).

Because a single-layer network has practical limitations on the linear separable problem, the deep neural network was developed to solve any classification problem. It has one or more hidden layers, each of which contains computational nodes called hidden nodes. The depth of the model is defined by the number of hidden layers. Figure **2.5** depicts the topology of a deep neural network with two

hidden layers and an output layer. The input data is routed to the first hidden layer, and the outputs of that layer are routed to the second hidden layer, and so on. The input signal propagates layer by layer until it reaches the output layer because each layer receives the previous layer's output as an input. Neurons in the output layer generate and propagate an error signal backward through the network (Lu, 2017).

A group of stacked models (layers) makes up Deep Neural Networks (DNN), which learn several hidden representations hierarchically. Aspects of input samples that have been enhanced in higher-level representations are suppressed and are useful for discriminating. The performance of numerous tasks, including speech recognition, object detection, natural language processing, and pattern recognition, has been enhanced using deep learning models (Lecun et al., 2015; Schmidhuber, 2015).

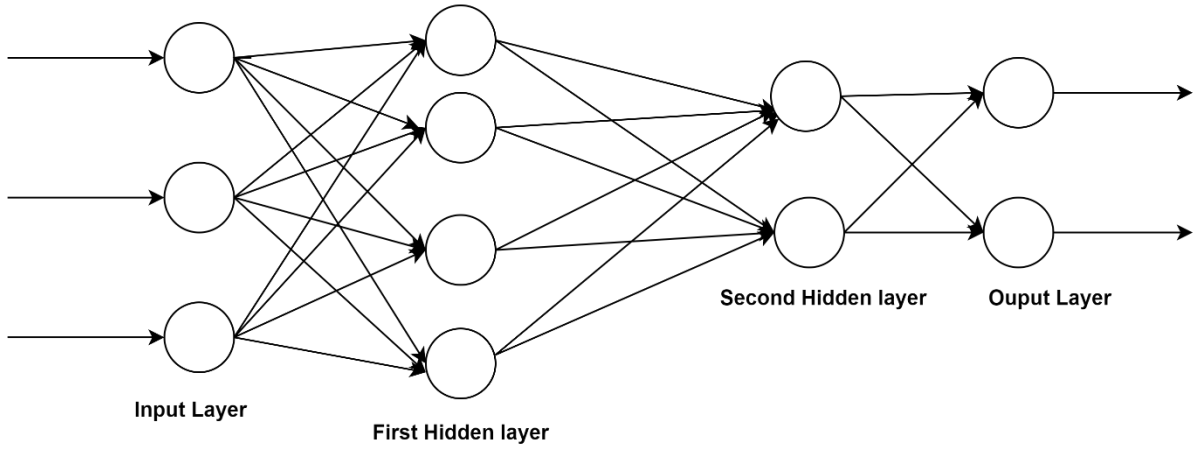

Figure 2.5 Structure of deep neural network with two hidden layers

Every neuron in the hidden and output layers performs two functions. One method is to compute the differentiable activation function on the input vectors and weights before passing the output through the hidden layers. In the second operation, which flows backward through the network, the gradient of the error with respect to weights connected to that neuron's inputs is computed (Lu, 2017).

2.9.1 Recurrent Neural Network

Recurrent Neural Networks (RNNs) are artificial neural networks having a sequential information structure (Ullah & Mahmoud, 2022). Recurrent connections in a neural network RNN are a type of deep neural network that can learn features and long-term dependencies from time series and sequential data. RNNs consist of a stack of nonlinear units connected by at least one directed cycle. A well-trained RNN may represent any dynamical system; nevertheless, training RNNs is usually hampered by challenges with learning long-term dependencies (Salehinejad et al., 2017). Recurrent neural networks are ANNs with recurrent connections that can model sequential data for sequence recognition and prediction (Deng & Yu, 2013).

A class of neural networks called recurrent neural networks (RNNs) is well suited to handling time-series data and other sequential data (DiPietro & Hager, 2019). Because of their ability to learn sequential dependencies, RNNs have gained popularity in applications such as speech recognition, speech synthesis, machine vision, and video description generation. One of the most difficult challenges in training RNNs is learning long-term dependencies in data. It is caused by the large number of parameters that must be optimized over long periods during RNN training (Salehinejad et al., 2017).

Each unit in an RNN has a hidden state that is updated at each time step based on the current input and the previous time step's hidden state. This allows the network to keep track of the sequence's information over time and make predictions based on the entire sequence rather than just the current input.

RNNs can be implemented in several different architectures, including one-to-one, one-to-many, many-to-one, and many-to-many. A one-to-one RNN, for example, is used for binary classification, whereas a many-to-many RNN is used for sequence-to-sequence tasks like machine translation.

2.9.2 Long Short-Term Memory Network

Long short-term memory (LSTM) has revolutionized machine learning and neurocomputing. According to several online sources, this model has improved Google's speech recognition, machine translations on Google Translate, and Amazon's Alexa responses. As of 2017, Facebook was using this neural system to perform over 4 billion LSTM-based translations per day. Interestingly, until LSTM, Recurrent neural networks had demonstrated rather discrete

performance. One reason for this recurrent network's success is its ability to deal with the exploding/vanishing gradient problem, which is a difficult issue to overcome when training recurrent or very deep neural networks (Van Houdt et al., 2020). A recurrent neural network with a state memory and multilayer cell structure is known as a long short-term memory. (Smagulova & James, 2019). The neural network known as the long short-term memory network was created to enable long-term dependency of artificial neural networks (Li & Zhao, 2019).

Long Short-Term Memory is a Recurrent Neural Network type that was created to address the issue of vanishing gradients in traditional RNNs. It is a deep-learning model that has a wide range of applications. including language modeling, machine translation, speech recognition, and many others. A memory cell, an input gate, an output gate, and a forget gate comprise the LSTM architecture. The memory cell stores data throughout time, and the gates control the flow of data into and out of the memory cell. Because of this structure, the LSTM can retain information for a longer period, making it suited for activities that need the preservation of long-term dependencies.

2.9.3 Multilayer Perceptron

Artificial neural networks, or neural networks in general, are a subset of artificial intelligence. multilayer perceptron’s form one type of neural network. A multilayer perceptron is made up of a network of simple interconnected neurons, or nodes as depicted in Figure 2.6 that denote a nonlinear mapping between an input and an output vector. Weights and output signals that are a function of the sum of the node's inputs modified by a simple nonlinear transfer, or activation, function connect the nodes. The multilayer perceptron can approximate extremely nonlinear functions by superimposing many simple nonlinear transfer functions. If the transfer function is linear, the multilayer perceptron can only model linear functions. And it is used for classification by assigning output nodes. It is critical to recall that the typical goal of multilayer perceptron training is to obtain strong generalizations on unseen data. The multilayer perceptron has been demonstrated to be an effective prediction, function approximation, and classification tool. The practical advantages of a modeling system that can precisely reproduce any measurable relationship are enormous. The advantages of the multilayer perceptron approach are especially apparent in applications where a full theoretical model cannot be constructed, such as when dealing with non-linear systems (Gardner & Dorling, 1998).

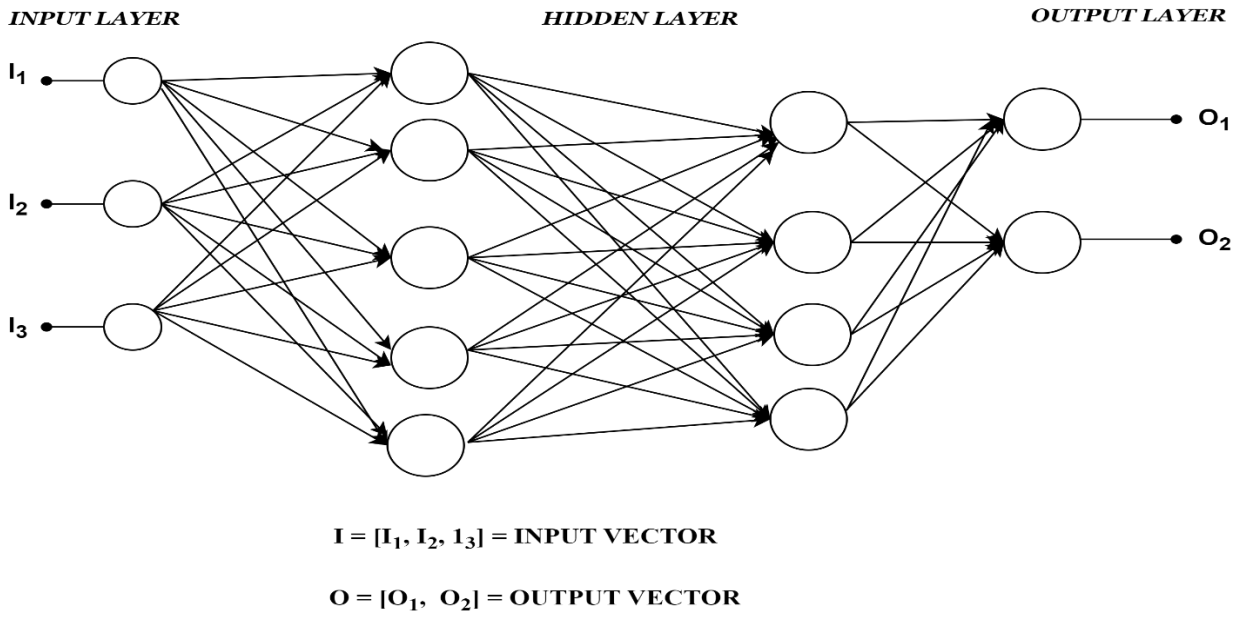

Figure 2.6 A Multilayer Perceptron with Two Hidden Layers

In an MLP neural network, the inputs of each perceptron are linked to the inputs of another perceptron from the previous layer. The perceptron will fire based on the defined threshold and the inputs' weighted sum. ANN/MLP should ideally be composed of three layers. The input layer, the hidden layer, and the output layer are all present. The input layer consists of nodes representing the number of system variables. Information travels from the hidden layers to the output layers. The flow is controlled by weight factors associated with each connector. The output layer nodes represent the system's classification decision. To determine the prediction output and classify each input, The output node values are compared to the limits (Karunathilaka, 2020).

Artificial neural networks emerged from the way biological neurons in the brain work. A biological neuron will fire a potential action if the cumulative input of the signals arriving exceeds a certain threshold, represented as Θ. This threshold however varies around an average value and is not the same for every neuron so it is uncertain if a neuron is doing what is expected. The firing thresholds are being updated continuously, which is the key factor for the adaptive learning abilities of the network. The same is true for a single-layer artificial perceptron which can only solve linear problems. In terms of both classification and regression, the Multilayer Perception (MLP) network design may be the most often used one today. Feed-forward neural networks called MLPs are

frequently trained using back propagation and typically consist of several layers of nodes with unidirectional connections (Informatik & Joachims, 1998; Osowski et al., 2004).

$$y = \sum_{i=1}^{n} W_i * X_i \quad \text{(2.1)}$$

Where n represents the number of inputs corresponding to the features. The weights per input are denoted to Wi and Xi is the input data. The perceptron then translates the inputs to an output signal, for the threshold, using a transfer function. For example, the output will be 1 (firing state) when $\sum_{i=0}^{n} \mathbf{Wi} \times \mathbf{Xi} > \Theta$**.** Commonly used examples of these functions are Unit Step, Sigmoid, or Gaussian. The weights determine the slope of the transfer function and the Bias allows shifting the transfer function horizontally along the axis while leaving the curvature unaltered. Like the biological neurons, learning arises in updating Bias and weights to reduce the error rate. The perceptron weight adjustment is denoted by:

$$\Delta W = \mu * \delta * X \quad \text{(2.2)}$$

With $\mu < 1$ as the learning rate and δ = (predicted output − desired output) (Van Efferen & Ali-Eldin, 2017).

An MLP has the same structure as a single layer plus one or more hidden layers, with all nodes connecting each other between layers. Backpropagation is an algorithm used by the network to train itself. This supervised learning algorithm computes outputs first with a sigmoid function and then propagates errors backward. Each unit receives the amount of error generated in this manner, and the weights are adjusted accordingly. Backpropagation, in a nutshell, uses the output error to adjust the weights of inputs at the output error and then repeats this adjustment for the previous layers. The error in one of the output nodes is then denoted as: (Van Efferen & Ali-Eldin, 2017).

$$\delta_0 = \text{output} * (1-\text{output}) * (\text{expected}-\text{output}) \quad \text{(2.3)}$$

And the error rate for a node *h* in the hidden layer can be calculated as:

$$\delta_h = \text{output} * (1-\text{output}_h) * (W_{h-\delta_h}) \quad \text{(2.4)}$$

The training phase is the repeated application of predetermined input values to the network, which adjusts the weights. As this process iterates, the weight values will be optimized, and the error

function will be minimized. To validate the network's performance, it is tested against testing samples prepared at the start (Karunathilaka, 2020).

2.10 Activation Functions

A mathematical function that is applied to the output of each node in an artificial neural network to determine whether or not the node should be activated based on a threshold is known as an activation function. It translates the weighted total of the input into an output signal, which is subsequently utilized to create a prediction by the following layer of nodes. Nonlinearity is introduced into the network by activation functions, allowing it to learn complicated correlations between inputs and outputs. Deep neural networks have been used successfully to solve real-world complex problems in a wide range of emerging domains, and many more deep learning architectures have been developed to date. Deep learning architectures use activation functions to perform various calculations across the hidden and output layers of any specific deep learning architecture to accomplish these cutting-edge outcomes. The ability of AFs to improve pattern learning in data, thus automating the process of feature detection and justifying their use in the hidden layers of neural networks, as well as their utility for classification purposes across domains, is demonstrated (Nwankpa et al., 2018). Artificial neural networks typically have a fixed, non-linear activation function at each neuron (Agostinelli et al., 2015).

In artificial neural networks, Activation Functions are used to convert an input signal to an output signal, which is subsequently provided as input to the next layer of the stack. We calculate the sum of the products of the inputs and their corresponding weights in an artificial neural network, then apply an activation function to it to generate the output of that layer and feed it as the input to the next layer (Sharma et al., 2020). There are numerous activation functions available. The most frequently used activation functions are the ReLu, sign, linear, step, and sigmoid functions (Karunathilaka, 2020).

2.10.1 Sigmoid Function

As a non-linear function, it is the most commonly used activation function. It is a mathematical function that transforms numbers from 0 to 1 and has a distinctive "S"-shaped curve or sigmoid curve. It is defined as $\mathbf{f(x) = 1/1+e^{-x}}$ (Sharma et al., 2020)**.**

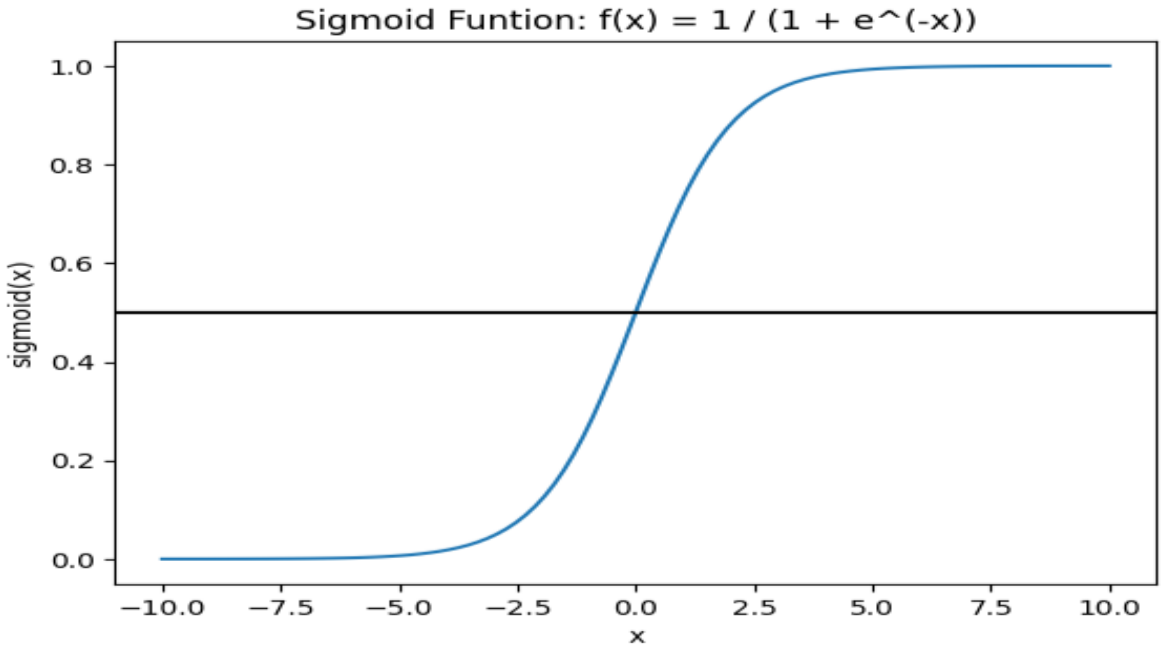

Figure 2.7 Sigmoid Function

2.10.2 ReLu Function

ReLu (rectified linear unit) is a non-linear is activation function that widely used in neural networks. The ReLu function has the advantage of not simultaneously stimulating all neurons. This signifies that a neuron will be silenced only if the linear transformation output is zero. It is mathematically defined as f(x) = max (0,x) (Sharma et al., 2020).

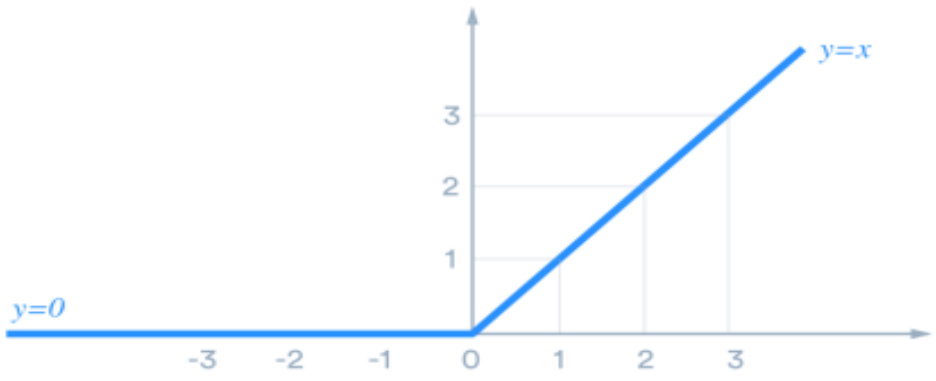

Figure 2.8 ReLu Visualization (Karunathilaka, 2020)

ReLU is linear (identity) for all positive values, and zero for all negative values. This means that:

- There is no difficult mathematics involved. As a result, it is simple to compute. This improves training or prediction performance.
- It accelerates convergence. Because the slope is linear, it does not plateau or "saturate" as x increases. Other activation functions, such as sigmoid or tanh, suffer from the vanishing gradient problem.
- It is only partially activated. Because ReLU is 0 for all negative inputs, any particular unit is unlikely to activate at all (Karunathilaka, 2020).

2.10.3 SoftMax

The SoftMax function restricts the output function to the range [0,1]. That is, the output can be understood as a probability. In other words, SoftMax functions are sigmoid functions with more classes, which implies they may determine the probability of many classes in the same way. In most neural networks, the SoftMax function is the final layer. The most significant statement is that SoftMax layers must have the same number as the output layer (Mercioni & Holban, 2020).

The SoftMax function is a sigmoid function combination. Because we know that a sigmoid function returns values in the range of 0 to 1, we can treat these as probabilities of data points from a specific class. Unlike sigmoid functions, which are utilized for binary classification, The SoftMax function can be used to solve multiclass classification problems. The probability is returned by the function for each data point in each of the individual classes (Sharma et al., 2020).

2.11 Related Works

To gain a thorough understanding of this research topic about fraud detection and classification, specifically, SIMbox/Bypass fraud detection, related literature such as journals, articles, research papers, and magazines were reviewed in addition to the Internet. Several scientific study papers and studies have been undertaken on telecom fraud detection and classification of telecom fraudsters' impact on telecom firms. SIMbox/Bypass fraud is one of the most typical telecom frauds that has had a significant impact on telecom companies. Several studies have been conducted to implement an intelligent system for detecting SIMbox/Bypass fraud, as stated below.

