Deep Learning Based SIMBox Fraud Detection using CDR data: A case of Safaricom Ethiopia

By

Fikirte Endalew

Supervised By

Dr.Elefelious Getachew

School of Information Technology and Engineering
Addis Ababa Institute of Technology
Addis Ababa University

A Thesis Submitted to the School of Information Technology and Engineering in Partial Fulfillment for the Degree of Master of Science in Cyber Security

Addis Ababa, Ethiopia
June 2024

Deep Learning SIMBox Fraud Detection using CDR data: The case of Safaricom Ethiopia

by

Fikirte Endalew

Examiners' Committee

Name	**Signature**	**Date**
_____	_____	_____
(**Thesis Advisor**)		
_____	_____	_____
(**Dean of the School**)		
_____	_____	_____
(**Internal Examiner**)		
_____	_____	_____
(**External Examiner**)		

A Thesis Submitted in Partial Fulfillment of the Requirements

For Masters of Science in Cyber Security

Addis Ababa Institute of Technology

School of Information Technology and Engineering

June 2024

Abstract

The telecommunications industry is a critical component of modern society, facilitating communication and data exchange across individuals and businesses globally. However, this interconnectedness also presents vulnerabilities that malicious actors exploit. In the telecom sector, fraud usually refers to deliberate misuse of voice and data networks as well as service theft. One of the most difficult problems telecom organizations worldwide have is SIMBox fraud. A SIMBox fraud diverts international calls to a cellular device through the internet via a device called a SIMBox, routing telecom services to local networks into the network as local services, using hundreds of low-cost or even unpaid SIM cards, which are often obtained with forged identities.

Ethiopia's distinctive ethnolinguistic, cultural, and socioeconomic landscape significantly shapes its Call Detail Record (CDR) data. To effectively detect SIM Box fraud within this context, it is imperative to develop fraud detection models that are specifically tailored to the Ethiopian telecommunications environment.

Detecting fraud activities becomes increasingly challenging as the number of subscribers and CDR log volumes and velocities increase. In order to identify telecom fraud via data mining techniques, deep learning techniques have become more and more popular in the telecom sector and other domains in recent years. While Machine Learning algorithms demonstrate effectiveness in detection, a fundamental challenge lies in balancing speed with accuracy. This challenge requires a careful balance between the two, as optimizing one metric often compromises the other. Telecom operators are facing financial losses due to SIM box fraud. Early detection of these fraudulent activities is critical to minimize revenue leakage. Therefore, evaluating the effectiveness of various fraud detection systems is essential to ensure a swift response.

In this thesis, a CRISP-DM based methodology is followed to collect, discover, preprocess and model as well as evaluate CDR based SIMBox fraud detection for Safaricom Ethiopia. BERT, MLP, LSTM and classic rRNN deep learning models are implemented with evaluation. The results show that the rRNN algorithm with GRU architecture showed the highest accuracy of 99.7% followed by LSTM, BERT and MLP at 99.1%, 98.6% and 96.7% of accuracy respectively.

Acknowledgment

Without the constant encouragement and assistance of numerous people, this thesis would not have been achievable.

Initially and primarily, I am incredibly grateful to my husband, Tariku Eshetu. His unwavering love, understanding, and constant support throughout my entire educational journey have been my rock. His sacrifices and encouragement allowed me to dedicate the time and focus necessary to finish this thesis.

I want to thank my parents from the bottom of my heart , Endalew Teshome and Lakech Eshete. Through their unwavering trust in my potential and their steadfast encouragement have been a source of strength throughout my life. Their sacrifices to ensure I had access to education have paved the way for this achievement.

I sincerely appreciate my advisor, Dr. Elfelious Getachew, This thesis has been greatly shaped by his persistent support, intelligent criticism, and important direction. His expertise and patience have helped me navigate the research process and refine my ideas.

My sincere appreciation to Mr. Muluken Sholaye for everything. His exceptional technical support and willingness to answer my questions have been invaluable throughout this project.

My sincere thanks go to Mr. Mengistu Mamuye, who facilitated the data acquisition process by connecting me with Safaricom Ethiopia teams. I express my gratitude to the Safaricom Ethiopia team for granting me access to the data required for this study.

Finally, I am eternally grateful to God for granting me the courage, perseverance, and ability to finish my thesis.

Thank you all.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ML	**M**achine **L**earning
DL	**D**eep **L**earning
ISP	**I**nternet **S**ervice **P**rovider
CDR	**C**all **D**etail **R**ecord
LSTM	**L**ong-**S**hort **T**erm **M**emory
RNN	**R**ecurrent **N**eural **N**etworks
M2RNN	**M**any-to-one **R**ecurrent **N**eural **N**etwork
DNN	**D**eep **N**eural **N**etworks
MSC	**M**obile **S**witching **C**enter
GSM	**G**lobal **S**ystem for **M**obile communication
MLP	**M**ulti-**L**ayer **P**erceptron
ANN	**A**rtificial **N**eural **N**etwork
GPS	**G**lobal **P**ositioning **S**ystem
CNN	**C**onvolutional **N**eural **N**etwork
OCS	**O**perations **S**upport **S**ystem
CCB	**C**losed **C**ircuit **B**illing
SVM	**S**upport **V**ector **M**achine
CPU	**C**entral **P**rocessing **U**nit
RAM	**R**andom **A**ccess **M**emory

GRUs	**G**ated **R**ecurrent **U**nits
PII	**P**ersonally **I**dentifiable **I**nformation
IMSI	**I**nternational **M**obile **S**ubscriber **I**dentity
PABX	**P**rivate **A**utomatic **B**ranch **E**xchange
BERT	**B**idirectional **E**ncoder **R**epresentations **T**ransformers
CRISP-DM	**C**ross-**I**ndustry **S**tandard **P**rocess for **D**ata **M**ining
FPD	**F**loating **P**hone **D**ata
LAC	**L**ocation **A**rea **C**hanges
GPRS	**G**eneral **P**acket **R**adio **S**ervice
SMS	**S**hort **M**essage **S**ervice
IMEI	**I**nternational **M**obile **E**quipment **I**dentity
EDA	**E**xploratory **D**ata **A**nalysis
Wi-Fi	**W**ireless **F**idelity
PCA	**P**rincipal **C**omponent **A**nalysis

Chapter 1

Introduction

1 Introduction

1.1 Background

Fraud is described as deliberate, dishonest, or fraudulent behavior carried out by one or more parties, typically with the purpose of making money. In the telecom sector, fraud usually refers to deliberate misuse of voice and data networks and service theft[1]. When looking for traffic anomalies, it can be helpful to take into account time stamps, Call Detail Records (CDR), and traffic data. The biggest threat to the telecom sector is fraud, which causes millions of dollars in annual revenue loss for various ISPs[1]. In the telecom industry, there are many different types of fraud activities, including billing fraud, bypass fraud, GSM gateway fraud, SIMBox fraud, over-the-top (OTT) fraud, call forwarding/call diversion fraud, call transfer fraud, CDR manipulation fraud, dealer fraud, interconnect/low-cost routes fraud, and internal fraud, international revenue sharing fraud, etc. In the area of telecommunications, a CDR is a data record created by a telephone exchange or other telecommunications equipment that records the specifics of a phone call or other telecommunications transactions that pass through the facility (such as text messaging). The record contains the following details about the call: the duration, source and destination numbers, completion status, and time. Call detail records (CDRs) are among the most useful data sources for a telecom operator. A phone call or other related transaction's single instance is detailed in the CDR, which

is a data record. A CDR record is created when a transaction travels via a Mobile Switching Center (MSC) or similar telecommunication node. Based on the functionality of telecommunication nodes, the level of information in CDR may vary. Generally speaking, it includes details like the call's origin and destination addresses, start and end times and duration[2].

Real-time CDR generation results in a stream of events that contains useful patterns that depict user activity. The majority of bypass detection solutions use CDR as their data source. Traditional fraud detection systems heavily relied on manual analysis to analyze call detail records (CDRs) stored in a central database to identify patterns and trends, this process is inherently time consuming. Traditional fraud detection methods typically rely on a combination of manual and automated techniques, such as manual reviews and signature-based detection. Criminals can exploit this delay by making multiple fraudulent calls before being flagged by the system. Therefore, real-time CDR stream analysis is necessary to support the real-time detection of such calls. Additionally, Grey calls produce a number of distinct patterns in CDR streams, and systems can effectively use those patterns to detect such fraudulent numbers in real time[2].

There are various traditional solutions for fraud detection in the market. However, due to the large volume and velocity of CDR data, these security solutions have numerous limitations, including manual intensity, a reactive approach, and inaccurate detection regarding real-time fraud detection, predictive capabilities, and the deployment of corrective measures. Traditional fraud detection systems often struggle with performance bottlenecks and reduced detection rates due to their reliance on outdated methods. These systems may also have difficulty predicting future fraud trends, as they primarily rely on historical data patterns. Furthermore, manual intervention is frequently required to initiate corrective measures, leading to delays and potential financial losses. In addition, Because of innovative and frequently changing fraudulent or dishonest activity in the telecom industry, rule based and other hard-coded telecom fraud detection methods have proved ineffective[3]. Therefore, a state-of-the-art AI-based fraud detection model is necessary for Safaricom Ethiopia to maintain its competitive advantage and operate effectively without substantial profit loss.

Telecommunication providers utilize a range of methods to manage fraud. The application of deep learning algorithms is the latest trend in the telecom fraud problem.

Different deep learning algorithms have been implemented to assess and identify fraudulent telecom transactions from the CDR log data.

1.2 Motivation

The telecommunication sector in Ethiopia had been monopolized by Ethio-Telecom for a century. But recently, the policy shifts by the Ethiopian government have been slowly opening itself to foreign investment, which will attract many service providers such as Safaricom.

Although the underlying principles of SIM box fraud remain consistent worldwide, the specific manifestations and patterns of this activity can vary significantly between countries. Cultural, linguistic, socioeconomic, and technological factors unique to each region influence the types of SIM Box fraud that occur and the characteristics of the associated CDR data. In Ethiopia, the country's diverse ethnolinguistic, cultural, and socioeconomic landscape creates a unique context for SIM Box fraud. This necessitates the development of more contextualized fraud detection models specifically tailored to the Ethiopian telecommunications environment. By considering the specific characteristics of the Ethiopian market, such as language usage, social structures, and economic conditions, fraud detection systems can be better equipped to identify and mitigate the risks associated with SIM Box fraud. Fraudsters use a variety of strategies to obtain an unauthorized revenue share from a range of services as the number of services increases.

Consequently, a number of techniques have been employed to detect SIMBox fraud, the most well-known being rule-based. Second, organizations in the telecommunications sector typically share fraud information (best practices, recommendations) amongst operators; third, call center and bill process detection (where the center reports foreign numbers it is unfamiliar with, the domain expert looks into suspicious activity and then decides what needs to be done). Fourth, a deep learning method has recently proven effective in detecting telecom fraud [4].

1.3 Statement of the Problem

In conventional fraud detection systems, the identification of fraudulent behavior is rule-based and relies on a threshold or a predefined set of known fraud patterns to detect anomalous data. However, there are a lot of efficiency issues with this method as it relies heavily on domain experts' knowledge, lacks the ability to provide early warnings, and is susceptible to unknown fraud patterns and a high number of false positives [4].

Detecting fraud activities becomes increasingly challenging as the number of subscribers and CDR log volumes and velocities increase. Deep learning methods have become increasingly prominent in recent years in the telecommunications industry and other fields to detect telecom fraud through data mining methods. While current fraud detection methods are effective, a key challenge lies in balancing rapid identification with accuracy. Fraudsters employing SIMBoxing techniques pose a significant threat to operators' revenue. Early detection is critical to minimize financial losses. Therefore, it is essential to assess the capabilities of the fastest detecting systems.[4].