Kasra Babaei researched various aspects of the Detection and classification of fraud in telecommunications. It discusses various types of fraud, as well as their detection and correction. A fraud detection system's performance is highly dependent on the data for which it was designed.

As a result, investigating the performance of the same system on different datasets would be beneficial. There are numerous challenges that they face during fraud detection, including concept drift, imbalanced data distribution, the dimensionality curve, data limitations, noisy data, and the cost of real-time detection misclassification. Furthermore, they have mentioned the disadvantage of using a rule-based approach to identify bypass fraud because rule-based approaches are ineffective against new attacking patterns (Babaei et al., 2019).

Duygu Sinanç Terzi proposed big data analytics and the MapReduce parallel programming paradigm, to detect Telecom fraud, which provides simplicity and flexibility for large-scale applications. Finally, the model was applied to call detail records from a telecommunications company. The proposed method detects telecom fraud with 86% accuracy and is suitable for incorporation into a fraud management system for real-world implementation, according to the results. Many difficulties are phased, including non-disclosure agreement restrictions, dynamic patterns of fraud, data interval and frequency limitations, the difficulty of obtaining labeled data, regular distributions of both fraud and legitimate classes, the difficulty of analyzing data in distributed file systems, and an assessment of the success of the obtained results (Terzi et al., 2021).

The lossy counting algorithm, proposed by Bruno Veloso, Shazia Tabassum, Carlos Martins, Raphael Espanha, Raul Azevedo, and Joao Gama, is used to detect interconnect bypass fraud based on abnormal behaviors such as the occurrence of a burst of calls from specific numbers. Based on this assumption, they proposed using a new fast-forgetting technique in conjunction with using the Lossy Counting algorithm. They use frequent set mining to collect distributed patterns from various countries. They aim to Identify abnormal behaviors as soon as possible, such as bursts of phone calls, repetitions, mirrors, distributed behaviors, and a small number of calls spread across a large number of destination numbers. Lossy Counting, in contrast, is incapable of capturing distributed patterns (Veloso et al., 2020).

Other scholars, including Nassir Abuhamoud, Ibrahim Alsadi, and Salwa Ali, attempted to detect SIMBox/Bypass fraud using CDR files associated with the Almadar Aljadid operator. The CDR data were reviewed for four months using Neo4j technology to identify SIM cards used in SIMBox devices. In their study, they employed Neo4j technology to examine CDR information to identify the SIM cards used in SIMBox devices. The analysis was based on four data features such as users'

phone numbers making calls, but not receiving any, those receiving text messages but not sending any, numbers related to users making calls from a fixed location, and users' numbers whose calls exceeded the predefined duration. They have created many queries to discover abnormal behavior as the CDR records are imported into a histogram database. customer's behavior is considered abnormal if the user's behavior was significantly different from their regular behavior. According to those scholars, the following Table **2.1** depicts scenarios that could be considered abnormal user behavior (Abuhamoud et al., 2021).

Table 2.1 Abnormal Type and Description

Abnormal type	**Description**
Abnormal based on duration	Behavior is considered abnormal if a user makes calls with a duration longer than n minutes, where n is defined by the mobile operator company.
Abnormal based on Location.	In this scenario, we will check whether a user makes all his/her calls from a permanent location.
Abnormal based on messages	This means that we will check if any users are receiving SMS without sending any messages.
Abnormal based on calls	This means that we will check if any users are making calls without receiving any calls.

Because IMSI (International Mobile Subscriber Identity) and IMEI (International Mobile Equipment Identity) data have limitations for their study, they could conduct additional research to analyze user behavior based on the IMSI ratio to each IMEI. The lack of data for those features is retained as a limitation of this study (Abuhamoud et al., 2021).

Other scholars, Okumbor N. Anthony and Ateli A. Joy (Anthony & Joy, 2019) presented that, with the advancement of telecommunications and technology, traditional types of fraud have been replaced with more complex types of fraud known as bypass fraud, which occurs when international traffic that should be routed through a valid international gateway is instead routed over VOIP. The SIMboxing framework, its operation, and fraud issues were all discussed. SIMbox/ bypass fraud has become a difficult challenge for telecom companies in some parts of Africa and

Asia, according to their findings. The inability to detect SIMbox/Bypass fraud using a traditional approach such as TCG (Test call generator) because fraudulent change their behavior as technology advances is stated as a limitation in this paper. Finally, they pointed out that recent developments involving SIMbox fraud have exacerbated telecoms' challenges. They must rely on advanced technologies to prevent these attacks because regulatory intervention is not an option. The approach should enhance the capabilities of traditional models by combining and integrating advances in artificial intelligence.

Another study (Raghavan & Gayar, 2019) compares multiple machine learning and deep learning models for fraudulent detection on different data sets. The primary goal of the study was intending to benchmark different machine learning approaches such as a k-nearest neighbor, random forest, and support vector machines, as well as deep learning methods such as autoencoders, convolutional neural networks, restricted Boltzmann machine (RBM), and deep belief networks (DBN). As many companies invest in new techniques to improve their bottom line, this could potentially aid practitioners and businesses in better understanding how different methods work on different types of datasets. Even though the paper guides the selection of appropriate machine learning and deep learning algorithms based on the provided data, it is limited to detecting fraud in supervised learning methods. KNN and Random Forest appear appealing and produce good results, but they do not perform well in dynamic environments. Because fraud trends evolve and are difficult to identify.

Another scholar Alae Chouiekh in a paper (Chouiekh & El Haj, 2018) stated that increased fraud activity can have a negative influence on revenue gains and service quality. As a result, telecom providers must build effective algorithms that detect and/or prevent such fraud early on. In their research, they applied deep learning algorithms to detect fraudsters in mobile communication. Fraud datasets from a real mobile communication carrier's customer details records were used, and learning features were retrieved and classed as a fraudulent or non-fraudulent event activity. Several tests were carried out to assess the performance of their suggested model. In their study, they compared the performance of a convolution neural network to various classic machine learning methods in predicting fraudulent occurrences in a mobile communication network. Their findings indicate that the DCNN architecture produced ideal results, exceeding SVM, RF, and GBC (Gradient boosting Classifier) algorithms with an accuracy of 82%. Their study is a good attempt

at solving fraud challenges in mobile telecommunications using deep learning architecture such as DCNN. However, it is worth noting that their methods of study did not include a benchmark of different CNN architectures.

Another researcher presented a machine learning technique to detect and classify SIMbox/Bypass Fraud in their article titled "Machine Learning-Based Approach for Identification of SIM Box Bypass Fraud in a Telecom Network Based on CDR Analysis: Case of a Fixed and Mobile Operator in Cameroon." They used three classification methods to identify this specific scam in their study: Random Forest, Support Vector Machine, and XGBoost. They compared the effectiveness of these various approaches and assessed the model using CDR data that was obtained from a Cameroonian operator's network. With 92% accuracy, the Random Forest method outperformed other algorithms. Limitations are maintained due to the unbalanced dataset and the dynamic nature of fraud that causes concept drift (Djomadji et al., 2023).

Recently, (F. Tesfaye, 2020) researched to detect and classify Bypass fraud in near-real time using users' CDR data and ML algorithms for Ethio Telecom. The SW aggregation mode is used with a 4-hour minimum timespan. The sliding Window (SW) aggregation method is employed with the supervised Machine Learning (ML) algorithm to produce a relevant dataset instance and minimize the detection delay to one hour. They have identified nine (9) data features that are collected from the Ethio Telecom company that is used for detecting SIMbox/Bypass fraud. Three supervised ML classifier algorithms, Random Forest (RF), Artificial Neural Network (ANN), and Support Vector Machine (SVM) were used, with the two validation techniques being 10-fold cross-validation and supplied test. In their performance evaluation, the RF (Random Forest) algorithm perform better compared with the other two algorithms model in the detection of SIMbox fraud with ten cross-fold and separate test data. Mainly, the RF algorithm with SW aggregation mode scores higher performance values on all evaluation metrics and validation techniques within an hour. SW concept works for near-real-time on RF algorithm with better detection performance. Generally, they got an overall accuracy for RF (Random Forest) with SW aggregation mode using ten cross-fold validation achieved 96.20% accuracy and 94.90.% accuracy achieved while using separate test data validation techniques. However, near-real-time detection necessitates up-to-date data.

Table **2.2** below summarizes the summary of related work in general.

Table 2.2 Summary of Related Work

Researcher/Author	**Case study**	**Contribution**	**Limitation**
(Babaei et al., 2019)	CDR data	They have discussed various types of fraud, their detection and correction, and their performance is greatly dependent on the data for which it was designed.	Imbalance data distribution, limitation of data, and Concept drift
(Terzi et al., 2021)	CDR data	Big data analytics to identify telecom fraud has been presented. and finally, the CDR of a telecoms company was applied to the model. The findings show that the suggested method has an accuracy of 86% in detecting telecom fraud and is acceptable for integration into a fraud management system for use in the real world.	dynamic patterns of fraud, limitation of data intervals, frequency, and the difficulty of obtaining labeled data.
(Veloso et al., 2020)	CDR data	They proposed combining a new fast-forgetting technique with the Lossy Counting algorithm based on abnormal behaviors such as a burst of calls from specific numbers.	Lossy Counting is unable to capture distributed patterns.
(Abuhamoud et al., 2021)	CDR data.	They proposed examining CDR files related to the Almadar Aljadid operator with Neo4j to detect SIMbox fraud. They employed four characteristics to	Limitation of data for those features such as IMEI and IMSI.

		detect fraudulent behavior. such as user numbers making calls but not receiving any, receiving text messages but not sending any, numbers associated with users making calls from a fixed place, and user numbers whose calls exceed the predefined duration	
(Anthony & Joy, 2019)	CDR data	They recognized that SIMbox fraud is a difficult problem for telecom companies in various parts of Africa and Asia. They further alleged that recent advances in SIMbox fraud had exacerbated telecom issues. They must use innovative technologies to avoid these attacks because there is no regulatory remedy.	The dynamic nature of fraudsters and these features cannot be captured by traditional TCG.
(Raghavan & Gayar, 2019)	CDR data	The focused to compare various machine learning and deep learning methodologies. They also examine various machine learning and deep learning models for fraudulent detection on various data sets.	In dynamic environments, KNN and Random Forest do not perform well. Because fraud patterns shift over time and are difficult to detect.

(Chouiekh & El Haj, 2018)	CDR data	Convolution neural networks and machine learning algorithms were proposed, and their effectiveness at detecting fraud in mobile communication networks was evaluated. The DCNN architecture outperforms the RF, SVM, and GBC algorithms, achieving an accuracy of 82%.	their methods of the study did not include a benchmark of different CNN architectures.
(Djomadji et al., 2023)	CDR data	Three machine learning (ML) algorithms Random Forest, Support Vector Machine, and XGBoost were proposed, and their effectiveness at detecting and classifying SIMbox/Bypass Fraud was assessed. The Random Forest method outperformed other algorithms, achieving 92% accuracy.	The imbalanced dataset and the dynamic nature of fraud that results in concept drift maintain limitations.
(F. Tesfaye, 2020)	CDR data	Detect and classify SIMBox fraud in near-real time using ML algorithms and user CDR data. The SW aggregation mode is used with a 4-hour minimum period. They attained an overall accuracy of 96.20% for RF with SW aggregation mode using ten cross-fold-validation and 94.90% accuracy when using separate test data validation techniques.	For near-real-time detection, up-to-date data is limited.

Finally, they attempted to detect and classify SIMbox/Bypass fraud using Bigdata analytics methods, map-reduce parallel programming approaches, lossy counting algorithms, Machine Learning Approaches, and Deep learning algorithms such as deep convolutional neural networks, as evidenced by their related work. However, related work on this specific problem encounters issues such as the dynamic nature of fraud that cannot be captured by TCG (Test Call Generator) as well as the FMS (Fraud Management System) approach, data limitation, the inflexibility of the aforementioned approach to accommodate change as the nature of fraud change over time, the limitation of up to date data for near real-time machine learning algorithm, and supervised learning algorithms loss of performance and difficulties in capturing the dynamic patterns of fraud. To address this issue, we proposed Deep Learning/ Deep Neural Networks models such as MLP, LSTM, and RNN that capture the complex scenario of fraud as they change their behavior and train the model with enough data to detect and classify calls as normal (non-fraudulent) calls from legitimate service providers or fraudulent (call originated from SIMbox device).

CHAPTER THREE

RESEARCH METHODOLOGY

3.1 Overview

This chapter provides the Research methodology, which discusses the research method, approach, data gathering procedures, research finding measures or metrics, testing and assessment process, and algorithm employed in more detail. To investigate explanatory research problems, we attempt to describe the methodologies or models that we utilize in the suggested solution to detect and classify SIMbox/Bypass Fraud. Then we proposed deep learning/deep neural network models MLP, LSTM, and RNN, and finally, we evaluated the model performance.

3.2 Overall Methodology of the Research

We used the Research methodology to propose the solution, which deals with data collection, data analysis, algorithm selection techniques, and interpretations to achieve the objectives and answer the research questions. It is the science of analyzing how research is conducted methodically. In the case of Ethio Telecom, we discussed research methods for detecting and classifying SIMbox/Bypass fraud. Furthermore, research the existing method to gain a thorough understanding of the literature review, problem identification, data preparation, preprocessing, feature selection, and testing methods.

We employed the procedures and approaches mentioned below to meet the research objectives and develop our research:

- First, after we identified the research area, we reviewed different materials from different sources. Such as research articles, journals, books, related works, and sites.
- Second, we have identified the problem to be solved in the identified area.
- Third, we have collected relevant data to solve the identified problem.
- Fourth, we have preprocessed the collected data.
- Fifth, we have identified the required parameters to model the solution and then design the proposed solution.

- Sixth, we have done analysis and evaluation through the proposed algorithms (Models) to get the result of the proposed solution.
- Finally, we have analyzed the results and evaluated the performance of the proposed method based on our metrics in comparison with the existing methods and we have reported the acquired result.

Figure **3.1** shows the overall methodology employed to carry out this thesis. The process begins by evaluating several publications and journal papers written in the domain of SIMbox/Bypass Fraud Detection and Classification and identifying gaps in relevant works done in the area. Following that, the problem to be addressed is formulated using the gaps identified. Next to that, data is collected using various methods, and pre-processing is performed on the collected data. Following that, the development and implementation methodologies are identified by researching literature, and the data is trained, tested, and validated. Finally, the findings are thoroughly discussed and reported, with the researcher's conclusion and recommendations stated.

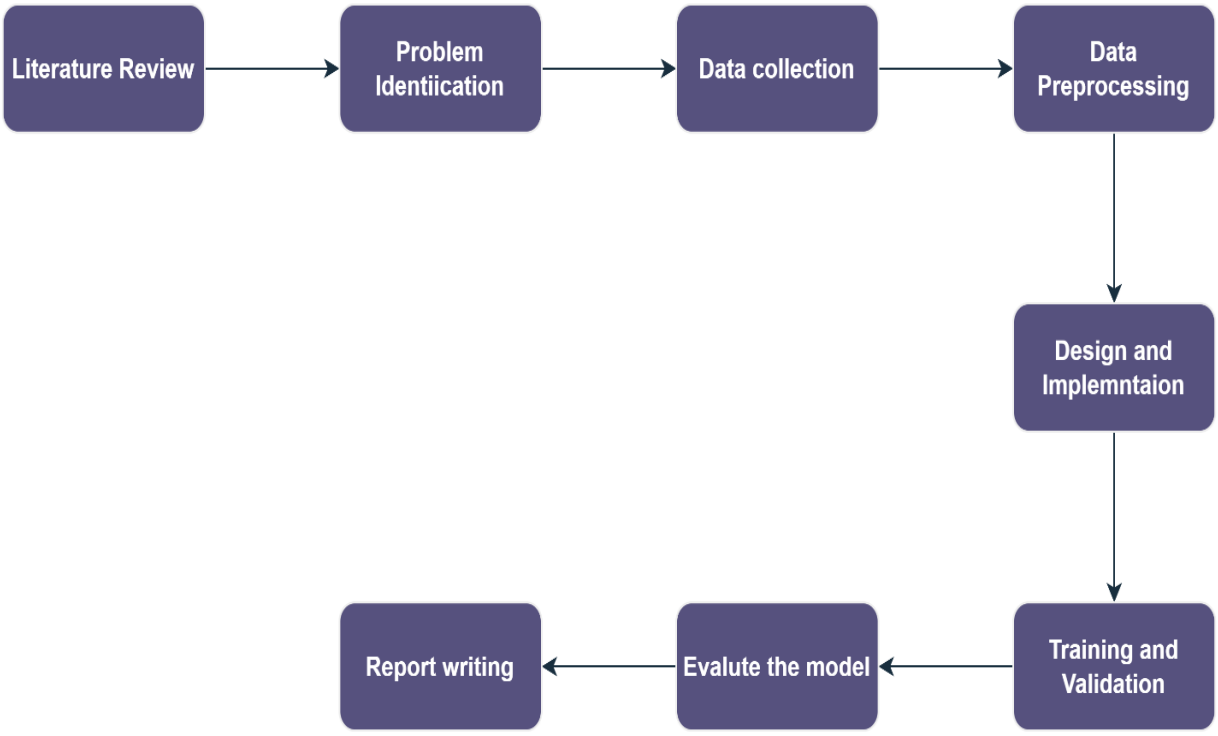

Figure 3.1 Overall Methodology of the Research

3.2.1 Literature Review

A thorough review of books, journal articles, and relevant documents from the Internet and other sources is conducted to comprehend the underlying principles and theories of the various approaches, techniques, and tools that can be used in Designing SIMbox/Bypass Fraud Detection. Investigate which Deep Learning model is utilized to solve the given problem, as well as how it has been employed in the previous related works. Furthermore, research works done both locally and internationally in the area of SIMbox/Bypass Fraud detection and Classification are explored since it allows the researcher to have a strong understanding of the problem area to support the results scientifically and to evaluate experimentally.

3.2.2 Problem Formulation

This section addresses shortcomings in existing SIMBox/Bypass Fraud Detection and Classification research and proposes potential solutions to the cited problem. The researcher concentrated on Detecting and Classifying SIMBox/Bypass Fraud by examining gaps in various literature. The primary goal of this research is to create a model that can detect and classify SIMBox/Bypass fraud. To do this, deep learning methods such as MLP, LSTM, and RNN are trained using CDR data collected from Ethio Telecom. The model is trained and assessed using performance-measuring metrics.

3.3 Data Collection

When we make a call through a telecom service provider, the call is always maintained on the telecommunications network, and descriptive information about the call is saved as call detail records. The dataset for this thesis was obtained from Ethio Telecom. The information was gathered from various mobile operator databases, including Cell ID, Data containing Loan information, and CDR (Usage Data). Even though we collected data from three different categories, we focused on CDR, which contains a subscriber's call history, and Cell ID, which is used to distinguish which unique BTS the calling party makes a call over a network.

For this thesis, subscriber usage history and CDR data from the Ethio telecom billing system were collected and prepared for the training. CDR generated from active customers' usage over two months was collected, stored in a database, processed, and outputted in the format required by the training.

When a subscriber makes a call over the operator's network, a toll ticket is generated that contains complete call information such as subscriber ID, called number, call duration, call start and end time, destination location, and so on. This information is specific to the call and is designated as CDR. CDRs are generated on a call basis by telephone switches or other telecommunication equipment and contain all of the information necessary to describe the important characteristics of a telephone call and other telecommunication transactions, whether from or to a subscriber in the network. It includes the fields required by billing systems to rate a specific call and bill the subscriber. CDRs detail each voice, SMS, and Internet data usage transaction that originates and terminates on a subscriber's device. A Field is a characteristic or feature of a CDR instance. It may include phone numbers involved in the call, the date and time of the call, the duration of the call, the identity of the cell transmitting the call to the subscriber's phone, and other information. For data preprocessing to be successful, it is necessary to delve deeper into the meaning of the records to gain a comprehensive understanding of the data (Hagos, 2018).

3.3.1 Telecommunication Data

Telecommunication data is digital information transmitted via a telecommunications network. Telecommunication data can consist of a wide variety of digital information, including voice, video, text, images, metadata, and other types of data. It is sent via a variety of communication protocols and technologies, including wired and wireless networks, satellite links, and microwave links**.**

Telecom companies generate and store enormous amounts of data. These data include call detail data (CDR), which describes the calls that travel over telecommunication networks, network data, which describes the states of the network's hardware and software components, and customer data, which describes telecommunication customers (Weiss, 2005). as we have various types of telecommunication data, we focused on Call Detail Records. In addition to CDR data also there is ISP (internet service provider) data which contains the internet usage data of customers. Because ISP data is largely used by ISPs for network maintenance, billing, and ensuring the smooth functioning of their services, they lack complete information for the detection and classification of SIMBox/Bypass Fraud because they lack voice usage data and CELL ID data which is critical in SIMBox/Bypass Fraud Detection and Classification.