Mobile phone behavior is shaped by a complex array of demographic, social, cultural, and contextual factors, as shown in [5]. The level of technological adoption within a society also plays a significant role. Countries with high smartphone penetration may exhibit different calling behaviors compared to those where feature phones are more common. Moreover, differences in internet penetration, access to Wi-Fi, and usage of messaging apps can all impact calling patterns captured in CDR data[5]. Furthermore, [6] found that phone usage patterns exhibit significant differences among various social and economic subgroups within the population. Additionally, based on self-reported survey data, the analysis of phone use and access patterns shows significant gender differences in mobile phone use between men, women, and economic groups [7]. Moreover, a closer examination of calling patterns and social network structures using mobile operator billing logs reveals more nuanced and modest differences between social groups [7].

To the best of my knowledge, the telecom industry in Ethiopia does not have any existing state-of-the-art deep learning-based SIMBox fraud detection mechanism. SIM box fraud detection is a country-specific challenge due to the unique regulatory environments, telecommunications infrastructure, and cultural factors that vary between

different regions. Although generic fraud detection solutions can be applied to some extent, building customized models for each country is essential to address the specific nuances and challenges associated with SIM box fraud in that particular context. Factors such as local pricing structures, network topologies, and the prevalence of certain types of fraudulent activity can significantly influence the effectiveness of fraud detection systems. Therefore, developing country-specific solutions ensures that the models are tailored to the unique characteristics of each region, leading to more accurate and efficient detection of SIM box fraud. This study will utilize a highly optimized deep learning algorithm with efficient computing time to improve the accuracy of fraud detection in the telecom industry. In addition, a large and high-quality data set will be collected from Safaricom Ethiopia, as their superior data retention policies ensure quality data. The selected deep learning algorithms will include a number of hidden layers with enabled backpropagation. To ensure the most up-to-date research is considered, a comprehensive literature review of local and international articles relevant to the topic will be conducted and compared. Local research on SIMBox fraud has been based primarily on machine learning. Deep learning, while present in some studies, has not reached its full potential due to the lack of utilization of the latest algorithms, such as rRNNs, LSTMs, MLPs, and BERT.

1.4 Research Question

The goal of this research is to find an answer to the following question;

1. What data features can be utilized to develop a deep learning model that effectively identifies the main Sim-box fraudulent activities encountered by Safaricom Ethiopia?

2. Which deep learning classification algorithm most effectively identifies SIMBox fraud from CDR data?

1.5 Objectives

1.5.1 General Objectives

The main goal of this research is to design and implement a deep learning based SIMBox fraud detection model for Safaricom Ethiopia using call details records (CDR) logs.

1.5.2 Specific Objectives

- To prepare the dataset using Data Cleaning ,Data Transformation techniques.
- To select effective features for SIMBox fraud detection using Correlation Analysis, Dimensionality Reduction techniques.
- To select the best deep learning classification learning algorithm for detecting fraud.
- To build models and evaluate their performances.
- To recommend the best models for SIMBox fraud detection.

1.6 Expected Contribution of the Study

This research investigates SIMBox fraud detection within Safaricom Ethiopia's network using a unique combination of data and deep learning techniques. Our approach offers significant advantages over previous studies.like Real-time Detection, Enhanced Detection, Advanced Analytics Deep learning algorithms, such as recurrent neural networks (RNNs) can effectively analyze the complex and dynamic nature of SIMBox fraud.

We leverage a comprehensive dataset Safaricom Ethiopia provided, encompassing labeled (identified fraud cases) and raw data. This dataset boasts a wider range of features (80 columns) than those used in prior research. This richer data allows for a more in-depth analysis of potential fraud indicators, potentially leading to more accurate fraud detection.

While existing research in Ethiopia primarily focused on EthioTelecom and the studies relied on machine learning, which can struggle with real-time analysis of complex data

sequences [8],[9], This research breaks new ground by employing modern deep learning algorithms such as rRNNs, LSTMs, MLPs, and the novel application of BERT. These models are specifically designed to handle sequential data like call records, enabling near real-time fraud detection.

The findings from this research will empower Safaricom Ethiopia and other telecommunication providers to detect and prevent illegal international call activities associated with SIMBox fraud and achieve near real-time fraud detection capabilities, allowing them to respond to fraudulent attempts swiftly. This proactive approach can significantly reduce financial losses due to SIMBox fraud and enhance network security for both service providers and their customers.

1.7 Scope of the study

SIMBox scams exploit a loophole in telecommunication pricing. Fraudsters use devices equipped with multiple SIM cards to reroute international calls through local networks. Because local calls are significantly cheaper than international calls, scammers can pay much lower rates (or sometimes nothing at all) to the carrier while charging the victim the full international price This results in significant revenue loss for the local telecom operator, as they don't collect the high international call fees. While various machine learning techniques can detect SIMBox fraud with limited accuracy and efficiency, this study will specifically focus on detecting SIMBox activity using state-of-the-art deep learning algorithms. Given the prevalence of SIMBox fraud, this study focuses on developing a robust detection system specifically for local networks. We will utilize four well-known classification algorithms - RNN, LSTM, MLP, and BERT - implemented within a Jupyter Notebook environment.

1.8 Structure of the document

The format of this thesis is as follows. The current literature on fraud, including its various kinds and methods of operation, is reviewed in Chapter 2. This chapter also introduces the deep learning techniques and tools used in this study. The methodologies

and procedures employed in this study are thoroughly described in the research methodology portion of Chapter 3. Chapter 4 presents the proposed system, including its design and architecture. Chapter 5 outlines the comprehensive experimental analysis used to evaluate the proposed system. The findings from the experiments are discussed in detail in Chapter 6. In Chapter 7, the study concludes by summarizing the key discoveries and suggesting possible directions for additional research.

Chapter 2

Literature review

2 Literature review

2.1 Fraud in Telecom Industry

Telecom industries have never been more challenged by fraud. The telecom business has become harder to operate in due to the volume and variety of fraud kinds as well as the misuse of technology advancements. Instead of merely observing what fraudsters are doing, the industry/sector is working hard to protect its clients and reduce revenue leakages. The misuse of a profit organization's system, which does not always result in immediate legal consequences, is another name for fraud[10]. Fraud encompasses a wide range of illegal actions and is defined as the deliberate falsification of information or manipulation of facts. It is defined as any illegal behavior including deceit, concealment, or betrayal of confidence. Usually, the purpose of frauds is for individuals or groups to use illegal means to get money, property, or services, or to prevent payment or loss of services, in order to obtain personal or commercial advantage[9].

The preceding definitions primarily focus on fraud as a broad concept. The telecom sector requires its own definitions of fraud that are specifically adapted to its needs. In this regard, several academics characterized fraud from various angles. Fraud is defined as any voice or data transmission via a telecommunications network where the user

intends to evade or reduce legitimate call charges. It is also defined as obtaining unbillable services and undeserved fees. Telecommunication fraud additionally happens when a person utilizes deception to get telephone services for free or at a discounted price. Even though they use illegal tactics, fraudsters regard themselves as business owners driven and guided by concerns related to cost, marketing, pricing, network design, and operations as any legitimate network operator[11][9].

All in all, Telecommunications fraud, also known as phone fraud, involves using telecommunication services without any intention of paying for them. The fact that there is no risk of localization is the main factor that attracts fraudsters to telecoms fraud. This is due to the fact that all acts are carried out remotely, making the process of localization time-consuming and costly. The fundamental access code, which may be obtained through the use of social engineering strategies and the development of technology advancements, enables the execution of fraud. The financial gains from telecommunications fraud are readily transferable into cash[12].

The telecom industry is home to a wide variety of fraudulent activities. Scholars differ in how they enumerate and classify different forms of fraud. According to [13], there are about six fraud scenarios. These include handset theft, free phone call fraud, premium rate fraud, PABX fraud, subscription fraud, and roaming fraud. In the paper [13], these fraud types are described in some detail. Similarly [12] classified fraudulent activity into four categories: procedural, contractual, hacking, and technology. Due to their dynamic nature and the fact that they can be performed by combining them, there is no specific number for fraud kinds, nor can we be exhaustive by listing them.

Fraudsters that commit subscription fraud obtain accounts without intending to pay the fee. This indicates that the cheater uses the services even if they are not subscribed. Then, unusual activity and fraudulent calls or transactions set the account apart. Either heavy self-use or call selling could be done with this account[14].

On the other hand, superimposed fraud in contrast to subscription fraud, overlaid fraud uses a legitimate account. In this instance, customer actions that deviate from typical patterns are detected and flagged for further investigation. Fraudsters use a variety of methods to steal phone services. These methods include cloning mobile phones to make unauthorized calls and accessing calling card privileges without permission. Cellular cloning, calling card theft, and cell phone theft are a few examples of these situations[14].

Superimposed fraud is the most prevalent type of fraud in private networks. In this case, an employee, the fraudster, gained access to expensive services and outgoing trunks using the authorization code of another employee[12].

The most typical fraud found on the GSM network is subscription fraud. By utilizing a false Identification, a person subscribes to a service. The fraudster can then use the service for their own benefit or to make money. Under the first scenario, the user may give the phone to someone else or use it for personal use. The second is for actual financial gain. In this case, the scammer appears as a small firm to obtain several phones for direct call selling. According to [15], the fraudster, who has no intention of paying his bill, is now selling the airtime to those who want to make inexpensive long distance calls.

A private branch exchange (PBX) fraud occurs when an impostor gains access to a private switching network and utilizes linked external phone lines to dial premium numbers that they control. Private branch exchange fraud happens when an organization's internal network is compromised. There are numerous methods for breaking into a PBX. Companies might leave their default passwords or use social engineering to compromise them; an internal employee or vendor could also be the source of the assault[8].

The call center representatives of a mobile phone company must be convinced to move a phone number to a new device in order to execute a SIM switching assault. They will unwittingly hand on the victim's phone number to the attacker if they do this. A SIM change can be much simpler if there's a cooperative insider to work with. For an attacker to obtain the required information about the victim, social engineering with a cell carrier employee is not even necessary. In order to expand their SIM-swapping attacks, cybercriminals are increasingly enlisting the help of workers of mobile phone providers using social media accounts. Attackers can use the prospect of financial benefit to entice insiders by pretending to be a business recruiting through[9].

SIMBox fraud is a cunning scheme that targets telecom operators by exploiting the difference in call termination charges between international and local calls. Criminals use a device called a SIMBox, essentially a multi-SIM card holder connected to the internet[9]. International calls are received via the Internet using Voice over internet protocol (VoIP). The SIMBox then routes these calls back into the mobile network through a multitude of low-cost, often illegally obtained SIM cards. This makes the

international calls appear as local calls within the network, significantly reducing the cost for the fraudsters. The unsuspecting caller is still charged the full international rate, while the telecom operator loses out on the revenue they would have earned for handling the international call. This scam can cause significant financial losses for telecom companies around the world[16].

According to [13], Fraud is a significant cause of revenue leakage in the telecom sector. The sector suffers tens of billions of dollars in losses due to telecommunication fraud annually. Furthermore, telecom fraud negatively impacts service quality, lost revenue, and inefficient operations. These frauds result in either direct or indirect financial loss for the service provider. When resources are used up without payment to the service provider, this is known as a direct loss. When a user successfully damages the market worth or reputation of the service provider, this is referred to as an indirect loss.

2.1.1 Call Detail Records

Call Detail Records (CDR) can be generated using data from mobile phone usage and movement patterns. While phone location information is crucial for network functionality, CDRs retrieve this data from the network itself, not individual phones. This eliminates the need for battery-draining location tracking apps. Consequently, precise GPS coordinates aren't available, but the sheer volume of data, encompassing movement details from thousands of users, is expected to yield significant results[17].