3.3.2 Call Detail Record

When a call is placed on a telecommunications network, the call's descriptive information is saved as a call detail record. The number of call detail records generated and stored is enormous. Call detail records contain enough details to describe the important aspects of each call. Each call detail record will include, at a minimum, the originating and terminating phone numbers, the date and time of the call and the duration of the call, the number of SMS (Text) and data (Internet) usage of each subscriber within the payment information of each service (Weiss, 2005).

3.3.3 Understanding the Data

We obtained three categories of data from Ethio Telecom: Cell ID, Loan History, and Usage History. We focused on usage data, which includes the history of the customer's voice calls internet (Data) usage, and SMS. (Text data). These types of data are critical for detecting and classifying SIMBox/Bypass Fraud. In addition to the usage data, we have included the calling party's CELL ID.

Even though we have more than 20 columns or data features from usage data and CELL_ID data categories not all of them are equally important for our research, so we have chosen the most important features to detect and classify SIMBox/Bypass Fraud. While identifying those data features, we focused on the features that are most likely to answer our research questions about detecting and classifying SIMBox/Bypass Fraud. So, we chose twelve (12) basic data features that contain more information to clearly show the fraudulent activities. From those twelve features calling numbers and called numbers are hashed Because they contain sensitive information about subscribers.

3.4 Datasets and Their Descriptions

We obtained a two-month CDR data from Ethio Telecom for our research, which is quite large. After the data request was approved by the security department of Ethio Telecom Head Office, the company's domain expert extracted a variety of data from the Ethio Telecom Data server. There are around 39 columns in the data. Usage data (Customer data record) has 20 columns, loan data has 17 columns, and Cell ID has 2 columns. All of them have more than one million (1 million) rows of data, which is enormous and difficult to analyze using standard computers or those with low processor speeds.

The Following Table **3.1** summarizes the usage data (CDR) data with their descriptions.

Table 3.1 CDR Data Descriptions

No	***Data Features***	***Description***
1	DATA_DAY	attribute in the CDR that contains the date and possibly the time of the call or message.
2	SUBSTR (T. SERV_NO)	refers to the service number or phone number associated with the call.
3	SGMT_NAME	refers to the segment name or identifier associated with a particular portion or segment of the call.
4	CUST_TYPE_NAME	refers to the customer type or category associated with the user who made or received the call or message.
5	NETWORK_TYPE	identifies the type of network used to carry a particular call in a telecommunications system.
6	ACTIVATE_DATE	indicates the date and time when a particular service or account was activated in a telecommunications system.
7	STATUS_NAME	indicates the current status of a call in a telecommunications system.
8	OFFER_NAME	identifies the specific service offer or package associated with a particular call in a telecommunications system.
9	OFF_PEAK_USG_MINUTE	refers to the amount of time (in minutes) that a user spends on a call during off-peak hours. i.e. (Holidays, weekends, and nighttime). During those times the need for a network is low and the rates are lower compared to peak hours.
10	OFF_PEAK_AMT_ETB	is a field or attribute in the CDR that contains the amount charged for calls made during off-peak hours in Ethiopian Birr.
11	PEAK_USG_MINUTE	refers to the maximum usage of a specific resource during a call or communication session.

12	PEAK_AMT_ETB	is a field or attribute containing the amount charged for calls made during Peak hours in Ethiopian Birr.
13	INTER_USAGE_MINUTE	is a measure of the duration of a call or communication session between two different networks or carriers, used for inter-carrier billing and settlement.
14	INTER_AMT_ETB	is a field or attribute in the CDR that contains the amount charged for calls made during inter_usage in Ethiopian Birr.
15	SMS_LOCAL_USAGE	refers to the local SMS (Short Message Service) usage recorded in the Call Detail Record (CDR) of a telecommunications network.
16	SMS_LOCAL_FEE_ETB	refers to the fee charged in Ethiopian Birr for sending or receiving local SMS messages within the same network or carrier, recorded in the Call Detail Record of a telecommunications network.
17	SMS_INTER_USAGE	it indicates whether an SMS message was sent from one network operator to another. Specifically, it indicates whether the SMS message crossed international borders and was routed through a gateway or interconnect operated by a different network operator.
18	SMS_INTER_FEE_ETB	is a field in a CDR that indicates the cost of an SMS message that crosses international borders and is routed through a gateway or interconnect operated by a different network operator, in Ethiopian Birr (ETB).
19	DATA_USAGE_MB	is a field in a CDR that indicates the amount of data used during a data session, such as browsing the internet or using a mobile application, in megabytes (MB)
20	DATA_REVENUE_ETB	indicates the revenue earned by a telecommunications service provider for the data services (Internet) used by a subscriber, in Ethiopian Birr (ETB).

In addition to this, CDR there are two other types of data. The first category of data contains the subscriber's loan information with total columns of 17 and a million rows of data, and the third category of data contains the Cell ID of each subscriber to identify which BTS each Customer belongs to, which has its contribution to detecting SIMbox/Bypass Fraud. The CELL ID is a one-of-a-kind identifier assigned to each cell tower to which the subscriber's phone is linked during the call. Telecom operators can detect potential cases of Bypass Fraud by analyzing CELL ID. For example, if a subscriber frequently makes international calls from a specific CELL ID that is not located in an area where international calling is common, this could indicate that the subscriber is using a bypass method to make those calls. Furthermore, if a group of subscribers is making calls from the same CELL ID consistently, it could be a sign of SIM box fraud, in which a group of SIM cards is used to make illegal calls via a bypass route.

3.5 Data Preparation and Analysis

3.5.1 Data Preparation

The process of cleaning, transforming, and organizing raw data into a format suitable for analysis is referred to as data preparation, also known as data preprocessing. This is an important phase in any data analysis or machine/Deep learning algorithm since the quality and applicability of the data has a direct impact on the accuracy and effectiveness of the analysis. Data preparation is a crucial step in neural network modeling for complex data analysis, and it has a significant impact on the success of a wide range of complex data analysis tasks (Lai, 2006).

Typical data preparation steps include removing duplicate or irrelevant data, handling missing values, converting data types, normalizing or scaling data, and encoding categorical variables. It could also include feature engineering, which is the process of generating new features from existing data to improve the performance of machine/Deep learning models. Data preparation is therefore critical in ensuring that the data used in analysis or machine/deep learning models are accurate, complete, and relevant for model training.

3.6 Data Preprocessing and Feature Selection

Feature selection, as a dimensionality reduction technique, aims to choose a small subset of the important characteristics from the original ones by deleting irrelevant, redundant, or noisy features. Feature selection usually leads to better learning performance, i.e., higher learning accuracy,

reduced processing cost, and better model interpretability. In general, irrelevant features are those that cannot be used to differentiate samples from different classes (supervised) or clusters. Taking out irrelevant features does not affect learning performance. Removing irrelevant information may aid in the learning of a better model, as irrelevant features can confound the learning system and create memory and compute inefficiencies (Wang et al., 2020).

When dealing with a large number of input features, a common method is to employ specialized strategies to lower the dimensionality of the original problem, which occasionally enhances learning performance. Dimensionality reduction approaches are usually classified into two types: feature selection and feature extraction. The main difference is that feature extraction combines the original features and creates a collection of new features, whereas feature selection picks a subset of the original characteristics (Los, 2011).

Dimensionality reduction is a popular technique for reducing dimensionality and is divided into feature extraction and feature selection. Both feature extraction and feature selection can improve performance, reduce computational complexity, build better generalization models, and reduce storage requirements. By merging the original feature space, feature extraction maps the original feature space to a new feature space with lower dimensionality. As a result, further analysis of new features is difficult because the transformed features obtained from feature extraction have no physical meaning. Feature selection, on the other hand, selects a subset of features from the original feature set. As a result, feature selection preserves the true meaning of each chosen feature, making it superior in terms of feature readability and interpretability (Wang et al., 2020).

Because we have different data features (Attributes), not all of them are equally important in detecting and classifying SIMbox/Bypass Fraud. We selected the features that are important for detecting bypass fraud from all of the data. Those features are the DATA_DAY attribute in the CDR that contains the date and possibly the time of the call or message, CALLING_NUMBER the subscriber number where the call originated, CALLED_NUMBER the called(receiver) number, OFF_PEAK_USG refers to the number of times (in minutes) that a user spends on a call during off-peak hours. i.e. (Holidays, weekends, and nighttime). During those times the need for a network is low and the rates are lower compared to peak hours, PEAK_USG refers to the maximum usage of a specific resource during a call or communication session, INTER_USAGE is a measure of the

duration of a call or communication session between two different networks or carriers, used for inter-carrier billing and settlement, CALL DURATION refers to the overall voice call duration, SMS_LOCAL_USAGE refers to the local SMS (Short Message Service) usage recorded in the Call Detail Record (CDR) of a telecommunications network, SMS_INTER_USAGE it indicates whether an SMS message was sent from one network operator to another. Specifically, it indicates whether the SMS message crossed international borders and was routed through a gateway or interconnect operated by a different network operator, Number of SMS(Text) data is the total number of text data of each subscriber. almost all of the Calls which are originated from the SIMBox device cannot have SMS rather those fraudulent which mimic human behaviors, to appear as legitimate customers, Traffic Up and Traffic Down (DATA USAGE) (both of them are the data which contain internet usage), similar to SMS most of the time fraudulent call haven't internet usage. Therefore, by using those features we can detect and classify the SIMBox/Bypass fraud.

Table 3.2 Selected Features from CDR

Sno	**Field**	**Description**
1	DATA_DAY	The date and time of the call or message
2	CALLING NUMBER	The subscriber number where the call is originated
3	CALLED NUMBER	The called(receiver) number
4	OFF_PEAK_USG	Call made during Holidays, weekends, and nighttime
5	PEAK_USG	maximum call during a communication session
6	INTER_USAGE	Measures of the duration of a call or communication session between two different networks or carriers.
7	CALL DURATION	The overall voice call duration
8	SMS_LOCAL_USAGE	refers to the local SMS usage recorded in the (CDR)
9	SMS_INTER_USAGE	SMS messages are sent from one network operator to another.
10	NUMBER OF SMS	Total Number of text messages
11	Cell_A(CELL_ID)	The caller cell_ID
12	DATA_USAGE	The total amount of Internet usage

3.6.1 Preprocessing Data

Because of their large size, real-world databases are prone to noise, missing values, and data inconsistency. These data sources are numerous and heterogeneous, and they must be properly processed to improve data quality (Han, 2016). A few data processing techniques can help improve data quality.

- **Data cleaning** is applied to eliminate noise and rectify discrepancies in data.
- **Data integration** is well, useful to merge data from multiple sources into coherent data.
- **Data reduction** is used to minimize the amount of the instance data by aggregating, removing redundant characteristics, or clustering.
- **Data transformations** are normalizing the data which is scaled to fall within a smaller range. These techniques are working together to improve their results.

3.6.2 Data Cleaning

The process of identifying and correcting errors, inconsistencies, and inaccuracies in raw data is known as data cleaning. This is an important step in the data preparation process because it ensures that the data is correct, complete, and consistent so that it can be used for analysis, modeling, and decision-making.

Data cleaning is a step in the process of preparing a relevant data source for Machine/Deep Learning. Remove or fill the data vacant values, maintain data consistency, remove noisy data values, and remove redundant values by keeping a single record to avoid biasing the Machine/Deep Learning. The data cleaning process takes time and requires close attention to avoid the creation of irrelevant data at the end. Because the collected data is stored in various tables or locations, all data cleaning activities are applied to all data sources. Ensure that the same columns in each table have the same size, data type, and format (F. Tesfaye, 2020).

3.6.3 Data Aggregation

The process of collecting and summarizing large amounts of data from multiple sources into a more manageable and meaningful format is known as data aggregation. It entails combining individual data points into a summary statistic or metric, such as an average, count, sum, or percentage. By reducing the complexity and volume of information. it is easier to analyze and understand the data.

It enables the speedy and efficient capture of insights that can be utilized to make informed decisions or spot trends and patterns.

Data aggregation is a type of data preprocessing task performed on collected CDR data to provide complete information about specific users. A single record found under the raw CDR shows the activities of a specific user; however, understanding user behavior with a single CDR record is difficult. A collective CDR record must be aggregated to provide a complete picture of user behavior (F. Tesfaye, 2020). While we are doing data aggregation, we need to be careful because in CDR Usage the number of legitimate calls and the number of fraudulent calls is not proportional. Therefore, by considering this, we need to aggregate the important data features (Attributes) that are used to detect and Classify SIMBox/Bypass fraud.

3.7 Algorithm Training

Following the completion of data preprocessing, feature selection, and aggregation, the next step is to perform training and build a classification model using the chosen algorithm. The deep learning classification technique is covered in detail in Chapter 2. The deep learning algorithm deep neural network models namely MLP, LSTM, and RNN are used to create the models, as we have discussed in the second chapter of this thesis. Two distinct training techniques are used to train the models for the proposed Deep Learning algorithms. K-fold cross-validation and Separate test data are used explicitly.

The most commonly used training method is **K-fold cross-validation**. It divides the instance dataset into k-equal parts or folds, after which the classifier algorithm is trained with k-1 folds and tested with the remaining fold. This process is repeated iteratively by changing the test fold from the first to the Kth fold. Finally, the cumulative average error of each training and testing result is provided (F. Tesfaye, 2020; Hagos, 2018; Han, 2016).

Separate test data is another model training technique. The datasets are divided into two parts: one for training and one for testing. There is no specific labeled dataset ratio of testing and training data, but the divided datasets should be sufficient for both training and testing. If there are enough datasets, the dataset can be split 50/50. Unless the training and testing processes are influenced by insufficient datasets(F. Tesfaye, 2020). For this research, we have used 80% for training and 20% for testing.

3.8 Performance Measures of Classification Algorithms

The primary purpose of this study is to use deep learning techniques to detect and classify SIMBox/Bypass fraud. Following the completion of data collection, the preprocessing task is handled, and model training and testing are continued. We must test the model using test data to evaluate the performance of the algorithms. We can evaluate a given deep-learning model using a variety of evaluation metrics. Confusion matrix, classification accuracy, F-measure, Recall, Root Mean Squared (RMS), and ROC curve are examples of common evaluation matrices.

3.8.1 Confusion Matrix

The confusion matrix is one method of measuring the performance of supervised machine/deep learning algorithms, as the name suggests because the model becomes confused between the two classes. It is a 2X2 matrix containing classification class labels True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN) values.

Table 3.3 Conceptual Confusion Matrix

	Class 'N'	***Class 'Y'***
Class 'N'	***TP***	***FN***
Class 'Y'	***FP***	***TN***

Where, **TP** Number of instances correctly classified as Normal (class 'N').

FP Number of instances that belongs to SIM-box fraudulent (class 'Y') but classified as normal (class 'N').

FN Number of instances that belongs to Normal (class 'N') but are classified as Fraudulent (class 'Y').

TN Number of instances correctly classified as SIM-box Fraudulent (class 'Y')

Correctly classified instances for both fraudulent and Normal (legitimate) are indicated by TN and TP respectively. On the other ways around, incorrectly classified instances of fraudulent and normal (legitimate) identified by FN and FP respectively(F. Tesfaye, 2020). Several researchers (Al-Naymat et al., 2016; Datta et al., 2015; Hagos, 2018) use confusion matrix and classification accuracy as common classification metrics.

3.8.2 Classification Accuracy

Classification accuracy measures the ratio of correctly classified (both normal class 'N' and Fraudulent class 'Y') for the overall dataset. the below mathematical shows the percentage of correctly classified instances.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \quad \text{(3.1)}$$

3.8.3 F-Measure

A harmonic mean of precision and recall is F-measure and is calculated as shown in the following mathematical Equation.

$$\text{F-Measure} = \frac{2 * Precision * Recall}{Precision + Recall} \quad \text{(3.2)}$$

3.8.4 Recall

measures the ratio of correctly classified instances as Normal (class 'N') divide by the sum of correctly and incorrectly classified instances of normal (class 'N') as (class 'N') and (class 'Y'). Recall for Fraudulent (class 'Y') computed the same way using the below-shown formula.

$$\text{Recall} = \frac{TP}{TP+FN} \quad \text{(3.3)}$$

3.8.5 Precision

Measures the ratio of accurately classified incidents as Normal (class 'N') divided by the sum of correctly classified instances of Normal (class 'N') as (class 'N') and incorrectly classified instances of Normal (class 'Y') as (class 'N'). Precision for Fraudulent (class 'Y') is computed the same way using the formula below. The higher precision suggests that the model has fewer miss-classified occurrences (lower FP).

$$\text{Precision} = \frac{TP}{TP+FP} \quad \text{(3.4)}$$

3.8.6 ROC Curve

The True Positive Rate (TPR) and False Positive Rate (FPR) are graphical representations of the ROC. On the X and Y axes, FPR and TPR are shown. The algorithm is regarded as perfect when

the ROC curves are too close to the top-left corner of the area. If, on the other hand, the ROC curves fall below the linear line (X=Y), the algorithm is classified as a low-level classifier.

3.9 Hardware and Software Tools Used

3.9.1 Software Tools Used

Word Processing Software: We utilized Microsoft Word and PowerPoint to write and format our thesis document and presentation.

Data Visualization Tools like draw.io, Excel, and Python libraries like Matplotlib were utilized to build visual representations of our data including charts, graphs, and plots.

Mendeley Desktop: is a simple reference management program that also functions as a social network for academics to identify relevant papers.

Google Scholar: We used literature search tools such as Google Scholar to find relevant academic publications and research articles.

3.9.2 Hardware Tools Used

To install and use the above tools, a personal computer with an internal hard disk of 1TB, 8GB of physical memory, processor intel® coreTM i7-7500U CPU @ 2.70GHz 2.90Hz, Windows 10 Pro, and system: 64-bit operating system is used.

3.9.3 Programming Language Tools

Python 3.9.0

To do data analysis, preprocessing, training, and testing the deep learning model, python language is used. It includes several libraries and tools that make it well-suited for detecting and classifying SIMbox/Bypass Fraud. Pandas, NumPy, and Scikit-learn are some of Python's data analysis libraries. By analyzing enormous numbers of call data records generated by telecom networks, these libraries can be utilized to uncover patterns of fraudulent activity. TensorFlow, Keras, and PyTorch are among the libraries included, in addition to the ones listed above. These libraries can be used to develop models that detect and classify SIMbox/Bypass Fraud based on patterns and abnormalities in CDR data. The Python language's versatility, open-source nature, and built-in modules and tools make it well-suited for detecting and classifying SIMbox/Bypass Fraud, as well as other types of telecom fraud.

CHAPTER FOUR

PROPOSED SOLUTION

4.1 Chapter Overview

We presented a solution to the identified problem in this chapter. The study's major purpose is to detect and classify SIMBox/Bypass Fraud using deep learning/deep neural models such as MLP, LSTM, and RNN, and the model performance is measured using accuracy, precision, recall, F1-score, Confusion matrix, and Roc Curve. The most important step in any research is to propose a solution to the identified problems. This proposed solution is based on an optimized algorithm that considers the most efficient method of detecting and classifying SIMbox/Bypass fraud.

4.2 Proposed Model Architecture

We have proposed a deep learning/deep neural network, which allows computational models composed of multiple processing layers to learn data representations with several levels of abstraction. A deep-learning architecture is a multilayer stack of simple modules, all (or nearly all) of which are learnable and many of which compute nonlinear input-output mappings. Each module in the stack transforms its input to improve the selectivity and invariance of the representation. As technology advances, fraudulent activities' behavior changes from time to time. To combat this fraud, an algorithm that can identify and classify it as fraud is required, and a deep learning/deep neural network is proposed. DNN is capable of solving complex problems that necessitate the discovery of hidden patterns in data and/or a thorough understanding of intricate relationships among a significant number of interdependent variables. Deep learning neural networks are distinguished from other neural networks by their depth or number of hidden layers, which allows them to detect fraudulent activity in great detail(Shrestha & Mahmood, 2019).