Call Detail Record (CDR) data offers a vast pool of information since it can be collected from all mobile phones within a provider's network. This data is passively generated by phone activity, including both voice calls and mobile internet use on regular and smartphones. However, a key point to consider is user consent. While leveraging this readily available data presents valuable opportunities, it raises ethical questions about privacy. Transparency is crucial. Users should be clearly informed about how their data is collected, used, and protected. Explaining the purpose of data collection and emphasizing strong data handling practices, robust security measures, and user anonymity are essential to maintain public trust and mitigate potential backlash[17].

2.2 Deep Learning

Deep learning, a type of machine learning, excels at uncovering complex relationships within data by representing the world as a layered structure of concepts. This allows it to learn more flexibly and adapt to new information. Deep learning algorithms excel at uncovering hidden patterns and insights within large amounts of raw data. This ability makes them powerful tools for extracting key features that can be crucial for various applications[18]. Unlike traditional machine learning (ML) methods that depend on manually engineered features like local binary patterns or gradient histograms, deep learning excels at automatic feature extraction. This hierarchical approach allows the model to learn progressively complex representations of the data. In simpler terms, it starts by identifying basic patterns in the initial layers and gradually builds towards a more abstract understanding as it progresses through the network. This makes deep learning particularly well-suited for tasks where extracting meaningful features from raw data is crucial[18].

The field of deep learning, a powerful subfield of machine learning (ML), has revolutionized numerous scientific disciplines. Its ability to analyze massive datasets and generate accurate predictions has proven invaluable. Deep learning algorithms leverage artificial neural networks (ANNs), which learn from data by building layers of increasingly complex representations. Neural networks are structured with layers that process information: an input layer receives data, hidden layers transform it, and the output layer delivers the final result. The overall depth of the model is determined by the total number of layers. Deep learning encompasses a diverse set of algorithms, including convolutional neural networks (CNNs), long short-term memory networks (LSTMs), recurrent neural networks (RNNs), and auto encoders[19]. These diverse algorithms leverage various neural network architectures to achieve specialized functions. Deep learning algorithms can operate with virtually any type of data, claim [18], but they need a large amount of computing power and data to handle complex issues.

Among various neural network architectures, two prominent types stand out: multilayer perceptrons (MLPs) and feedforward neural networks. As per [20], feedforward networks are the foundational type, known for their simplicity. In these networks, information travels in a one-way direction, starting from the input layer where data enters. It then

progresses through hidden layers, where processing and feature extraction occur. Finally, the processed information reaches the output layer, delivering the network's final result.

2.2.1 Convolutional Neural Networks(CNNs)

Convolutional Neural Networks (CNNs), also known as ConvNet, are among the most popular deep neural network architectures used in computer vision. In at least one of their layers, convolution processes data with a grid-like topology rather than general matrix multiplication to analyze visual images. Convolutional networks are basically advanced neural networks that have had great success in real-world practical applications. CNNs have multiple layers and an image input, in contrast to simple neural networks. For object recognition, modern convolutional networks offer a model of visual processing that neuroscientists may study. Numerous statistical characteristics of natural images are translation-invariant. Convolutional neural networks (CNNs) excel at tackling classification tasks where the model automatically extracts meaningful features from the data. This process unfolds progressively through the network's layers, starting with low-level details and gradually building towards more complex representations [18].

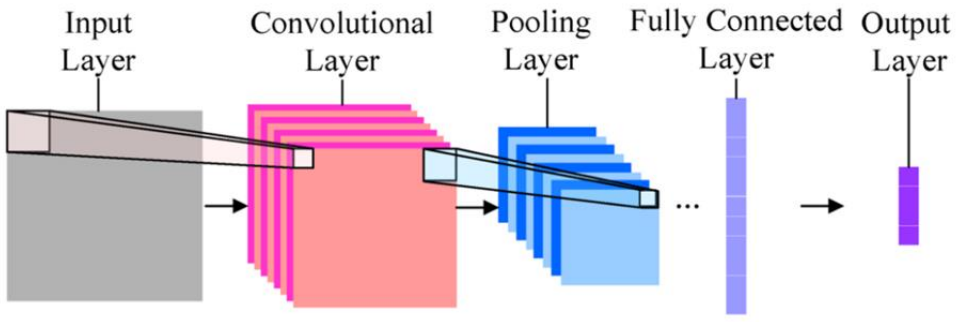

FIGURE 2.1: Convolutional Neural Network

The CNN algorithm's three fundamental phases are multiple convolutions, nonlinear operations, and multiple pooling operations. The first involves applying multiple convolution operations, which essentially scan the input data for specific patterns. These convolutions are followed by non-linear activation functions that introduce important variations in the data. Finally, pooling operations summarize and reduce the complexity of the data, allowing the network to focus on the most relevant features. On the other side, in the third step, the feature map's size is decreased through the pooling

procedure. Convolutional Neural Networks (CNNs) often utilize pooling layers to reduce the dimensionality of their data while preserving important features. Common pooling functions include max pooling, which identifies the most significant value within a defined window, and average pooling, which calculates the average value within the window. These techniques help the network focus on essential information and improve computational efficiency. Following the final pooling layer, the processed data is fed into a fully-connected (FC) layer. Unlike previous convolutional layers with localized connections, each neuron in the FC layer connects to all neurons in the preceding layer. This dense network allows the model to learn complex relationships between features and ultimately perform classification tasks, assigning input data to specific categories[18].

CNN's architecture is unique in that it requires very little image pre-processing to accomplish segmentation, feature extraction, and classification in a single processing module. It is thought that a minimum amount of domain knowledge is sufficient to complete pattern recognition jobs efficiently. Numerous CNN-based applications, including face detection, face recognition, gender recognition, object recognition, character recognition, and texture recognition, have demonstrated this. Convolution, pooling, and fully-connected layers are just a few of the connections and layers that make up a CNN, which also implements regularization. To prevent models from over fitting during training, we can employ regularization techniques. These approaches, such as data augmentation, dropout, or weight decay, help the model generalize better to unseen data[18].