While standard machine learning methods are linear, deep neural networks are built in an increasing hierarchy of complexity and abstraction. Each layer applies a nonlinear change to its input and generates a statistical model based on what it learns. Simply defined, the input layer receives the input layer and passes it onto the first hidden layer. These hidden layers execute mathematical computations on our inputs. One of the difficulties in developing neural networks is determining the number of hidden layers and the number of neurons in each layer. Each neuron has an activation function that is used to standardize the neuron's output. The term "deep" in Deep

Learning refers to having more than one hidden layer. The output data is returned by the output layer. Epochs are continued until the output has reached an acceptable level of Accuracy (Mandru et al., 2022).

Cybersecurity attacks are expanding at an exponential rate, rendering existing detection systems ineffective and emphasizing the need to develop more relevant detection models and approaches. This is still an ongoing research challenge because present attack detection methods are unable to keep up with the large quantity and variety of attacks. Researchers have recently focused on machine learning approaches, particularly deep learning techniques, due to their exceptionally high performance in a variety of detection-based disciplines(Ben Fredj et al., 2020).

In general, while we have various types of deep learning/Deep Neural Networks models, we have picked three DNN models, namely MLP, LSTM, and RNN, that are often used in cyber security applications due to their ability to handle and classify vast volumes of data. They can, for example, be used to detect anomalies or intrusions in network traffic, identify malicious nodes based on their behavior, and conduct other security-related duties.

As we have discussed the limitations of previous works in detecting and classifying the SIMbox/Bypass fraud the limitation of data, and the dynamic nature of fraud, which cannot be handled using either an FMS (Fraud Management system) approach or TCG (Test Call Generator), limitation of data for near real-time detection and classification, concept drift, and supervised machine learning algorithms show limitation to capture dynamic fraudulent behavior. As a result, to overcome these limitations, we proposed a deep learning algorithm/Deep neutral Network capable of understanding fraudulent behavior from CDR data. As we all know, data is the most important component of deep learning. We obtained CDR data from Ethio Telecom, determined the features that are crucial for training our model, and divided the data into training and testing. Following that, we used training data to train our model, which was then built utilizing deep learning/Deep Neural Network models such as MLP, LSTM, and RNN. Then we utilize testing data to see if our model determines the given data as fraudulent (call from a SIMbox device) or a legitimate call (call from a valid service provider).

Figure **4.1** below depicts the proposed solution for our research.

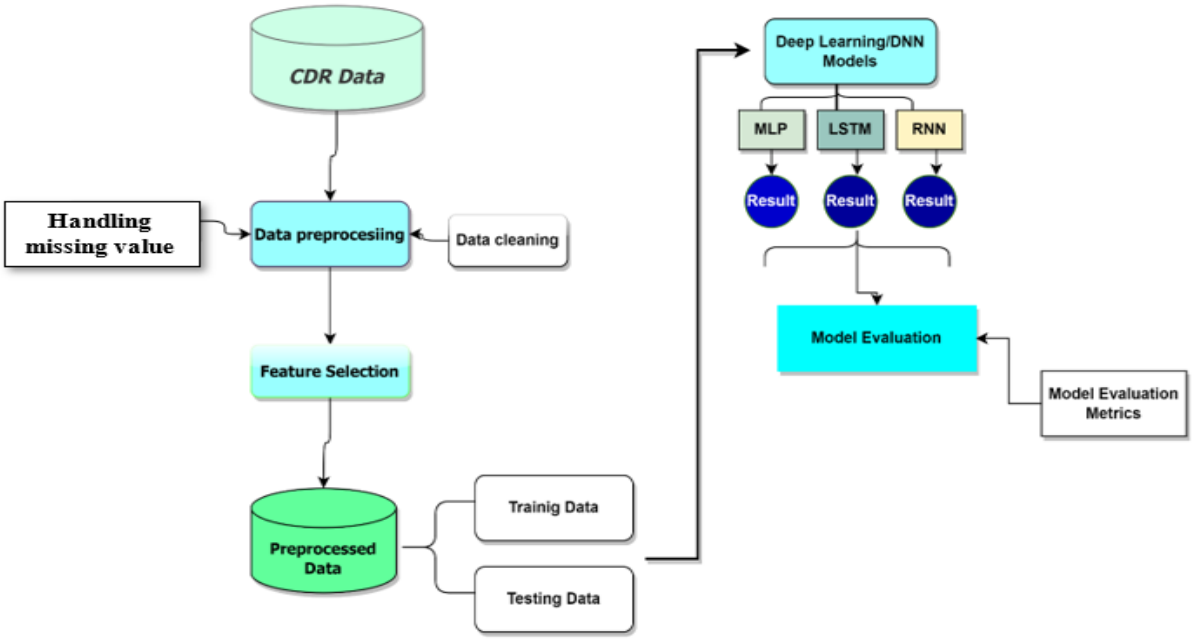

Figure 4.1 Proposed Solution

4.3 Proposed Data Preprocessing

Following the identification of important features, we prepared the data for deep learning algorithms by doing preprocessing, data cleaning, and training preparation. Following the completion of the data preparation work, we divided our data into training and testing data (for separate training and testing validation techniques) and the 10-cross-fold validation approach to compare the performance of our classification algorithms. As previously stated, we used the MLP, LSTM, and RNN Deep Learning/Deep neural Network Models to train our model, which is well-known for detecting and classifying deeply fraudulent behavior. After all, the fundamental purpose of this thesis is to detect and classify calls from SIMBox devices as fraudulent and calls from genuine service providers as legitimate based on the data provided to the model. Therefore, we have a target class known as 'FRAUD' with label 0(legitimate call) and label 1(fraudulent call) as follows:

- Fraudulent customer (1) and
- Legitimate (Non-fraudulent) (0) customer

As the collected two-month CDR data from Ethio Telecom is huge and complex to use as it is, we have applied feature selection and extraction to manage and make it suitable for training. The following Figure **4.2** shows the raw CDR file.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	DATA_DAY	SUBSTR(T	SGMT_NA	CUST_TYP	NETWORK	ACTIVATE	STATUS_N	OFFER_NA	OFF_PEAK	OFF_PEAK	PEAK_USG	PEAK_AM	INTER_US	INTER_AM	SMS_LOCA	SMS_LOCA	SMS_INTE	SMS_INTE	SMS_LOCA	DATA_USA	DATA_REVENUE_ETB	
2	20230102	9010000	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	46	3358	292	21316	0	0	3	3129	0	0	3129	3.79E+08	629015	
3	20230102	9010001	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	0	0	1203	0	0	0	16	0	0	0	0	96468992	0	
4	20230102	9010679	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	362	0	3096	0	0	0	0	0	0	0	0	5.61E+08	0	
5	20230102	9011380	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	0	0	0	0	0	0	0	0	0	0	0	0	0	
6	20230102	9012253	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	928	0	4487	0	0	0	4	0	0	0	0	1.82E+09	0	
7	20230102	9012343	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	85	6205	5133	374709	0	0	0	0	0	0	0	1.17E+08	194520	
8	20230102	9012370	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	0	0	315	0	0	0	0	0	0	0	0	8.4E+08	0	
9	20230102	9013064	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	0	0	0	0	0	0	0	0	0	0	0	0	0	
10	20230102	9014011	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	0	0	535	39055	0	0	0	0	0	0	0	0	0	
11	20230102	9015541	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	0	0	88	6424	0	0	0	0	0	0	0	3.26E+08	541310	
12	20230102	9016060	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	0	0	0	0	0	0	0	0	0	0	0	41891840	69410	
13	20230102	9017676	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	0	0	0	0	0	0	0	0	0	0	0	0	0	
14	20230102	9018568	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	0	0	0	0	0	0	0	0	0	0	0	0	0	
15	20230102	9019530	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	6	0	831	0	0	0	1	0	0	0	0	7.74E+08	0	
16	20230102	9020022	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	0	0	663	48399	0	0	1	1043	0	0	1043	76124160	126320	
17	20230102	9022004	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	0	0	3249	0	0	0	3	0	0	0	0	40779776	67670	
18	20230102	9023031	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	0	0	0	0	0	0	0	0	0	0	0	42991616	0	
19	20230102	9023536	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	0	0	289	0	0	0	9	0	0	0	0	5.23E+09	0	
20	20230102	9024818	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	340	0	3219	0	0	0	6	1740	0	0	1740	0	0	
21	20230102	9024834	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	0	0	1141	0	0	0	0	0	0	0	0	6.75E+08	0	
22	20230102	9025134	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	0	0	0	0	0	0	0	0	0	0	0	0	0	
23	20230102	9025200	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	7	0	3457	0	0	0	0	0	0	0	0	5.83E+08	0	
24	20230102	9027778	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	0	0	29	2117	0	0	0	0	0	0	0	0	0	
25	20230102	9028844	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	0	0	281	20513	0	0	0	0	0	0	0	4.03E+08	668630	
26	20230102	9030022	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	67	3417	176	12848	0	0	0	0	0	0	0	0	0	
27	20230102	9030133	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	304	17990	1204	87892	0	0	0	0	0	0	0	1.42E+09	2352745	
28	20230102	9030452	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	0	0	0	0	0	0	0	0	0	0	0	0	0	
29	20230102	9030486	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	0	0	0	0	0	0	0	0	0	0	0	0	0	
30	20230102	9030777	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	0	0	206	15038	0	0	0	0	0	0	0	0	0	
31	20230102	9030854	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	0	0	127	9271	0	0	0	0	0	0	0	2.19E+09	0	
32	20230102	9030954	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	0	0	0	0	0	0	0	0	0	0	0	0	0	
33	20230102	9031030	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	0	0	735	0	0	0	0	0	0	0	0	1.62E+09	0	
34	20230102	9031820	Postpaid	Enterprise	LTE	2.02E+13	Active	4G LTE Mo	39	0	2649	0	0	0	16	0	0	0	0	4.81E+08	0	

Figure 4.2 Raw CDR file

The collected CDR data from Ethio Telecom to detect and classify SIMBox/Bypass Fraud detection is so large, as illustrated in Figure **4.2** above, that it cannot be simply preprocessed with our standard personal computer. As a result, we used the filter approach to identify the most essential feature for our study. The filter approach will compute the importance of each feature and then rank them in order of importance. The main features to detect and classify SIMBox/Bypass fraud are chosen based on the filter method rank given to Features and our criteria.

The following Figure **4.3** shows the filter method we used for preprocessing and feature selection.

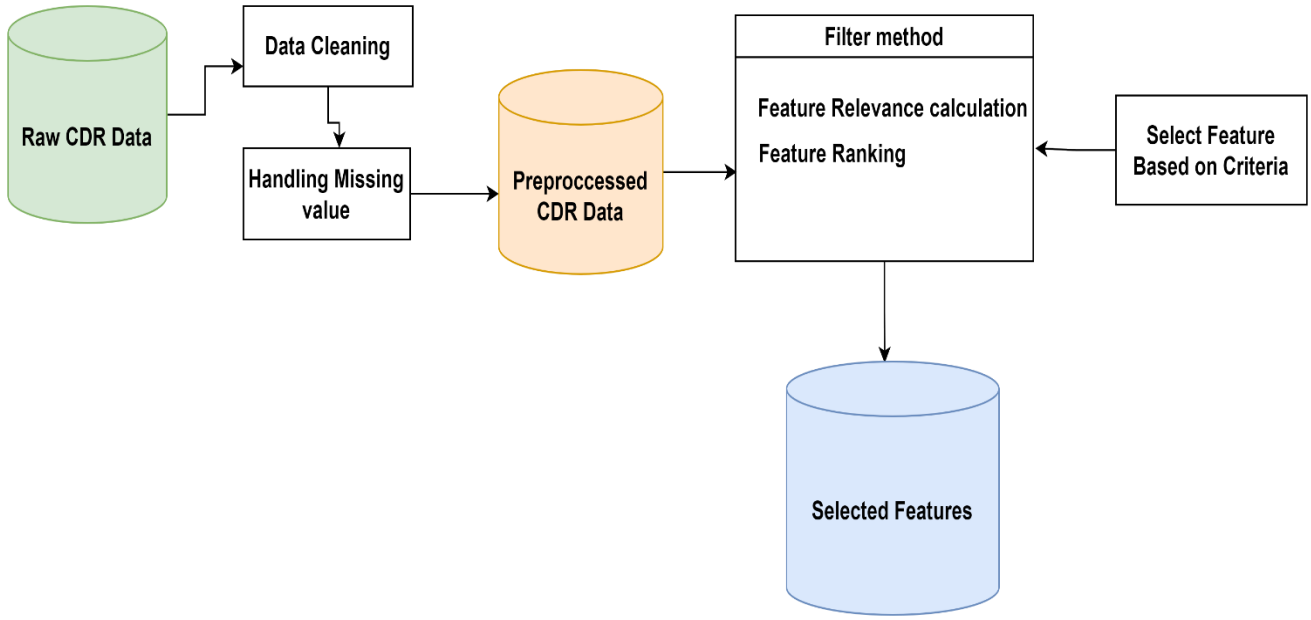

Figure 4.3 Proposed Data Preprocessing and Feature Selection

The raw CDR file that we obtained from Ethio Telecom has 20 columns (data features). However, not all of the data attributes gathered are equally essential for our study. As a result of using the filter method feature selection which is used to select important features based on their importance in a given case by ordering them in a rank. In addition to the filtering method, we have selected the important features which contain the usage data of the customer, which has potential information to identify the customer as legitimate or fraudulent. As a result, we identified 12 data features for our study. These features include each customer's usage throughout two months. These usage data contain the calling and called numbers, the voice usage data (both international and local calls per minute), the number of SMS used by each customer both locally and internationally, the data usage in megabytes, and the CELL ID, which contains the distinct BTS ID of each customer while making a call. This feature tells us whether a customer is making a call from the same location (belongs to the same BTS) or moving from one BTS to another, which is important to identify the fraudulent activity by overcoming their human behavior simulation to mimic a legitimate customer.

The following Figure **4.4** shows the selected features for our study.

	A	B	C	D	E	F	G	H	I	J	K	L
1	DATA_DA	CALLING_	CALLED_N	OFF_PEAK	PEAK_USC	INTER_US	CALL_DUR	SMS_LOCA	SMS_INTE	NUMBER_	CELL_ID	DATA_USAG
2	20230102	9006722	9189876	63	1574	0	1637	2	0	2	6.36E+14	29884416
3	20230102	9006832	9678765	200	2768	0	2968	1	0	1	6.36E+14	329625600
4	20230102	9010000	9976543	46	292	0	338	3	0	3	6.36E+14	379062272
5	20230102	9010001	9123456	0	1203	0	1203	16	0	16	6.36E+14	96468992
6	20230102	9010679	9654675	362	3096	0	3458	0	0	0	6.36E+14	560988160
7	20230102	9012253	9174221	928	4487	0	5415	4	0	4	6.36E+14	1824522240
8	20230102	9012343	9324760	85	5133	0	5218	0	0	0	6.36E+14	117223424
9	20230102	9012370	9234459	0	315	0	315	0	0	0	6.36E+14	839667712
10	20230102	9014011	9312215	0	535	0	535	0	0	0	6.36E+14	0
11	20230102	9015541	9765432	0	88	0	88	0	0	0	6.36E+14	326209536
12	20230102	9019530	9876543	6	831	0	837	1	0	1	6.36E+14	774373376
13	20230102	9020022	9876543	0	663	0	663	1	0	1	6.36E+14	76124160
14	20230102	9022004	9987654	0	3249	0	3249	3	0	3	6.36E+14	40779776
15	20230102	9023536	9124567	0	289	0	289	9	0	9	6.36E+14	5229772800
16	20230102	9024818	9134565	340	3219	0	3559	6	0	6	6.36E+14	0
17	20230102	9024834	9254321	0	1141	0	1141	0	0	0	6.36E+14	674758656
18	20230102	9025200	9218203	7	3457	0	3464	0	0	0	6.36E+14	583008256
19	20230102	9027778	9209876	0	29	0	29	0	0	0	6.36E+14	0
20	20230102	9028844	9234654	0	281	0	281	0	0	0	6.36E+14	402935808
21	20230102	9030022	9276543	67	176	0	243	0	0	0	6.36E+14	0
22	20230102	9030133	9910782	304	1204	0	1508	0	0	0	6.36E+14	1417832448
23	20230102	9030777	9341396	0	206	0	206	0	0	0	6.36E+14	0
24	20230102	9030854	9121788	0	127	0	127	0	0	0	6.36E+14	2190397440
25	20230102	9031030	9134678	0	735	0	735	0	0	0	6.36E+14	1621098496
26	20230102	9031820	9876543	39	2649	0	2688	16	0	16	6.36E+14	481296384
27	20230102	9031832	9176543	0	1884	0	1884	8	0	8	6.36E+14	0
28	20230102	9032000	9234398	56	1119	0	1175	6	0	6	6.36E+14	497987584
29	20230102	9033334	9974356	0	778	0	778	0	0	0	6.36E+14	1406976
30	20230102	9033887	9154321	46	301	0	347	0	0	0	6.36E+14	2956984320
31	20230102	9038511	9324567	248	1832	0	2080	0	0	0	6.36E+14	455606272
32	20230102	9038896	9121099	112	138	0	250	6	0	6	6.36E+14	1088880640
33	20230102	9039134	9874356	577	2113	0	2690	6	0	6	6.36E+14	0
34	20230102	9039833	9436443	0	1327	0	1327	0	0	0	6.36E+14	0

Figure 4.4 Selected Features from CDR

4.4 Proposed Models

Deep neural network models such as Multilayer Perceptron, Recurrent Neural Network, and Long Short-Term Memory architectures are used in the proposed model for SIMBox/Bypass fraud detection. With its completely connected layers and ReLU activation, the Multilayer Perceptron provides a robust framework for capturing complicated correlations between input data and the target variable. With its recurrent connections, the Recurrent Neural Network allows the model to include sequential dependencies in the data, such as the time patterns of fraud operations. A specific

type of RNN, Long Short-Term Memory, improves the model's ability to capture long-term dependencies by selectively preserving and forgetting information over time. The suggested model improves the detection and classification of fraud events by exploiting the capabilities of both architectures. For the proposed models we need to define the required parameters before the model training. Such as:

Batch size: the batch size defines how much data the model processes before changing the weights.

The number of epochs: During training, the number of epochs determines how many times the model iterates through the full training dataset.

Activation Function: Activation functions add nonlinearity to the model and aid in the capture of complicated patterns.

Optimizer: Based on the estimated gradients, the optimizer selects the algorithm used to update the model's weights.

Loss Function: the loss function computes the difference between the anticipated and true output, which serves as a measure of how well the model performs.

Metrics: Metrics are quantifiable metrics that reflect how well a model is working and can be used to compare models or tune their hyperparameters.

The following Table **4.1** summarizes the hyperparameters we used for the model training.

Table 4.1 Proposed Models with Parameter

Parameters	**Models**		
	MLP	**LSTM**	**RNN**
Batch size	32	32	32
Epochs	100	100	100
Activation Function	ReLu, Sigmoid	Sigmoid	Sigmoid
Optimizer	adam	adam	adam
Loss Function	categorical_crossentropy	categorical_crossentropy	categorical_crossentropy
Metrics	Accuracy	Accuracy	Accuracy

4.5 Implementation of the Proposed Solution

In this section, we discussed the implementation of the proposed model for the Detection and Classification of SIMBox/Bypass Fraud. We also discussed the configuration of the working environment in Python and Jupyter Notebook implementation.

4.5.1 Environment Setup

Python 3.9.0: was chosen as the programming language to implement the proposed method.

Jupyter Notebook: is the most popular and recommended integrated development environment (IDE) for academics to update Python code, visualize specific diagrams, train, and test charts, and track process progress. The suggested procedure is carried out using a Jupyter Notebook with a 6.5.2 version installed via Python installation.

Python libraries such as:

Pandas: Pandas are a powerful library for data manipulation and analysis.

NumPy: NumPy is a fundamental library for numerical computing in Python.

Matplotlib: Matplotlib is a Python charting toolkit that allows you to build static, animated, and interactive representations.

SciPy: SciPy is a NumPy-based library that provides additional scientific and technical computing capabilities**.**

Scikit-learn is a machine-learning library that provides a comprehensive set of tools for data mining, classification, regression, clustering, dimensionality reduction, and model selection. It offers a uniform interface for implementing and analyzing various machine learning and deep learning algorithms.