Convolutional Neural Networks (CNNs) rely on a powerful technique called backpropagation for learning. This iterative process involves four key steps: forward pass, backward pass, loss function evaluation, and weight update. During training, filter weights within the CNN are initially assigned random values. In the forward pass, the training images are fed through the network layer by layer. A loss function then compares the network's output with the desired outcome, quantifying the error. The backward pass utilizes this error to calculate adjustments needed for the filter weights. These weight updates are made in small increments over multiple iterations until the network's performance reaches a satisfactory level, signifying convergence[18].

2.2.2 Recurrent Neural Networks(RNNs)

Recurrent neural networks (RNNs) are a powerful category of supervised machine learning model known for their ability to handle sequential data. Unlike traditional neural networks, RNNs incorporate feedback loops, allowing them to analyze information over time or within a sequence. Training RNNs involves feeding them a dataset consisting of paired inputs and desired outputs. By adjusting the connections within the network (weights), RNNs learn to minimize the difference between their predictions and the target outputs, essentially reducing the overall error[21].

Three layers make up a basic RNN: input, recurrent hidden, and output layers., as presented in Figure 2.2. There are N input units in the input layer. This layer receives a series of vectors across time t as inputs, such as

$$(..., x_{t1}, x_t, x_{t+1}, ...)$$

, where

$$x_t = (x_1, x_2, ..., x_N)$$

. A weight matrix is used to define the links between the input units of a fully connected RNN and the hidden units in the hidden layer

$$W_{IH}$$

. The hidden layer has M hidden units

$$h_t = (h_1, h_2, ..., h_M)$$

, that are connected to each other through time with recurrent connections[21].

3 Related Works

Several studies have been conducted on telecommunication fraud. This dynamic global issue is tackled through data mining, machine learning (ML), and deep learning techniques.

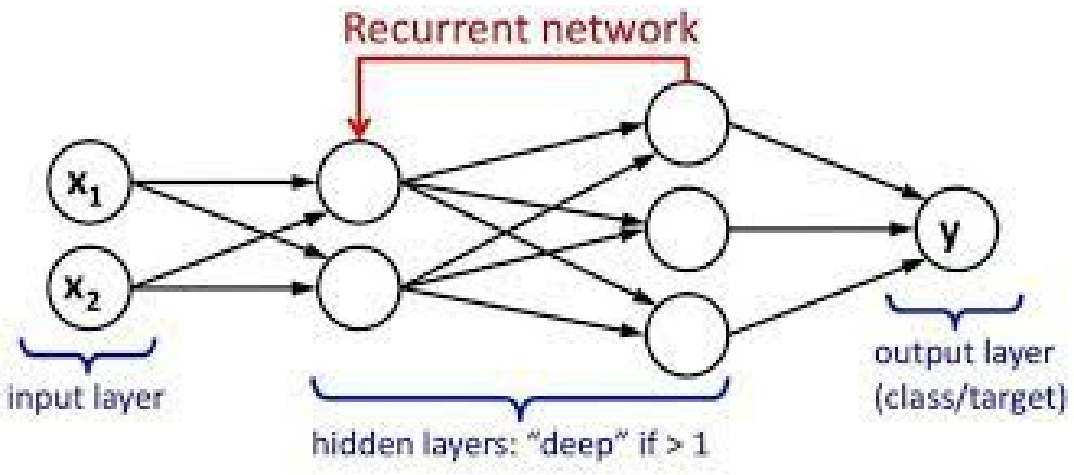

FIGURE 2.2: Recurrent Neural Networks[21]

Aregay [18] With a dataset of 124 identities gathered from the Internet, a personal photograph collection, and the capture of some friends, deep learning algorithms create a face recognition model. There are two distinct datasets for these identities, each including 1240 and 2596 photos. We divided the data into two separate sets. For the first set, we conducted two different experiments: one using a convolutional neural network (CNN) with transfer learning, and another using a multi-task cascaded convolutional network (MTCNN) combined with FaceNet and a support vector machine (SVM). Based on the accuracy achieved in the first set, then further retrained the experiment that performed better in the second set According to experimental findings, the accuracy for CNN with transfer learning was 72.54%, whereas it was 99.24% for MTCNN, FaceNet, and SVM in the first dataset. In contrast, the experiment with higher accuracy in the first dataset is further retrained in the second dataset.

As a strength, the thesis [18] has addressed an interesting and critical type of fraudulent activity Ethio Telecom is facing and chooses state-of-the-art CNN based techniques such as transfer learning and MTCNN with FaceNet as well as traditional Machine learning algorithms such as SVM. On the other hand, the dataset collected is not directly obtained from Ethio Telecom which might implicate some shortcomings if the model is applied to real subscription fraud detection at Ethio Telecom. Additionally, the dataset collected for the experiment is relatively smaller than other related works in the telecom sector.

A study was conducted on the application of data mining techniques for sim-box fraud detection in the context of Ethiopian Telecom[9]. In this work, call detail records (CDRs)

were categorized using models in order to propose a model that better differentiates between fraudulent and real subscribers. Data mining techniques for categorization are applied to Ethio Telecom data using J48, PART, and multilayer perceptron algorithms. A model for predicting fraudulent behaviors was also developed using the WEKA data mining tool. The study made use of pre-paid sample voice CDR data, GPRS, SMS, and other data, such as pre-paid wallet recharge logs from OCS and the CCB data warehouse of Ethiopian Telecom. According to the study's experimentation results, the J48 algorithm had an accuracy level of 99.98%, whereas the model from the PART algorithm demonstrated a 100% accuracy level. According to the study, SIMBox locations and other important data can be found by telecom carriers in general and Ethio Telecom in particular using the rules produced by the PART and J48 algorithms.