TensorFlow: TensorFlow is a prominent open-source deep learning package created by Google. It offers a complete ecosystem for developing and training machine learning models, particularly deep neural networks.

Keras: Keras is a high-level neural network API that can run on top of TensorFlow, Theano, or Microsoft Cognitive Toolkit (CNTK). It provides a user-friendly and straightforward interface for constructing and training deep learning models.

L2 Regularization (Ridge): L2 regularization modifies the loss function by adding a penalty proportional to the squared size of the model's weights.

4.6 Implantation of the proposed models

In this section, we covered the implementation of the proposed deep learning/deep neural network models, MLP, LSTM, and RNN to detect and classify SIMBox/Bypass Fraud. Before we proceed to the implementation of each model, we need to import the Python libraries which are required during our model training and load the CSV file from the local disk of the computer to the Jupyter notebook. the following Figure **4.5** depicts importing libraries and loading the CSV files to the Jupyter Notebook.

```
In [1]: import numpy as np
        import pandas as pd
        from sklearn.model_selection import train_test_split
        from sklearn.preprocessing import LabelEncoder
        from keras.models import Sequential
        from keras.layers import Dense, LSTM, SimpleRNN
        from keras.utils import np_utils
        from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
        from keras import regularizers

In [2]: # load dataset
        data = pd.read_csv('SIMBOX.csv')
```

Figure 4.5 Import Libraries and Loading CSV File

4.6.1 MLP Model Implementation

The Multilayer Perceptron is a feedforward neural network that is commonly used for deep learning tasks such as classification, regression, and pattern recognition. The MLP design typically includes an input layer, hidden layers, and an output layer. The first step in implementing an MLP is to design the architecture by specifying the number of input characteristics as well as the required number of hidden layers and neurons in each layer. The feature values are fed into the input layer, and each neuron in the hidden layers applies an activation function to the weighted sum of its inputs. The Rectified Linear Unit (ReLU), sigmoid, and SoftMax are all common activation functions in MLPs. The weights and biases of the neurons are iteratively modified during the training phase using optimization methods like gradient descent or its variants. Metrics including accuracy, precision, recall, and F1 score are commonly used to assess the performance of an MLP model. These metrics evaluate the model's ability to classify the input data accurately. Techniques

such as cross-validation and regularization can also be used to improve generalization and reduce overfitting.

After important libraries are imported and preprocessed data is loaded into Jupyter Notebook, we have defined the required parameter and trained our MLP model. The following Figure **4.6** shows the implementation of the MLP model from defining parameters to training the model.

```
In [5]: # define MLP model
        mlp_model = Sequential()
        mlp_model.add(Dense(128, input_dim=X.shape[1], activation='relu'))
        mlp_model.add(Dense(64, activation='relu'))
        mlp_model.add(Dense(64, activation='relu'))
        mlp_model.add(Dense(64, activation='relu'))
        mlp_model.add(Dense(y.shape[1], activation='sigmoid'))
        mlp_model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

```
In [8]: # train MLP model
        mlp_model.fit(X_train, y_train, epochs=100, batch_size=32, verbose=1)
        Epoch 92/100
        1620/1620 [==============================] - 2s 1ms/step - loss: 0.0452 - accuracy: 0.9876
        Epoch 93/100
        1620/1620 [==============================] - 2s 1ms/step - loss: 18.5897 - accuracy: 0.9866
        Epoch 94/100
        1620/1620 [==============================] - 2s 1ms/step - loss: 0.0449 - accuracy: 0.9872
        Epoch 95/100
        1620/1620 [==============================] - 2s 1ms/step - loss: 0.0454 - accuracy: 0.9875
        Epoch 96/100
        1620/1620 [==============================] - 2s 1ms/step - loss: 0.0454 - accuracy: 0.9863
        Epoch 97/100
        1620/1620 [==============================] - 2s 1ms/step - loss: 0.0445 - accuracy: 0.9874
        Epoch 98/100
        1620/1620 [==============================] - 2s 1ms/step - loss: 0.0449 - accuracy: 0.9868
        Epoch 99/100
        1620/1620 [==============================] - 2s 1ms/step - loss: 0.0442 - accuracy: 0.9870
        Epoch 100/100
        1620/1620 [==============================] - 2s 1ms/step - loss: 0.0447 - accuracy: 0.9868

Out[8]: <keras.callbacks.History at 0x297c1fd3460>
```

Figure 4.6 Implementation of the MLP Model

4.6.2 LSTM Model Implementation

Long Short-Term Memory is recurrent neural network architecture built primarily to handle sequential input with long-term dependencies. The first stage in implementing an LSTM is defining the architecture, which comprises the number of LSTM cells or units in the layer, the number of layers, and the input shape. Each LSTM unit has a memory cell that can keep information for lengthy periods, allowing it to capture sequence dependencies. The construction of LSTMs is distinct, consisting of three gates: the input gate, the forget gate, and the output gate. These gates control the flow of information by selecting which parts of the input to accept and which to discard or remember. The input gate selects which information to update, the forget gate selects which information to remove from the memory cell, and the output gate controls the cell's output. LSTMs learn by altering the weights of the connections between the LSTM units based on the gradient of the loss function throughout the training process. This is often accomplished through the use of optimization techniques such as Adam. The model is trained to minimize a loss function of your choice, such as mean squared error or categorical_crossentropy. Similar metrics to those used in MLPs can be used to evaluate the performance of an LSTM model, such as accuracy, precision, recall, and F1 score. To implement an LSTM, libraries or frameworks that provide RNN capabilities, such as Keras, TensorFlow, or PyTorch, are required.

Similarly, we trained an MLP model as well as the LSTM Model is implemented by defining the necessary parameters, as illustrated in Figure **4.7** below.

```
In [6]: # define LSTM model
        lstm_model = Sequential()
        lstm_model.add(LSTM(128, input_shape=(1, X.shape[1])))
        lstm_model.add(Dense(y.shape[1], activation='sigmoid'))
        lstm_model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

```
In [9]: # train LSTM model
        X_train_lstm = np.reshape(X_train, (X_train.shape[0], 1, X_train.shape[1]))
        lstm_model.fit(X_train_lstm, y_train, epochs=100, batch_size=32, verbose=1)
        Epoch 92/100
        1620/1620 [==============================] - 3s 2ms/step - loss: 0.0856 - accuracy: 0.9747
        Epoch 93/100
        1620/1620 [==============================] - 4s 2ms/step - loss: 0.0848 - accuracy: 0.9748
        Epoch 94/100
        1620/1620 [==============================] - 3s 2ms/step - loss: 0.0822 - accuracy: 0.9759
        Epoch 95/100
        1620/1620 [==============================] - 4s 2ms/step - loss: 0.0935 - accuracy: 0.9717
        Epoch 96/100
        1620/1620 [==============================] - 4s 2ms/step - loss: 0.0884 - accuracy: 0.9730
        Epoch 97/100
        1620/1620 [==============================] - 4s 2ms/step - loss: 0.0845 - accuracy: 0.9751
        Epoch 98/100
        1620/1620 [==============================] - 4s 2ms/step - loss: 0.0845 - accuracy: 0.9748
        Epoch 99/100
        1620/1620 [==============================] - 4s 2ms/step - loss: 0.0848 - accuracy: 0.9755
        Epoch 100/100
        1620/1620 [==============================] - 4s 2ms/step - loss: 0.0859 - accuracy: 0.9739

Out[9]: <keras.callbacks.History at 0x297c2269f40>
```

Figure 4.7 Implementation of the LSTM Model

4.6.3 RNN Model Implementation

The Recurrent Neural Network is a form of neural network architecture built primarily for processing sequential input by capturing dependencies and patterns across time. RNNs have a recurrent connection that allows data to be transmitted from one step in the sequence to the next. The initial stage in implementing an RNN is defining the architecture, which commonly involves one or more hidden layers of recurrent units, such as the Simple RNN or LSTM units. Each RNN hidden unit has a memory state that stores information from earlier steps in the sequence. The RNN learns by altering the weights of the connections between the units based on the gradient of the loss function during training. This is often accomplished through the use of optimization techniques such as Adam. The RNN goes through the input sequence one-time step at a time, with each time step representing a different point in the sequence. This enables the RNN to capture the data's temporal dependencies and patterns. Similar metrics to those used in MLPs and LSTMs can be

used to evaluate the performance of an RNN model, such as accuracy, precision, recall, and F1 score. RNN implementation necessitates the use of libraries or frameworks that provide RNN functionality, such as Keras, TensorFlow, or PyTorch.

In a similar case, we trained MLP and LSTM models as well as RNN models by varying their parameters. RNN is also one of the Deep neural network models chosen to detect and classify SIMBox/Bypass fraud. The implementation of the RNN model with its defined parameters is shown in Figure **4.8**.

```
In [11]: # define RNN model
         rnn_model = Sequential()
         rnn_model.add(SimpleRNN(128, input_shape=(1, X.shape[1])))
         rnn_model.add(Dense(y.shape[1], activation='sigmoid'))
         rnn_model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

```
In [26]: # train RNN model
         X_train_rnn = np.reshape(X_train, (X_train.shape[0], 1, X_train.shape[1]))
         rnn_model.fit(X_train_rnn, y_train, epochs=100, batch_size=32, verbose=1)
         Epoch 92/100
         1620/1620 [==============================] - 2s 2ms/step - loss: 0.0892 - accuracy: 0.9739
         Epoch 93/100
         1620/1620 [==============================] - 2s 1ms/step - loss: 0.0907 - accuracy: 0.9735
         Epoch 94/100
         1620/1620 [==============================] - 2s 2ms/step - loss: 0.0916 - accuracy: 0.9733
         Epoch 95/100
         1620/1620 [==============================] - 3s 2ms/step - loss: 0.0920 - accuracy: 0.9728
         Epoch 96/100
         1620/1620 [==============================] - 2s 2ms/step - loss: 0.0939 - accuracy: 0.9722
         Epoch 97/100
         1620/1620 [==============================] - 2s 2ms/step - loss: 0.0893 - accuracy: 0.9737
         Epoch 98/100
         1620/1620 [==============================] - 2s 1ms/step - loss: 0.0921 - accuracy: 0.9728
         Epoch 99/100
         1620/1620 [==============================] - 2s 2ms/step - loss: 0.0900 - accuracy: 0.9735
         Epoch 100/100
         1620/1620 [==============================] - 2s 2ms/step - loss: 0.0906 - accuracy: 0.9731
Out[26]: <keras.callbacks.History at 0x297c1ed0be0>
```

Figure 4.8 Implementation of the RNN Model

4.7 Implementation Evaluation Methods

We tested the model's performance using several evaluation metrics that are stated and discussed in Chapter three. Different metrics are devised to determine the Performance of our model, such as accuracy, precision, recall, F1 score, confusion matrix, and ROC Curve. Various performance measuring metrics can be used to assess the performance of the implemented models for SIMBox/Bypass fraud detection and classification. Accuracy, precision, recall, F1 score, confusion matrix, and ROC curve are the performance measures used to assess the deep learning/Deep neural network model for detecting and classifying SIMBox/Bypass Fraud. Precision measures the model's ability to correctly identify fraudulent calls, whereas accuracy measures the overall correctness of predictions. The F1 score combines precision and recall into a single metric that determines the model's ability to detect fraudulent behavior. The confusion matrix denotes the number of true positives, true negatives, false positives, and false negatives, and provides insight into the model's performance. At different classification thresholds, the ROC curve depicts the trade-off between sensitivity and specificity. These measures, when combined, give an in-depth evaluation of the deep neural network model's efficacy in identifying and classifying SIMBox/Bypass Fraud.

CHAPTER FIVE

RESULTS AND DISCUSSION

This chapter summarizes the research's experiment outcomes using 10-fold cross-validation and separate test data validation techniques. Three deep learning/deep neural network algorithms are used to detect and Classify SIMBox/Bypass fraud: such as MLP, LSTM, and RNN.

5.1 Model Performance Evaluation

The primary goal of this research is to detect and classify SIMbox/Bypass fraud using deep learning/deep neural network algorithms, and to compare the performance of each technique. As we described in earlier chapters, to overcome fraudulent actions, the Deep Learning technique is used to produce the necessary study outcomes. Experiments are carried out using the specified algorithms for the detection and classification of SIMbox/Bypass fraud. To experiment with supervised Deep Learning algorithms, 10-fold cross-validation, and Separate/Supplied test (80% for training and 20% for testing data) validation approaches are used. The model's final experiment findings are recorded, assessed, and compared to one another.

While comparing the models we need to consider and differentiate the models based on the training techniques such as 10-fold cross-validation and separate test data (80% for training and 20% for testing) techniques. MLP classifier algorithm achieved the highest accuracy of 99.17% than the other two classifier algorithms LSTM with an accuracy of 98.90% and RNN with an accuracy of 98.10% with Supplied test data training techniques. also, MLP beats the two models with an accuracy of 98.95% more than the other two classifier algorithms LSTM with an accuracy of 98.63%, and RNN with an accuracy of 98.42% with 10 cross-fold validation training techniques. A total of six independent models were built using those Deep learning/Deep neural network algorithms, on both training techniques 10-fold cross-validation and Separate test data all six models of Deep learning algorithm achieved nearly better performance.

Generally, the following Table **5.1** summarizes the performance of the classification algorithms by different performance measures metrics such as Accuracy, Precision, recall, and F1 score.

Table 5.1 Performance Measures of Classification Algorithms

Algorithm	**Performance measure metrics**							
	10-fold cross-validation				**Train test split data (80 training, 20 test)**			
	accuracy	**precision**	**recall**	**F1 score**	**accuracy**	**precision**	**recall**	**F1 score**
MLP	98.95%	99.20%	99.08%	99.14%	99.17%	99.31%	99.27%	99.29%
RNN	98.42%	98.80%	98.61%	98.70%	98.10%	98.70%	98.30%	98.50%
LSTM	98.63%	98.66%	98.36%	98.5%	98.90%	98.87%	98.74%	98.67%

As shown in Table **5.1** above, nearly all of the algorithms have higher accuracy, precision, recall, and F1 score. Even though they perform better in detecting and classifying SIMBox/Bypass fraud, MLP classifier achieved better accuracy in both validation techniques (10-fold cross-validation and train test split data), LSTM ranked second in both 10-fold validation technique and in train test split data validation technique, and RNN ranked third in both validation technique.

To evaluate the performance of the classification algorithm, we emphasized and accessed multiple performance measure metrics, since depending on the problem domain, we focused on performance measure metrics that reflect the model's ability to detect and classify SIMBox/Bypass Fraud. As previously indicated, we assessed the effectiveness of our classification model using performance measures such as accuracy, precision, recall, and F1 Score. In our approach, all metrics perform similarly. This demonstrates that the model has attained a high level of precision in classifying both positive and negative instances. It shows that the model is performing well in properly identifying the positive class while retaining a high level of accuracy in overall classifications. This demonstrated that our classification algorithms are capable of classifying calls from a valid service provider (Ethio telecom) as legitimate (non-fraud) calls and fraudulent (illegitimate) calls from SIMBox devices with greater accuracy by minimizing misclassified instances.

As accuracy, precision, recall, and F1 score have different values across our classification algorithms like MLP, LSTM, and RNN to detect and classify SIMBox/Bypass fraud, the following Figures **5.1** are captured during model training and testing using 10-fold cross-validation.

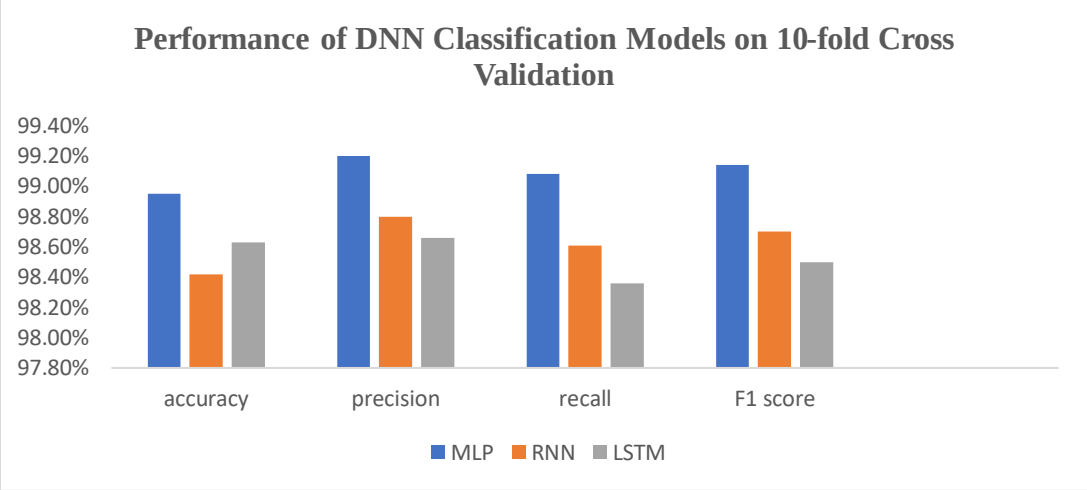

Figure 5.1 Performance of DNN Classification Models on 10-Fold Cross-Validation

As demonstrated in Figure **5.1** above, three classification algorithms are trained to detect and classify SIMBox/Bypass fraud using the 10-fold cross-validation technique. MLP has the highest accuracy and is ranked first, while LSTM and RNN are ranked second and third, respectively, based on their performance achievements. Similarly, we employed the 10-fold cross-validation technique to train our model, as well as separate train test split data techniques to train our classification algorithms to compare whether techniques are better for training and testing classification algorithms. Figure **5.2** below is also taken during the training and testing of our classification algorithms by measuring their performance using accuracy, precision, recall, and F1 score.

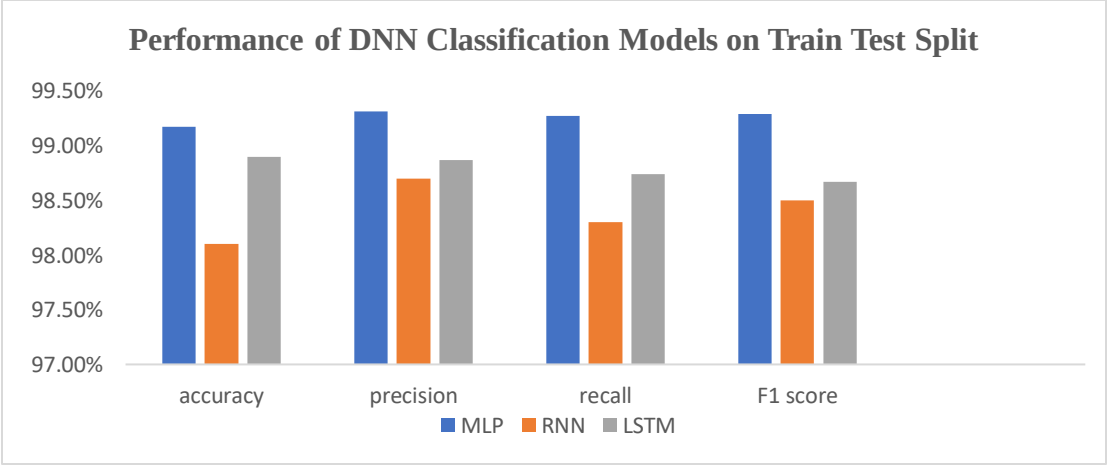

Figure 5.2 Performance DNN Classification Models on Train Test Split Data

As shown in the Figure **5.1** and **5.2** above, three classification algorithms are trained to detect and classify SIMBox/Bypass fraud utilizing the separate train test data validation technique (80% for training and 20% for testing). Based on their performance achievements, MLP has the highest accuracy and is ranked first, while LSTM and RNN are ranked second and third, respectively.

In addition to the performance measures described above, we used a confusion matrix to evaluate the performance of our classification algorithms. The confusion matrix depicts our model's performance by examining True Positives (TP) are positive instances that are correctly classified as positive, True Negatives (TN) are negative instances that are correctly classified as negative, False Positives (FP) are negative instances that are incorrectly classified as positive, and False Negatives (FN) are positive instances that are incorrectly classified as negative. The following figure shows the confusion metrics of our model using both 10-fold cross-validation and train test validation techniques.

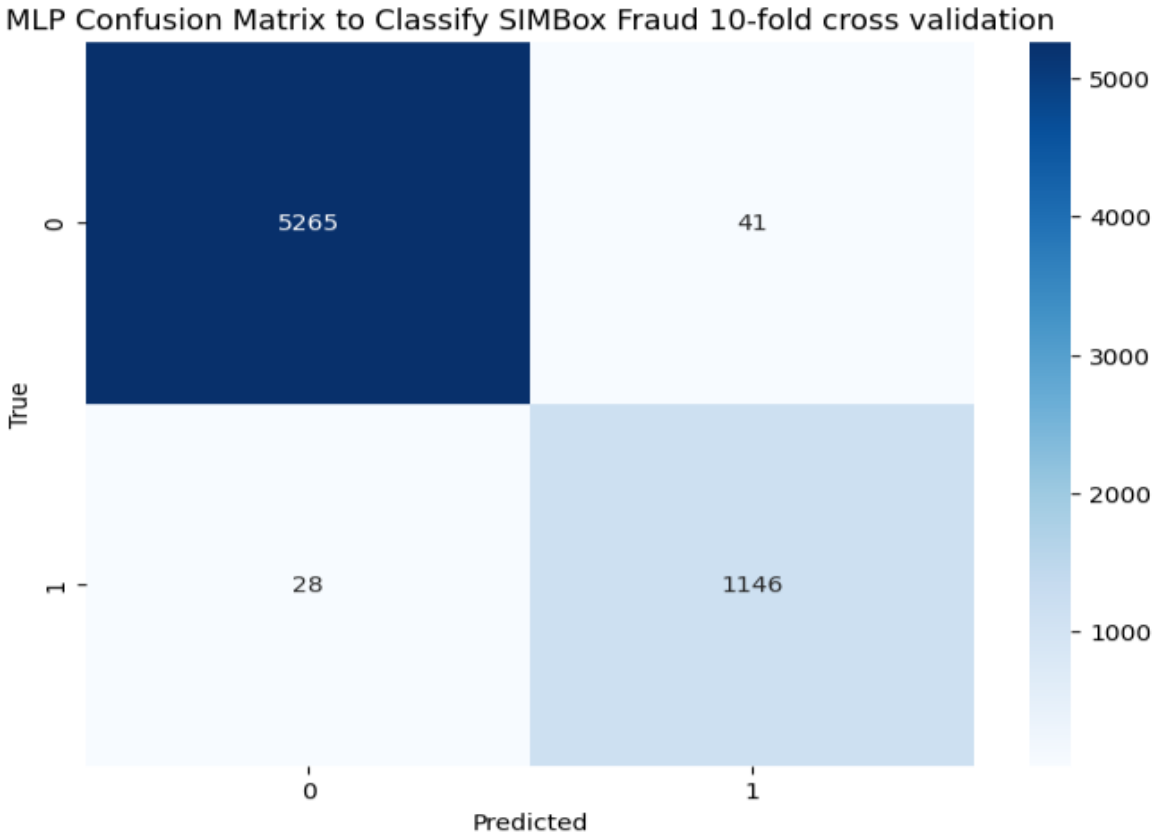

Figure 5.3 Confusion Matrix of MLP Model on 10-Fold Cross-Validation

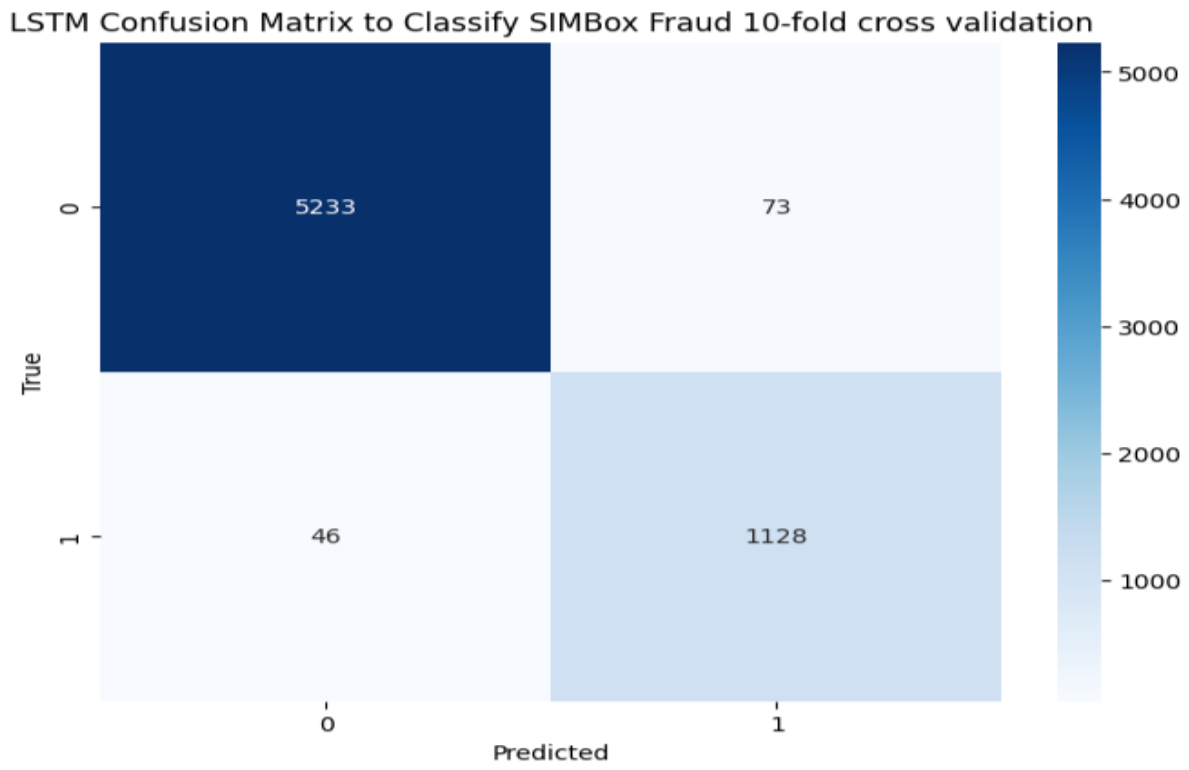

Figure 5.4 Confusion Matrix of LSTM Model on 10-Fold Cross-Validation

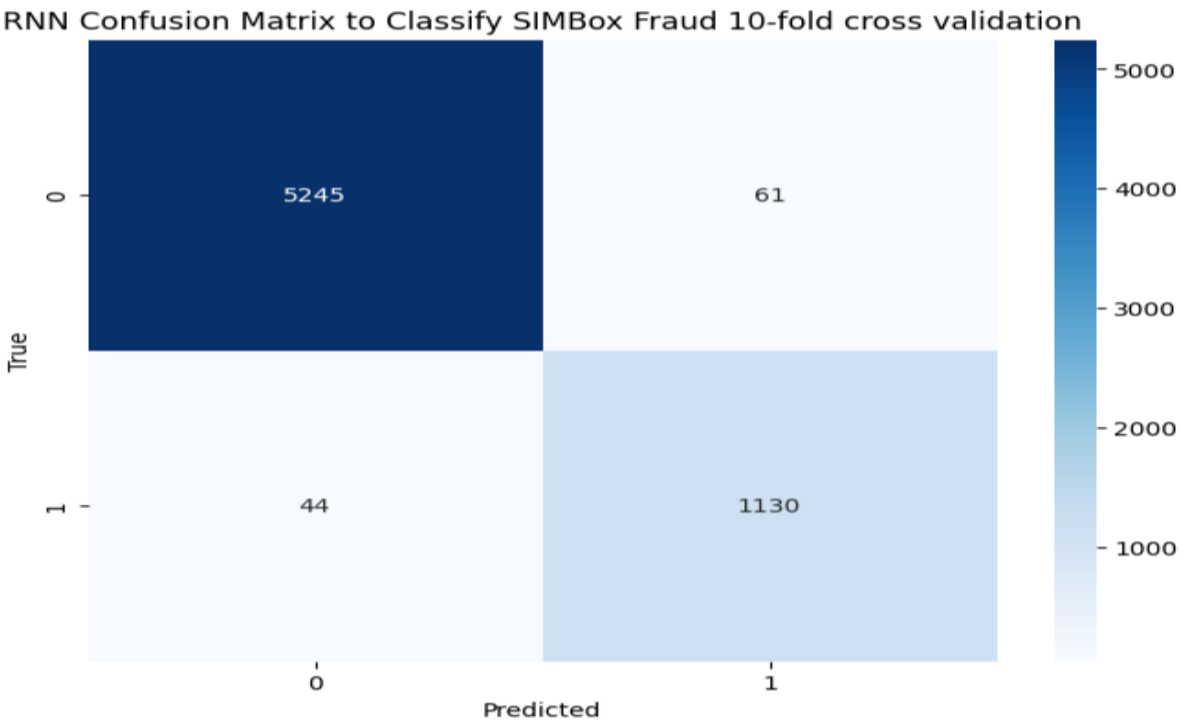

Figure 5.5 Confusion Metrix of RNN Model on 10-Fold Cross-Validation

As illustrated by the confusion matrix in Figures **5.3, 5.4,** and **5.5** above, our classification model has higher accuracy in 10-fold cross-validation, and the number of misclassified instances is quite small in comparison to the correctly classified instances. This indicates that the model is accurate in recognizing positive and negative instances. It suggests a solid mix of precision (ability to identify positive cases properly) and recall (ability to recognize all positive occurrences correctly).

In general, the confusion matrix of the 10-fold cross-validation technique of the classification model is summarized in Table **5.2** below.

Table 5.2 Confusion Matrix of 10-fold Cross-Validation

Confusion matrix			Number of Instances	Correctly classified	Incorrectly classified	Accuracy	model
X	Y	Classified as					
5265	41	X=legitimate	6480	6411	69	98.95%	MLP
28	1146	Y=fraudulent					
5233	73		6480	6361	119	98.63%	LSTM
46	1128						
5245	61		6480	6375	105	98.42%	RNN
44	1130						

As the confusion matrix provides useful insights into the model's performance, we can see from the above Table **5.2** that practically all of the classification models we've trained perform well in detecting and classifying SIMBox/Bypass fraud. When the number of FP and FN instances is low, it indicates that the model has made a few inaccurate classifications. A low number of false positives (FP) indicates that the model has a low rate of misclassifying negative instances as positive. Similarly, a low number of false negatives (FN) suggests a low rate of misclassifying positive instances as negative. When we compare the misclassified examples (FP and FN) to the correctly classified instances (TP and TN), the difference is relatively minor. In general, based on the provided confusion matrix performance measure applied on 10-fold cross-validation to detect and classify SIMBox/ Bypass fraud, the MLP achieves the best performance, followed by LSTM and RNN models.

As we described in earlier sections, we used model performance measure metrics such as accuracy, precision, recall, and F1 score to assess the performance of our classification algorithms. To be more certain about the model's performance in detecting and classifying SIMBox/Bypass Fraud, Confusion matrices must be used to obtain detailed insights into the classification model. The confusion matrix performance measure was applied to the train test split data in Figures **5.6**, **5.7,** and **5.8** below.

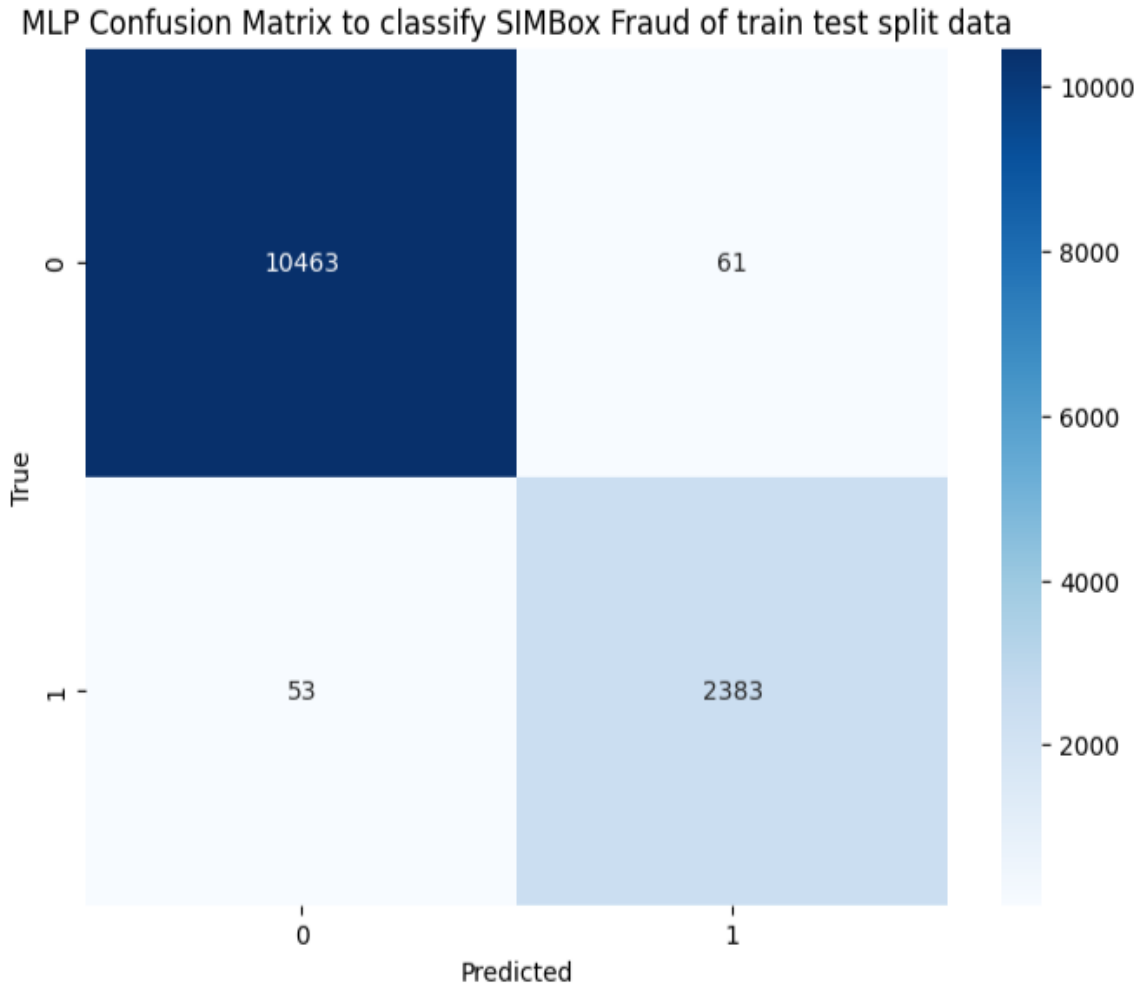

Figure 5.6 Confusion Matrix of MLP Model on Train Test Data

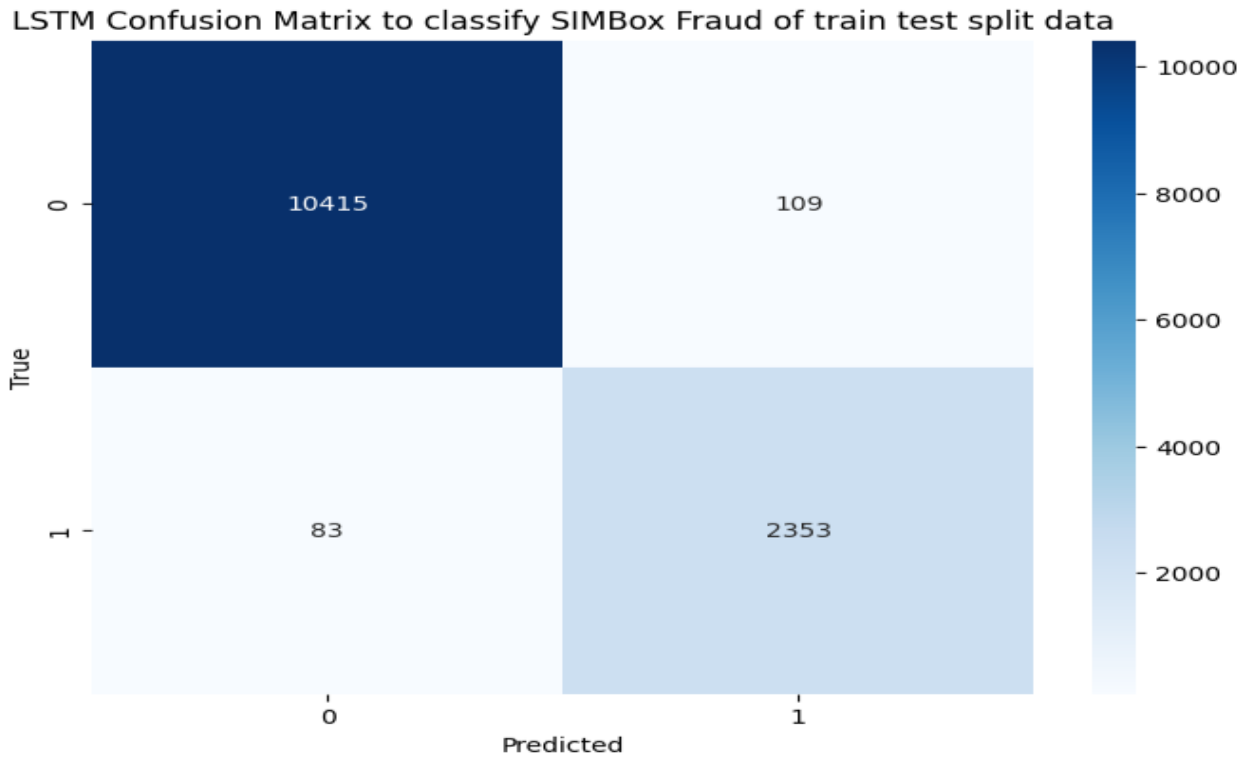

Figure 5.7 Confusion Matrix of LSTM Model on Train Test Data

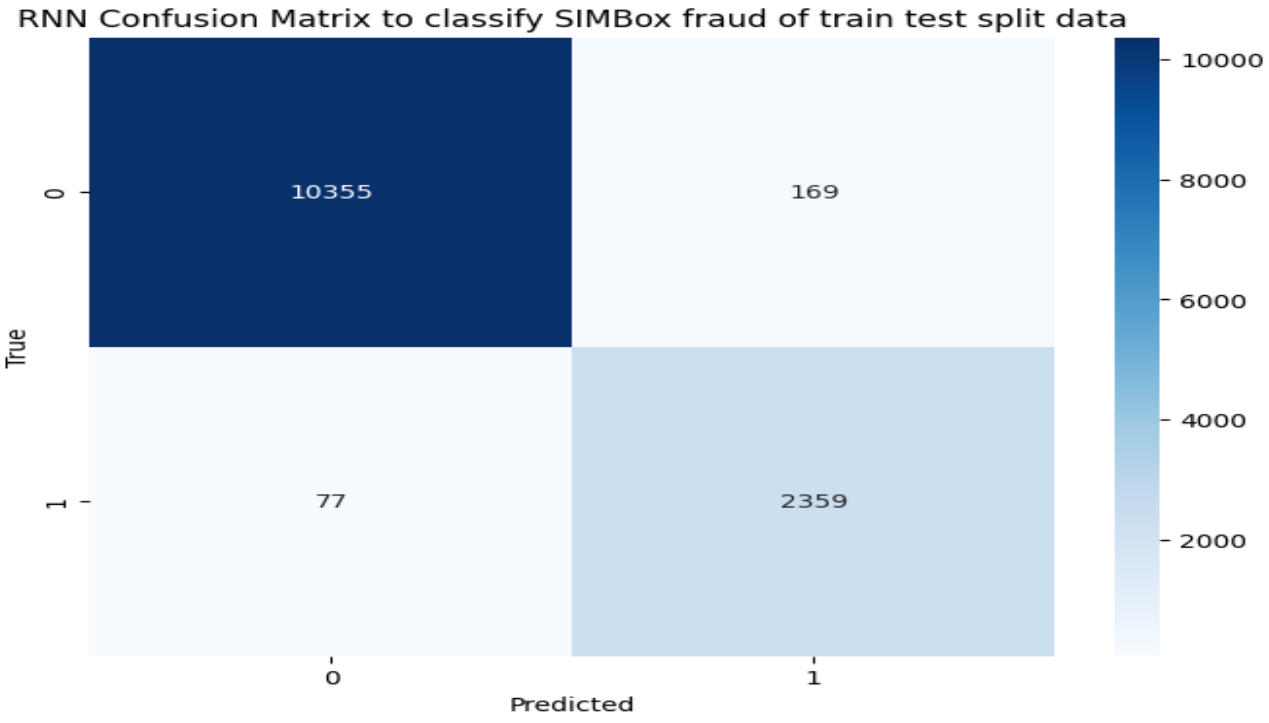

Figure 5.8 Confusion Matrix of RNN Model on Train Test Data

As shown in Figures **5.6, 5.7,** and **5.8** above, three classification algorithms are used to detect and classify SIMBox/Bypass fraud using train test split data validation techniques. The following Table **5.3** summarizes the detailed indications of the classification algorithms' confusion matrices.

Table 5.3 Confusion Matrix of Train Test Split Data

Confusion matrix			Number of Instances	Correctly classified	Incorrectly classified	Accuracy	model
X	Y	Classified as					
10463	61	X=legitimate	12960	12846	114	99.17%	MLP
53	2383	Y=fraudulent					
10415	109		12960	12768	192	98.90%	LSTM
83	2353						
10355	169		12960	12714	246	98.10%	RNN
77	2359						

As shown in the table above, in addition to other performance measures, the confusion matrices show their performance as per TP (the legitimate call, also classified as a legitimate call) from a valid service provider Ethio Telecom, TN (the fraudulent call, also classified as fraudulent) call from SIMbox device, FP (Fraudulent call but misclassified as a legitimate call) and FN (legitimate call but misclassified as a fraudulent call). As we can see, the model can classify calls to their proper classes by minimizing misclassified instances. To classify the instances into their appropriate classes, we employed three classification techniques and compared their performance. In terms of performance, the MLP outperforms the LSTM and the RNN.

When we take the confusion matrix performance, we must examine which value must be lowered depending on the application. In our scenario, we are identifying fraud, thus our primary goal is to reduce the number of negative instances misclassified as positive instances, which means FP. FP is the number of fraudulent calls that our model misclassified as legitimate calls. However, as demonstrated by both the 10-fold cross-validation technique and the train test split data validation technique, there is a very small number of misclassified instances when compared to correctly classified legitimate calls and correctly classified fraudulent calls. If the number of FP instances is significant, this indicates that a large number of fraudulent instances are misclassified as TP

legitimate ones. This has a multifaceted impact on telecom firms. As a result, in such an application, we must focus on reducing the number of fraudulent calls that are misidentified as legitimate calls. The confusion matrix performance parameter of our classification models in both 10-fold cross-validation and train test split validation techniques indicates that our model does well in recognizing both legitimate and fraudulent calls by minimizing misclassified instances. We can generalize from this that our model can recognize both fraudulent and legitimate calls as fraudulent calls and legitimate calls respectively by minimizing misclassified instances.