The research has chosen various highly effective rule based machine learning algorithms to identify SIMBox based fraudulent CDR call detection. The collected data is integrated from various types such as SMS, GPRS, OCS, and CCB data sources, which makes the dataset reliable and representative of real life situations. In addition, The accuracy of the algorithms is near 100% for most of the algorithms. On the downside, considering how recent the paper is, the researchers only applied classical ML algorithms of rule based nature, not Deep learning. They have also used Weka mostly with its default parameters with less concern about performance tuning.

In order to minimize detection time and revenue loss, Alamirew [4]developed an international revenue sharing (IRS) fraud model for the real-time detection of missed call fraud-generating schemes using an international voice call detail record. The study divides the data set into groups for genuine and fraudulent phone transactions using classifiers Like random forests, support vector machines, and artificial neural networks. The results demonstrate that random forest approaches outperform other methods in both training modes in terms of f-measure, accuracy, receiver operating characteristic curve, building time, and inference time (Splitting and 10-cross-validation). It reached precision on par with the state of the art and operated at 97.00% accuracy. On the plus side, the paper has focused on the IRS fraud model using realtime detection of missed call fraud to minimize revenue loss. A high volume of dataset, around 197,000, is collected from relevant sources. Algorithms of both classical Machin learning and artificial neural networks are chosen with both K-fold cross-validation and percentage

split validation methods, and results are presented in a detailed confusion matrix breakdown. The downside of the paper is that researchers didn't get enough fraudulent call dataset for the experiment(i.e., only 3921), and the Weka tool is used with little focus for parameter tuning.

Additional studies [22] had been carried out in 2013 on the development of a predictive model for data mining-based subscription fraud detection in the context of Ethiopian Telecom. The study, which employed data mining techniques for the analysis and detection of subscription fraud in Ethiopian Telecom, only used 25,000 records. The J48, PART, Random Forest, and Multilayer Perceptron of Artificial Neural Networks classification algorithms were used by the researcher along with WEKA software for data processing. The only data source used for the investigation was prepaid customs CDR data. The researcher in the study noted that more research might be done on postpaid subscriptions of telecommunication fraud and that the study was confined to prepaid subscription fraud. As a strength, the paper has presented literature and related works in a more organized manner in descending order of their relevance to the study with a focus on both local and international papers. In addition, highly relevant data was collected, which resulted in a high accuracy of the model. As a weakness, parameter tuning and efficiency are not well addressed in the experiment. In addition, the research used MS access to data pre-processing followed by model building on Weka instead of using Weka for the whole process, which is more efficient.

Hailemeskel [23] study built a model that uses machine learning methods to identify calls that are subscription fraud using Call Detail Records (CDR) data. Three machine learning algorithms for classifying data have been used: Support Vector Machine (SVM), the Random Forest (RF) and Artificial Neural Network (ANN) multilayer perceptron algorithms. A model for predicting fraudulent calls has been developed using the WEKA data mining tool. They came to the conclusion that the RF classifier, with an accuracy of 99.46%, outperforms the other two algorithms. This research identifies ten key factors that differentiate fraudulent subscribers from legitimate ones, providing valuable insights for subscriber verification. Some of the strengths of the research include a vast and representative choice of modeling algorithms from traditional and ANN. The paper also identified interesting rules and factors found due to the research. The accuracy is also close to perfection. As a weakness, critical features such as requests for IMEI (international mobile equipment identity) numbers and IMSI (International

Mobile Subscriber Identity) data are not part of the dataset collection due to unforeseen circumstances. The use of MS Access, MySQL, and MS Excel for data pre-processing, followed by model building using Weka, can also be seen as inefficient.

In addition, [24] real CDR records from an Ethio Telecom to cluster users using the K-means and fuzzy C-Means clustering methods. There were two main experimental phases in this study. they used traditional fuzzy C-means and K-means clustering techniques in the first experiment phase. They employed the normalization procedure in the second experimental phase before applying the fuzzy C-means and K-means clustering approaches to the data set. The K-means and Fuzzy C-means clustering techniques were used to determine which cluster was valued. As a result, the accuracy of the k-means clustering technique in clustering telecom customers was 99.6%, while the Fuzzy C-means clustering algorithm achieved 99.1% accuracy when comparing these clusters to the ground truth label data set. As a strength, contrary to other local research papers, the research applied a clustering approach and methodology to cluster customers and used a real CDR dataset. The researchers choose the most reliable clustering algorithms such as K-means and C-means. They focused both on the effectiveness and efficiency of the selected algorithm. The researchers also applied the optimal K value in a more scientific manner. On the downside, the dataset used is of random & consecutive 9 days, which might not reflect well on the entire Ethio Telecom business.

In Derebe study [25] evaluated three supervised machine learning algorithms' performances and found differences. J48, Support Vector Machine (SVM), and Artificial Neural Network (ANN). Before analyzing and comparing the algorithms, Call Detail Record (CDR) data were gathered, pertinent features were chosen, and various preprocessing techniques such as feature selection, data cleaning, data frame shaping, and feature kinds were executed. Consequently, it is determined that the J48 algorithm with Cross Validation (CV) options is the best classifier method, scoring 99.3% accuracy. The two algorithms with the greatest accuracy scores, ANN (CV) and SVM (ST), were ranked second and third, respectively, with 97.51% and 96.0%. The pros of the paper include applications of comparative performance analysis of various ML and deep learning algorithms as well as intense use of state-of-the-art feature selection, cleaning, integration techniques with good accuracy at the end. Cons of the paper, on the other hand, include the potential susceptibility of the model to overfitting due to sequential data collection of random days of CDR data.

By using the supervised Machine Learning (ML) method, Tesfaye[26],Using the Sliding Window (SW) aggregation mode, a pertinent dataset instance is generated, reducing the detection time to one hour. The techniques of Random Forest (RF), Artificial Neural Network (ANN), and Support Vector Machine (SVM) were utilized as three supervised machine learning classifiers. The researcher collected CDR data, relevant attributes were selected, and used rigorous pre-processing. They show that RF classifiers using SW aggregation mode cross-validation produce a higher classification accuracy of (96.2%). The application of sliding window aggregation mode to solve detection delay, the choice of modeling algorithms from various categories such as ML and ANN, continuous engagement with domain experts, and the use of intense data pre-processing steps can be seen as strengths for this research work. On the flip side, the dataset size can be considered relatively small, with only 5000 fraudulent entries.