Therefore, the false Positive (FP) rate should be considered when evaluating a fraud Detection System. A high FP rate means a large number of fraudulent cases are determined as non-fraudulent(legitimate) cases by the model, which can cause huge financial losses as there will be no further investigation into them (Babaei et al., 2019).

In addition to performance measures such as accuracy, precision, recall, F1 score, and confusion matrix, we can also check the performance of our model using the ROC curve (Receiver Operating Characteristic curve), which measures and plots the true positive rate (TPR) versus the false positive rate (FPR) at various threshold settings. The ROC curve of the classification algorithms is depicted in the following figure using the test split validation technique.

Choosing the right evaluation metric is often a problem-dependent process. While a metric might fit perfectly to some problems, it may be unsuitable for other problems. Based on previous works in this area, it can be concluded that accuracy is a useful metric for performance evaluation, although it should not be concluded that it is sufficient for determining whether a proposed approach is suitable or not. In telecommunication fraud detection, ROC and AUC are vital metrics that can present important information. The advantages of ROC consist of being conceptually simple and useful for experiments in which the data is skewed and giving a more extensive measure of classification performance (Fawcett, 2006).

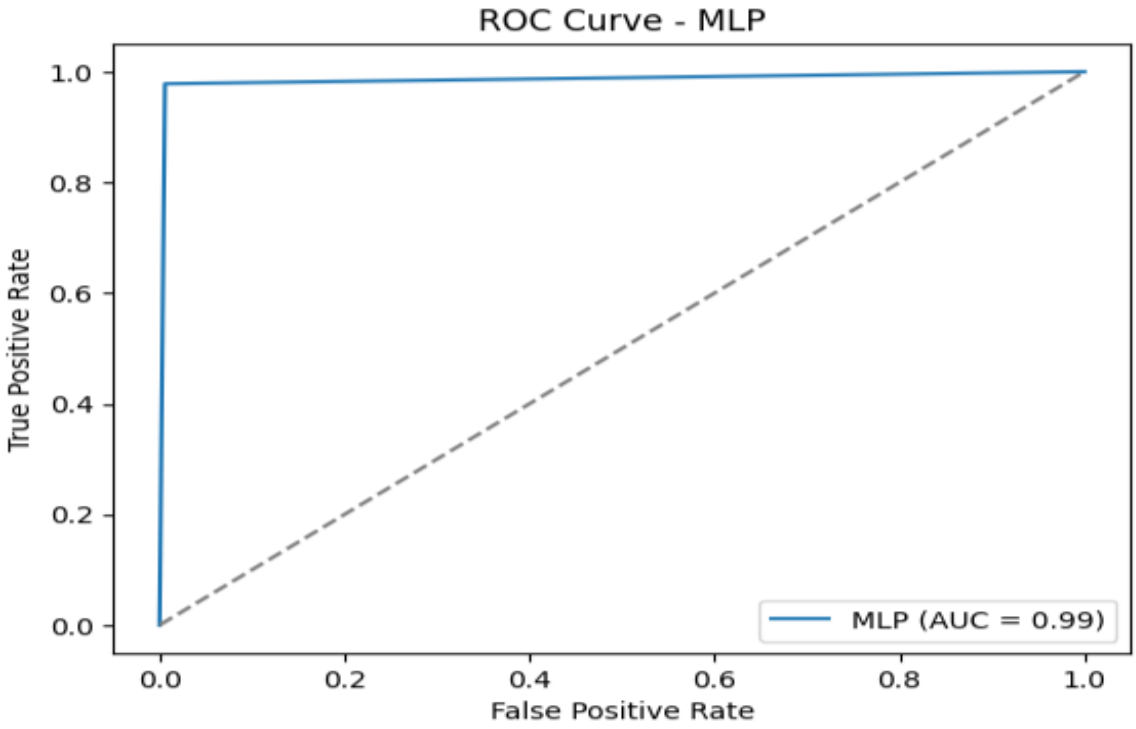

Figure 5.9 ROC Curve of MLP

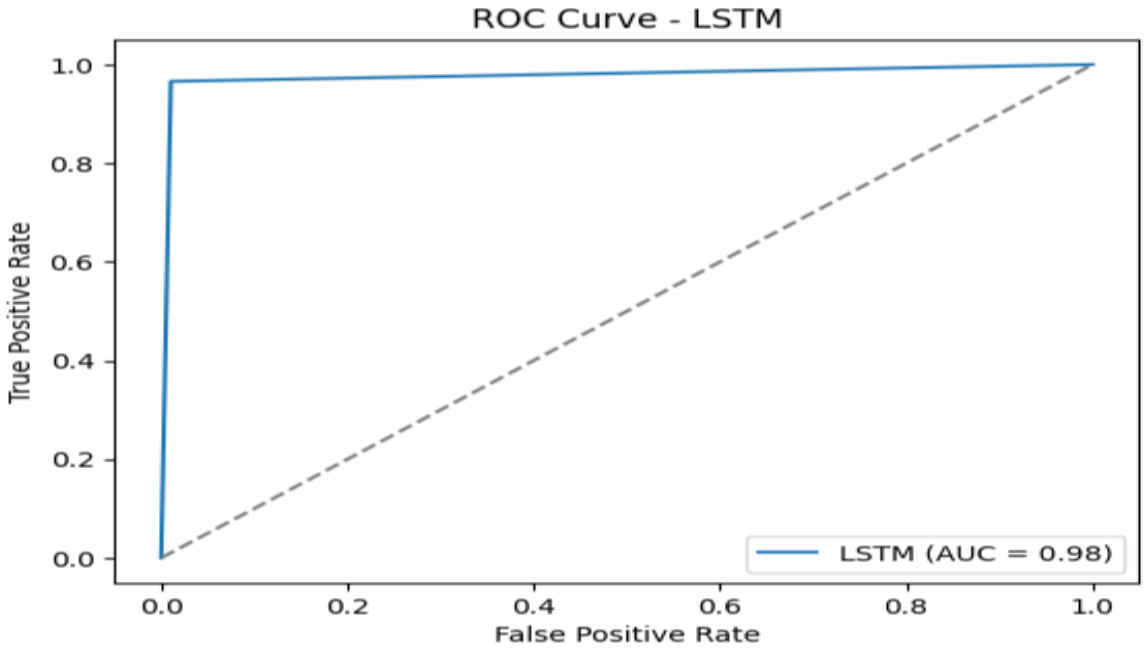

Figure 5.10 ROC Curve of LSTM

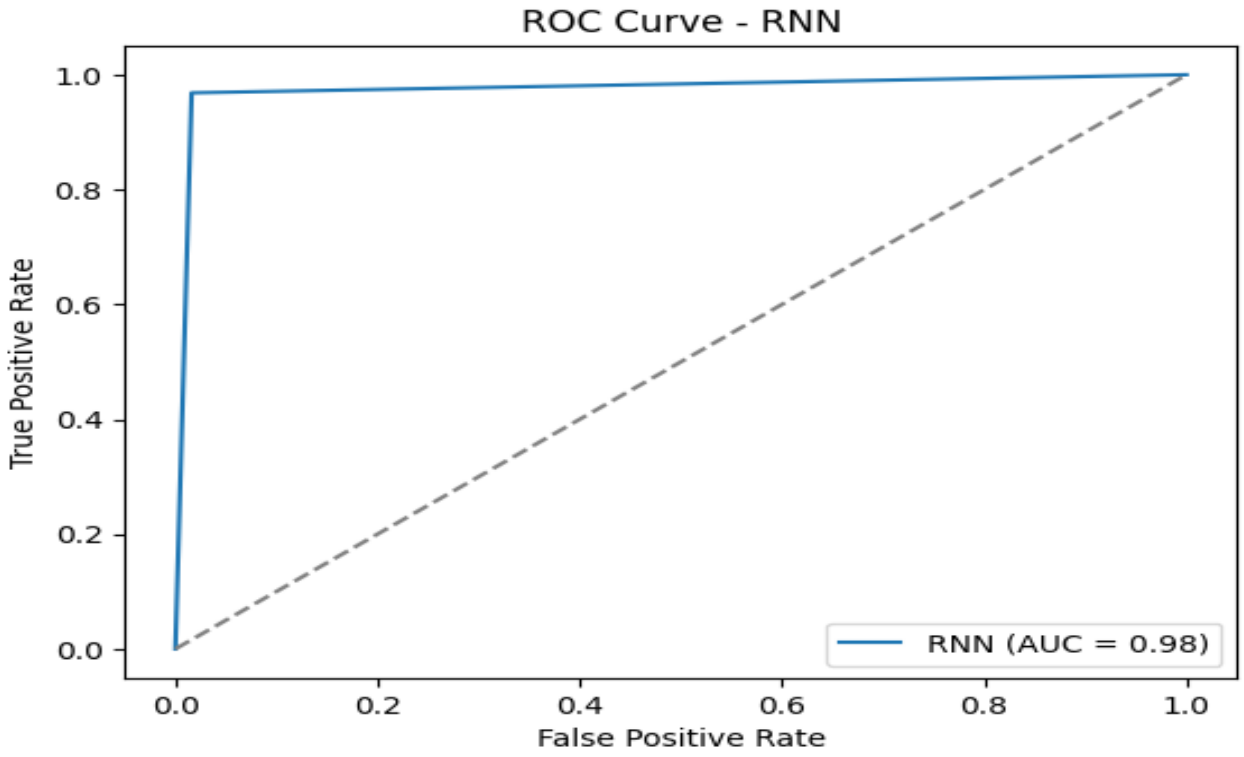

Figure 5.11 ROC Curve of RNN

As shown in Figures **5.9, 5.10, and 5.11** above, the X-axis represents the false positive rate (FPR), which is the proportion of negative instances incorrectly classified as positive, and the Y-axis represents the true positive rate (TPR), which is the proportion of positive instances correctly classified as positive, and the curve is calculated by FPR and TPR. The ROC curve of a perfect classifier would pass through the top left corner of the plot, suggesting a high true positive rate and a low false positive rate. The area under the ROC curve (AUC) is a common summary metric of classifier performance, with a larger AUC indicating better performance. Based on this, as indicated in Figures **5.9, 5.10, and 5.11** above, the ROC curve in all classification algorithms passes through the top left corner and they obtain 99%, 98%, and 98% MLP, LSTM, and RNN models scored AUC respectively.

In conclusion, with AUC-ROC values of 99% and 98% and 98% of MLP, LSTM, and RNN models respectively, and ROC curves passing through the top-left corner, the classification models exhibit excellent discrimination and have a high overall performance in terms of correctly classifying positive instances while maintaining a low false positive rate. It means that the models are capable

of classifying legitimate and fraudulent instances properly and confidently. This is especially desired in many real-world situations, such as fraud detection where detecting positive instances while limiting false positives is critical.

In general, our model performs well in detecting and classifying SIMBox/Bypass fraud in both 10-fold cross-validation and train test split (80% for training and 20% for testing) validation techniques, owing to the usage of multiple performances measuring metrics such as Accuracy, precision, recall, F1score, confusion matrix, and ROC curve. MLP, RNN, and LSTM were chosen and trained as deep learning/deep neural network techniques. We determined that from the total instances, 81.20% of the instances in the training dataset are genuine calls (calls from valid service provider Ethio Telecom) and 18.80% of the instances are fraudulent calls (calls from SIMBox devices). As we have seen in the previous sections of this chapter, MLP achieves 99.17% accuracy and ranks first, LSTM achieves 98.90% accuracy and ranks second, and RNN achieves 98.10% accuracy and ranks third trained with train test split data validation technique. The following Figure **5.12** shows the performance of our classification model when compared with its accuracy.

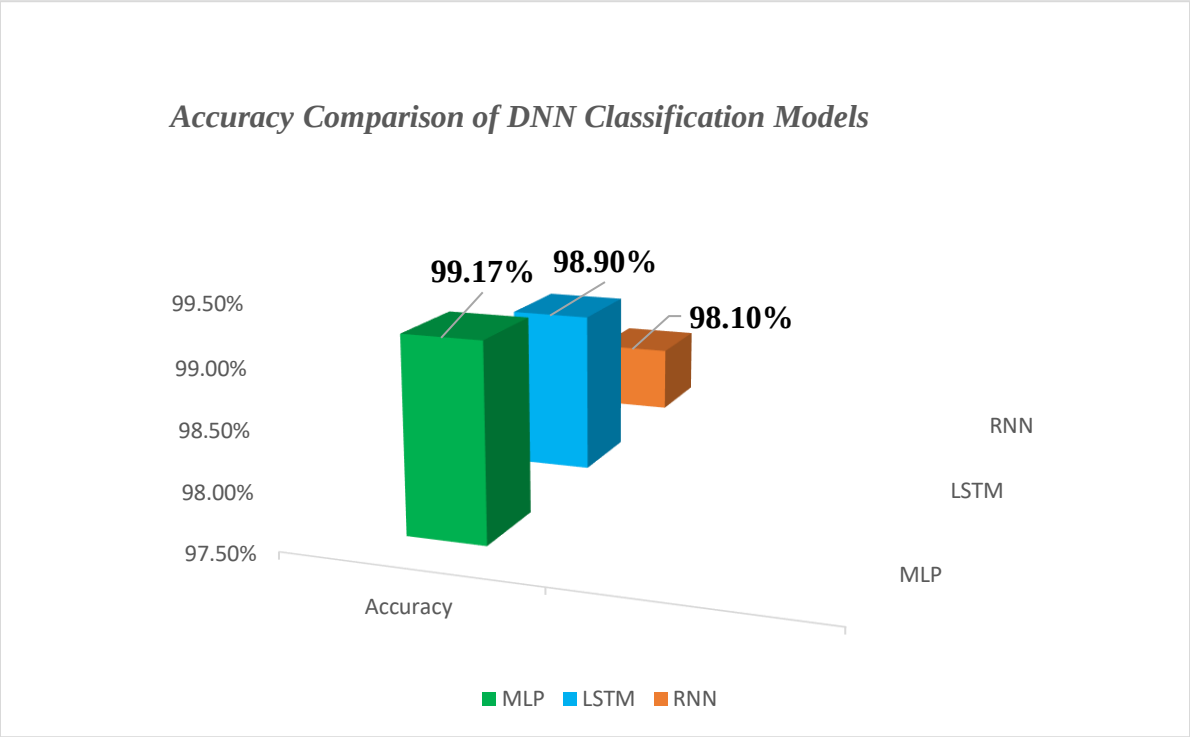

Figure 5.12 Accuracy Comparison of DNN Classification Models

As shown in the performance evaluation parts of this chapter, MLP outperforms the other two DNN classification models, LSTM and RNN, in detecting and classifying SIMbox/Bypass fraud. This is demonstrated by MLP's ability to deeply grasp fraudulent activities by overcoming the dynamic nature of fraud. Even though the other two DNN models got nearly identical results, MLP outperformed them for the following reasons.

The following are some of the reasons why MLPs are effective at detecting SIMbox fraud.

- **Non-linear relationships can be modeled using MLPs**: SIMbox/bypass fraud detection activities frequently necessitate the capture of non-linear relationships and complicated patterns in data. MLPs with numerous hidden layers and non-linear activation functions can model these complicated relationships effectively, allowing them to detect fraudulent patterns with greater accuracy.
- **MLPs can handle tabular data:** SIMbox/bypass fraud detection frequently includes analyzing structured data such as call detail records (CDRs) or usage logs, which MLPs can handle. MLP models are well-suited for tabular data processing and can effectively capture complicated patterns in data.
- **Limited temporal dependencies**: SIMbox/bypass fraud detection often focus on identifying fraudulent behavior based on subscriber usage rather than examining extensive data sequences. Because LSTM and RNN models are designed to capture temporal dependencies in sequential data, they are better suited for jobs involving longer-term relationships. However, in scenarios when fraud tendencies can be recognized based on localized patterns or individual instances, the simpler design of MLPs can perform just as well or even better.
- **Lower computational complexity:** MLP models have a simpler design than LSTM and RNN models, making them less computationally expensive and faster to train. This advantage becomes crucial when dealing with large-scale fraud detection systems that require massive amounts of data to be processed.
- **Labeled data availability**: MLP models frequently perform well in fraud detection tasks because they can efficiently use huge labeled datasets. With an abundance of labeled data, MLP models may learn and generalize from diverse fraud patterns, resulting in accurate detection.

MLPs, in general, are an excellent model for identifying and classifying SIMbox fraud in telecom networks, allowing for rapid and precise identification and response to fraud.

In general, we discovered that among the selected DNN models, the MLP model outperforms in detecting and classifying SIMbox/ Bypass Fraud. This is good news for telecom companies, particularly Ethio Telecom, on which our study focuses. to respond to new fraud strategies, fraud detection systems should be constantly updated and enhanced. Furthermore, combining fraud detection with other security measures such as authentication methods, network monitoring, and incident response systems can give a comprehensive approach to telecom network security.

5.2 Integration of Model to Telecom Network

Integrating fraud detection in a telecom network often means implementing a comprehensive system that monitors numerous aspects of network activity to detect fraudulent activities. We used a different strategy to detect this fraud by employing selected DNN models and CDR data from Ethio Telecom. The first method we used to detect fraudulent activity was to analyze CDRs call records, which can help identify abnormal calling patterns such as excessive call durations, frequent international calls, or high-volume calls, and making an international call from the same location repeatedly or frequently making an international call from an area where international calls are not common. The second strategy involved looking into the Subscriber Behavior Analysis from CDR, which may be used to detect abnormal actions such as SIM switching, call forwarding abuse, and using HBS to mimic a legitimate subscriber. Monitoring variables like call history, location data, and usage trends can help uncover fraudulent activity.

In general, our model performed well in detecting and classifying calls as legitimate (from a valid service provider Ethio Telecom) and fraudulent (from a SIMbox device) by overcoming the dynamic nature of fraud, which is a serious problem that FMS and TCG cannot capture. As a reactive technique, the fraud detection system detects fraudulent activity based on a subscriber's CDR and usage data. As a result, Ethio Telecom now employs FMS, which identifies fraud based on predefined rules and is unable to capture dynamic fraudulent activity. To address this issue, we designed the Intelligent Approach model, which overcomes the limitations of the current detection system. As a result, we recommend that companies implement this detection approach into their

fraud detection mechanism to overcome the dynamic fraudulent behavior that comes with technological progress and get more benefits from it.

5.3 Research Question Discussion

In general, our study answers the research questions that were sought to be answered at the end of this study. Such as:

RQ1: What are the features that can be used to detect and classify SIMbox/Bypass fraud?

Because data is the most critical component of the deep learning system, we obtained CDR data from an Ethio telecom company. We preprocess and choose the relevant features to detect and classify SIMBox/Bypass Fraud after collecting the CDR data. We have identified the behavior of the SIMBox/Bypass fraud via related works and literature reviews before selecting the main feature, and which data aspects are more significant to detect this fraud. As a result, using the CDR data, data attributes containing the usage history are more important in identifying fraudulent behavior from legitimate customers. These usage data include voice calls per minute, text (SMS) messages, data (Internet) usage per megabyte, and Cell ID, which uniquely identifies each BTS where a customer makes a call over the network. These usage data contain a wealth of information that is utilized to detect and classify calls based on their origin (legitimate service provider or SIMBox device). For example, many fraudulent calls have no SMS or data usage and are made from the same BTS, which may be traced by CELL ID. We chose twelve data features from Ethio Telecom's CDR data based on these indications, which offer detailed information about each customer's usage history and behavior analysis. Finally, we used the data attributes we chose to train our model to detect and classify SIMBox/Bypass fraud.

RQ2: What are the available deep learning algorithms for detecting and classifying SIMbox/Bypass fraud?

As technology evolves, fraudsters have seized the opportunity to conduct fraud in telecom firms. Because fraudulent behavior changes over time, a detection approach that fully learns the fraudster's behavior is required. One of these approaches is the deep learning technique, which can learn from data and make a decision. We proposed a Deep Learning approach to overcome the gap in the related works after reviewing the related works done to detect and classify SIMBox/Bypass

Fraud. As we discussed in our study, we chose three Deep Learning/deep neural networks, MLP, LSTM, and RNN, to make a significant contribution in the area of cyber security, especially in telecom fraud detection. In conclusion, due to their various capabilities, MLP, RNN, and LSTM are relevant models for SIMbox/Bypass fraud detection. MLP is capable of dealing with complex non-linear interactions, RNN is capable of processing sequential input and considering temporal patterns, and LSTM enhances RNN by addressing the vanishing gradient problem and capturing long-term dependencies. By combining their abilities to extract relevant characteristics and patterns from data, these models provide a comprehensive toolkit for detecting and classifying SIMbox fraud.

RQ3: Which deep learning algorithm will efficiently detect and classify SIMbox/Bypass fraud?

In the proposed DNN models for SIMBox/Bypass fraud detection, the MLP outperformed both the LSTM and RNN models. The MLP model fared better due to its inherent characteristics. MLP is a feedforward neural network made up of numerous layers of interconnected neurons. It excels at learning complicated patterns and correlations among data, making it well-suited for fraud detection tasks. Unlike LSTM and RNN, MLP lacks recurrent connections and memory cells, allowing it to process incoming data independently and in parallel. This MLP feature allows for faster training and inference, which contributes to its superior performance. LSTM and RNN models, on the other hand, are intended to capture sequential dependencies in data, making them better suited for tasks involving temporal information. However, in the context of SIMBox/Bypass fraud detection, temporal correlations may not be as important, and the emphasis may be more on collecting complex patterns and features within the data. As a result of its ability to understand complex correlations between variables, MLP outperforms LSTM and RNN models in this case.

Generally, for SIMBox/Bypass fraud detection, the Multilayer Perceptron outperformed the Long Short-Term Memory and Recurrent Neural Network models. The MLP's ability to learn complicated patterns as well as its independence from data sequence dependencies led to its excellent performance in this context.

CHAPTER SIX

CONCLUSIONS AND FUTURE WORKS

6.1 Conclusions

The illegal or fraudulent use of telecommunications services or equipment for monetary gain or other harmful goals is known as telecom fraud. Fraudsters now perceive technology advancements as an opportunity to dramatically harm telecom companies. SIMBox/Bypass fraud is one type of telecom fraud that has had a substantial impact on telecom operators, ranging from consumer dissatisfaction to significant revenue loss and security issues. Ethio Telecom has also been affected by SIMbox/Bypass fraud. fraudulent want countries with high international call termination fees but cheap local call costs. Even when telecom service providers deploy various detection technologies, telecom fraud remains a problem.

The primary goal of this research is to detect and classify SIMBox/Bypass fraud using user CDR data using deep learning/Deep neural network methods. MLP, LSTM, and RNN are three different deep learning/deep neural network techniques employed.

After collecting user CDR data from Ethio Telecom, we preprocessed the data and selected the features that are relevant for detecting and classifying SIMBox/Bypass fraud in our research. As fraudulent behavior evolves, we used deep learning/deep neural network methods that can detect and classify fraudulent activity in a complicated environment. We used MLP, LSTM, and RNN models, which are commonly used in cyber security. We trained our model using a 10-fold cross-validation technique as well as separate train test data (80% for training and 20% for testing). After all, we have verified that all of our models perform well in detecting and classifying SIMBox/Bypass fraud using several performance measure metrics. therefore, the overall accuracy of our classification model was 99.17% for MLP, 98.90% for LSTM, and 98.10% for RNN. As a result, the MLP model outperforms the other two classification models.

6.2 Major Contributions of the Research

This work makes a substantial contribution to the detection and classification of SIMbox/Bypass fraud in telecom companies. By developing superior detection techniques and precise classification algorithms. Furthermore, this research improves the accuracy and efficiency of fraud detection,

reducing financial losses for telecom carriers. The study also identifies essential features for fraud detection, encourages data fusion and integration, and provides decision support tools for the detection system's integration into Telecom Networks. Also, it will help to prevent financial loss, increase revenue assurance, improve operational efficiency, improve customer satisfaction, ensure regulatory compliance, and establish industry leadership. In addition, this research contributes to handling large Call Detail Records (CDR) data preprocessing and preparation for model training in the context of SIMBox/Bypass Fraud detection and classification. Addressing issues like data cleaning, normalization, and feature selection that come up while working with massive CDR data was the main goal. The creation and application of efficient and effective preprocessing procedures led to an improvement in data quality and preparedness for subsequent model training, and this represents the contribution. This contribution is essential because it establishes the framework for precise and trustworthy fraud detection, which enables telecom companies to reduce revenue losses and improve network security.

Generally, this study provides significant insights and tools for telecom companies to efficiently detect and classify SIMbox/Bypass fraud.

6.3 Future Work

When there is a difference in termination rates between local and international calls, it is one of the more interesting areas of fraud. Ethio Telecom, on the other hand, lowers the cost of local phone calls, which may encourage SIMbox fraudsters to intercept international calls. Continuous research based on CDR data analysis, as well as the addition of other CDR properties such as International Mobile Equipment Identity (IMEI) and Mobile Termination ID (Receivers cell ID), are required as future work or recommendations to improve SIMBox/Bypass fraud detection and classifications. Furthermore, we recommend that the designed intelligent classification model be integrated into their fraud detection mechanism, which overcomes the problem of the dynamic nature of fraud that the present fraud detection system (FMS) cannot capture. Finally, we recommend that in the future, a mitigation (proactive) mechanism be created and implemented that would be integrated into the telecom network to prevent and block calls from SIMBox devices.

Special Acknowledgement

This research was sponsored by Adama Science and Technology University with a grant number

ASTU/SM-R/816/23 Adama, Ethiopia

References

Abuhamoud, N., Alsadi, I., & Ali, S. (2021). Detecting SIMBox Fraud Using CDR Files and Neo4j Technology. *2021 IEEE 1st International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering, MI-STA 2021 - Proceedings*, *May*, 259–263. https://doi.org/10.1109/MI-STA52233.2021.9464510

Agostinelli, F., Hoffman, M., Sadowski, P., & Baldi, P. (2015). Learning activation functions to improve deep neural networks. *3rd International Conference on Learning Representations, ICLR 2015 - Workshop Track Proceedings*, *2013*, 1–9.

Airn, V. (2018). Analysis and Detection of SIM box. *International Journal of Advance Research, Ideas and Innovations in Technology*, *4*(3), 330–334.

Al-Naymat, G., Al-Kasassbeh, M., Abu-Samhadanh, N., & Sakr, S. (2016). Classification of VoIP and non-VoIP traffic using machine learning approaches. *Journal of Theoretical and Applied Information Technology*, *92*(2), 403–414.

Anthony, O. N., & Joy, A. A. (2019). *Grappling with the Challenges of Interconnect Bypass Fraud*. *6*(1), 35–41. https://doi.org/10.9790/0050-06013541

Babaei, K., Chen, Z. Y., & Maul, T. (2019). A study of fraud types, challenges and detection approaches in telecommunication. *Journal of Information Systems and Telecommunication*, *7*(4), 248–261.

Baskar, K. (2021). Innovations in Science and Technology Vol. 1. *Innovations in Science and Technology Vol. 1*, *December*. https://doi.org/10.9734/bpi/ist/v1

Ben Fredj, O., Mihoub, A., Krichen, M., Cheikhrouhou, O., & Derhab, A. (2020). CyberSecurity Attack Prediction: A Deep Learning Approach. *ACM International Conference Proceeding Series*, 0–5. https://doi.org/10.1145/3433174.3433614

Chouiekh, A., & El Haj, E. H. I. (2018). ConvNets for fraud detection analysis. *Procedia Computer Science*, *127*, 133–138. https://doi.org/10.1016/j.procs.2018.01.107

Datta, J., Kataria, N., & Hubballi, N. (2015). Network traffic classification in encrypted environment: A case study of Google Hangout. *2015 21st National Conference on*

Communications, NCC 2015. https://doi.org/10.1109/NCC.2015.7084879

Deng, L., & Yu, D. (2013). Deep learning: Methods and applications. *Foundations and Trends in Signal Processing*, *7*(3–4), 197–387. https://doi.org/10.1561/2000000039

DiPietro, R., & Hager, G. D. (2019). Deep learning: RNNs and LSTM. In *Handbook of Medical Image Computing and Computer Assisted Intervention*. Elsevier Inc. https://doi.org/10.1016/B978-0-12-816176-0.00026-0

Djomadji, E. M. D., Basile, K. I., Christian, T. T., Djoko, F. V. K., & Sone, M. E. (2023). Machine Learning-Based Approach for Identification of SIM Box Bypass Fraud in a Telecom Network Based on CDR Analysis: Case of a Fixed and Mobile Operator in Cameroon. *Journal of Computer and Communications*, *11*(02), 142–157. https://doi.org/10.4236/jcc.2023.112010

Elmi, A. H., Ibrahim, S., & Sallehuddin, R. (2013). Detecting SIM box fraud using neural network. *Lecture Notes in Electrical Engineering*, *215 LNEE*(January), 575–582. https://doi.org/10.1007/978-94-007-5860-5_69

F. Tesfaye. (2020). *Near-Real Time SIM-box Fraud Detec-tion Using Machine Learning in the case of ethio telecom*.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*(8), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010

G, R. (2021). A Study to Find Facts Behind Preprocessing on Deep Learning Algorithms. *Journal of Innovative Image Processing*, *3*(1), 66–74. https://doi.org/10.36548/jiip.2021.1.006

Gardner, M. W., & Dorling, S. R. (1998). Artificial neural networks (the multilayer perceptron) - a review of applications in the atmospheric sciences. *Atmospheric Environment*, *32*(14–15), 2627–2636. https://doi.org/10.1016/S1352-2310(97)00447-0

Gent, A. (2021). SIM boxing – why mobile regulators need to act now. *Computer Fraud and Security*, *2021*(5), 9–12. https://doi.org/10.1016/S1361-3723(21)00052-X

Ghillani, D. (2022). Deep Learning and Artificial Intelligence Framework to Improve the Cyber

Security. *American Journal of Artificial Intelligence*, *x*, *No. x*, x–x. https://doi.org/10.22541/au.166379475.54266021/v1

Hagos, K. (2018). *SIM-Box Fraud Detection Using Data Mining Techniques : The Case of ethio telecom*. 84. http://etd.aau.edu.et/handle/123456789/15238?show=full

Han, J. (2016). *Spectral Feature Selection for Data Mining*.

Ibrahim, I. A., & Saidu, B. (2015). Fraud Management System in Detecting Fraud in Cellular Telephone Networks Internet of Things (IoT), Open Government Data View project CONVOCATION LECTURES View project. *International Journal of Innovative Research in Computer Science & Technology (IJIRCST)*, *3*, 2347–5552. https://www.researchgate.net/publication/332809655

Ighneiwa, I., & Mohamed, H. S. (2017a). Bypass fraud detection: Artificial intelligence approach. *ArXiv*, 3–6.

Ighneiwa, I., & Mohamed, H. S. (2017b). Bypass fraud detection: Artificial intelligence approach. *ArXiv*, *April*, 3–6.

Informatik, F., & Joachims, T. (1998). *K unstliche Intelligenz Making Large-Scale SVM Learning Practical*.

Ji, S., Li, J., Yuan, Q., & Lu, J. (2020). Multi-Range Gated Graph Neural Network for Telecommunication Fraud Detection. *Proceedings of the International Joint Conference on Neural Networks*. https://doi.org/10.1109/IJCNN48605.2020.9207589

Karunathilaka, A. V. V. . (2020). *Fraud Detection on International Direct Dial Calls Fraud Detection on International Direct Dial Calls*.

Lai, K. K. (2006). An Integrated Data Preparation Scheme for Neural Network Data Analysis. *IEEE Transactions on Knowledge and Data Engineering*, *18*(2), 217–230. https://doi.org/10.1109/TKDE.2006.22

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Li, S., & Zhao, D. (2019). A LSTM-based method for comprehension and evaluation of network security situation. *Proceedings - 2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering, TrustCom/BigDataSE 2019*, *1*, 723–728. https://doi.org/10.1109/TrustCom/BigDataSE.2019.00103

Liu, T., Fang, S., Zhao, Y., Wang, P., & Zhang, J. (2015). *Implementation of Training Convolutional Neural Networks*. http://arxiv.org/abs/1506.01195

Los, U. M. D. E. C. D. E. (2011). *Spectral Feature Selection for Data Mining*.

Lu, Y. (2017). *Deep neural networks and fraud detection*.

Łuczak, P., Poniszewska-Maranda, A., & Karovič, V. (2021). The Process of Creating Web Applications in Ruby on Rails. In *Studies in Systems, Decision and Control* (Vol. 330). https://doi.org/10.1007/978-3-030-62151-3_9

Mandru, D. B., Aruna Safali, M., Raghavendra Sai, N., & Sai Chaitanya Kumar, G. (2022). Assessing Deep Neural Network and Shallow for Network Intrusion Detection Systems in Cyber Security. *Lecture Notes on Data Engineering and Communications Technologies*, *75*, 703–713. https://doi.org/10.1007/978-981-16-3728-5_52

Marah, H. M. (2015). *Fraud Detection in International Calls Using Fuzzy Logic*. *July 2017*. https://doi.org/10.1109/ICCVIA.2015.7351891

Mercioni, M. A., & Holban, S. (2020). The Most Used Activation Functions: Classic Versus Current. *2020 15th International Conference on Development and Application Systems, DAS 2020 - Proceedings*, 141–145. https://doi.org/10.1109/DAS49615.2020.9108942

Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). *Activation Functions: Comparison of trends in Practice and Research for Deep Learning*. 1–20. http://arxiv.org/abs/1811.03378

Ogwueleka, F. (2010). Fraud Detection In Mobile Communications Networks Using User Profiling And Classification Techniques. *Journal of Science and Technology (Ghana)*, *29*(3), 31–42. https://doi.org/10.4314/just.v29i3.50052

Oh, B., Ahn, J., Bae, S., Son, M., Lee, Y., Kang, M. S., & Kim, Y. (2023). *Preventing SIM Box Fraud Using Device Model Fingerprinting*. *March*. https://doi.org/10.14722/ndss.2023.24416

Osowski, S., Siwek, K., & Markiewicz, T. (2004). MLP and SVM networks - A comparative study. *Report - Helsinki University of Technology, Signal Processing Laboratory*, *46*(2), 37–40.

Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., Shyu, M. L., Chen, S. C., & Iyengar, S. S. (2018). A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys*, *51*(5). https://doi.org/10.1145/3234150

Raghavan, P., & Gayar, N. El. (2019). Fraud Detection using Machine Learning and Deep Learning. *Proceedings of 2019 International Conference on Computational Intelligence and Knowledge Economy, ICCIKE 2019*, 334–339. https://doi.org/10.1109/ICCIKE47802.2019.9004231

Salehinejad, H., Sankar, S., Barfett, J., Colak, E., & Valaee, S. (2017). *Recent Advances in Recurrent Neural Networks*. 1–21. http://arxiv.org/abs/1801.01078

Satapathy, S. K., Loganathan, D., Sangameswar, M. V., & Vodnala, D. (2021). *Automated Classification of Sleep Stages Based on Electroencephalogram Signal Using Machine Learning Techniques*. https://doi.org/10.1007/978-981-16-2248-9_39

Schmidhuber, J. (2015). Deep Learning in neural networks: An overview. *Neural Networks*, *61*, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003

Sharma, S., Sharma, S., & Athaiya, A. (2020). Activation Functions in Neural Networks. *International Journal of Engineering Applied Sciences and Technology*, *04*(12), 310–316. https://doi.org/10.33564/ijeast.2020.v04i12.054

Shing Lim, K., Hong Lee, L., & Sim, Y.-W. (2021). A Review of Machine Learning Algorithms for Fraud Detection in Credit Card Transaction. *IJCSNS International Journal of Computer Science and Network Security*, *21*(9). https://doi.org/10.22937/IJCSNS.2021.21.9.4

Shrestha, A., & Mahmood, A. (2019). Review of deep learning algorithms and architectures.

IEEE Access, *7*(c), 53040–53065. https://doi.org/10.1109/ACCESS.2019.2912200

Smagulova, K., & James, A. P. (2019). A survey on LSTM memristive neural network architectures and applications. *European Physical Journal: Special Topics*, *228*(10), 2313–2324. https://doi.org/10.1140/epjst/e2019-900046-x

Terzi, D. S., Sağıroğlu, Ş., & Kılınç, H. (2021). Telecom fraud detection with big data analytics. *International Journal of Data Science*, *6*(3), 191. https://doi.org/10.1504/ijds.2021.121090

Ullah, I., & Mahmoud, Q. H. (2022). Design and Development of RNN Anomaly Detection Model for IoT Networks. *IEEE Access*, *10*, 62722–62750. https://doi.org/10.1109/ACCESS.2022.3176317

Van Efferen, L., & Ali-Eldin, A. M. T. (2017). A multi-layer perceptron approach for flow-based anomaly detection. *2017 International Symposium on Networks, Computers and Communications, ISNCC 2017*. https://doi.org/10.1109/ISNCC.2017.8072036

Van Houdt, G., Mosquera, C., & Nápoles, G. (2020). A review on the long short-term memory model. *Artificial Intelligence Review*, *53*(8), 5929–5955. https://doi.org/10.1007/s10462-020-09838-1

Veloso, B., Tabassum, S., Martins, C., Espanha, R., Azevedo, R., & Gama, J. (2020). Interconnect bypass fraud detection: a case study. *Annales Des Telecommunications/Annals of Telecommunications*, *75*(9–10), 583–596. https://doi.org/10.1007/s12243-020-00808-w

Wang, S., Tang, J., & Liu, H. (2020). Encyclopedia of Machine Learning and Data Science. *Encyclopedia of Machine Learning and Data Science*, *October 2017*. https://doi.org/10.1007/978-1-4899-7502-7

Weiss, G. M. (2005). Chapter 56 DATA MINING IN TELECOMMUNICATIONS. *Data Mining and Knowledge Discovery Handbook*.

Zheng, Y. J., Zhou, X. H., Sheng, W. G., Xue, Y., & Chen, S. Y. (2018). Generative adversarial network based telecom fraud detection at the receiving bank. *Neural Networks*, *102*, 78–86. https://doi.org/10.1016/j.neunet.2018.02.015