Using data mining techniques, call detail records (CDRs) from Ethiopian Telecom were gathered in order to create models of both legitimate and fraudulent phone behavior[8]. In addition, four classification techniques are used: hybrid algorithms, neural networks, decision trees, and rule-based induction. The data set was first subjected to data analysis, and nine carefully chosen aspects of data were taken from Customer Database Records in order to classify the data. They ultimately outperformed the other four algorithms and obtained strong results from PART rule based and hybrid (J48 and PART) algorithms. As a strength the researchers applied various nature algorithms such as tree based and rule based types for model building as well as deep learning and hybrid algorithms. But only called number, calling number, GPRS, SMS, call fee, date and time, duration of call detail records and location number, leaving other parameters out such as IMEI.

Gebremeskel [18] also constructed a model that makes use of Call Detail Records (CDR) data and machine learning algorithms to identify subscription fraud calls. A quantitative laboratory experimental research approach was used to carry out this investigation. Artificial Neural Network (ANN) multilayer perceptron algorithms, Support Vector Machine (SVM), and Random Forest (RF) are the three machine learning classification methodologies that have been used. The WEKA data mining technology has been used to create a fraud call prediction model. With an accuracy of 99.46%, the researchers find that the RF classifier outperforms the other three algorithms. The main discovery

of this study is the application of ten intriguing characteristics to differentiate between authentic and fraudulent subscribers.

The above are research works that target identification of fraud detection using various mechanisms of Ethiopian origin, specifically Ethio Telecom. There is also a plethora of research work on the application of various ML/DL algorithms for fraud detection using various telecom datasets.

Misrak [27]study developed a model utilizing Call Detail Records (CDR) data and machine learning algorithms to detect SIMBox fraud calls. Random Forest (RF), Support Vector Machine (SVM), and Neural Network (NN) were employed as classification methodologies. RF and NN achieved 100% accuracy for 1-hour, 1-day, and 7-day, while the accuracy of SVM was 54% for 1-hour, 75% for 1-day, and 75% for the categories of 7-day. This indicates the superiority of RF and NN over SVM for this task.

These research efforts, focusing on Ethiopian Ethio Telecom, leverage large datasets and Python's Scikit-learn library for data analysis and modeling. While the study's strength lies in its comprehensive data usage and robust analysis tools, it has limitations in algorithm selection. The chosen traditional machine learning algorithms may not be optimal for handling sequential data, which could be a factor in SIMBox fraud detection.

Additional study kahsu [28] proposed a model employing Call Detail Records (CDR) data and machine learning techniques to identify SIMBox fraud calls. Random Forest (RF), Support Vector Machine (SVM), and Artificial Neural Network (ANN) were utilized as classification algorithms. RF achieved accuracy's of 95.92% 97.4% and 98.9% with 4hour daily and monthly datasets, respectively. ANN exhibited accuracies of 95.4% 98.0% and 99.05% with 4hour, daily and monthly datasets, while SVM attained 95.3% 97.5% and 99.04% accuracy.

A key strength of this research was the leverage of extensive datasets. However, a potential limitation was the utilization of only 8 attributes from 33 available CDR attributes. Additionally, the selected algorithms may not be ideally suited for handling sequential data, which could be a factor in SIMBox fraud detection.

TABLE 2.1: Summary of related works

Paper	Data	Area	Method	Result	Weakness	Strength
[23]	CDR	Subscription fraud	RF, SVM ANN WEKA tools	RF 99.46	Critical features such as IMEI and IMSE provide valuable information for understanding and managing mobile network operations, and they are essential for detecting and preventing various types of fraud in this study those are not part of the dataset collection. The accuracy is also close to perfection	
[25]	CDR	Subscription fraud	ANN, SVM, J48	J48 99.3, ANN 97.51, SVM 96.0	potential susceptibility of the model to overfitting due to sequential data collection of random days of CDR data.	various ML and deep learning algorithms as well as intense use of state-of-the-art feature selection, cleaning, integration techniques

[22]	CDR	Subscription fraud	J48, PART, RF, and ANN with WEKA	RF score good performance (99.9251)	Parameter tuning and efficiency is not well addressed	Highly relevant data is collected which resulted in high accuracy model
[9]	CDR SMS, GPRS, OCS, and CCB	SIM-box Fraud	J48, PART with WEKA tool	PART 100 J48 99.98	They Used WEKA tool and this tool mostly with its default parameters with less concern on performance tuning	The dataset is reliable and representative of real life situations. The accuracy of the algorithms is near 100
[8]	CDR	SIM-box Fraud	ANN, decision trees, and rule-based	good results from PART rule-based and hybrid (J48 and PART)	critical features such as requests for IMEI and other data are not part of the dataset collection.	applied various nature algorithms such as tree-based and rule-based types for model building as well as deep learning and hybrid algorithms

[4]	CDR	IRSF	He uses support vector machines(SVM) ANN and RF	RF 97.00	Didn't get enough fraudulent call dataset for the experiment only 3921	A high volume of dataset around 197,000 is collected from relevant sources

Chapter 3

Research Methods

3 Research Methods

3.1 Study Design

The CRISP-DM process model, a highly popular framework for data mining initiatives, is utilized in this study. As shown in figure3.1, CRISP-DM provides a structured approach with six phases: business understanding, data understanding, data preparation, modeling, evaluation, and deployment. This research adopts a comprehensive data mining approach to build a robust telecom SIMBox fraud detection system.

- **Business understanding** Defines the business objectives and the specific problems you are trying to solve with data mining. It identifies key stakeholders and their needs. Also, it assesses the feasibility of the project and the resources required.
- **Data understanding** Explores the available data sources to understand their content, quality, and completeness. It identifies potential issues like missing values, inconsistencies, and outliers and selects relevant data for further analysis.

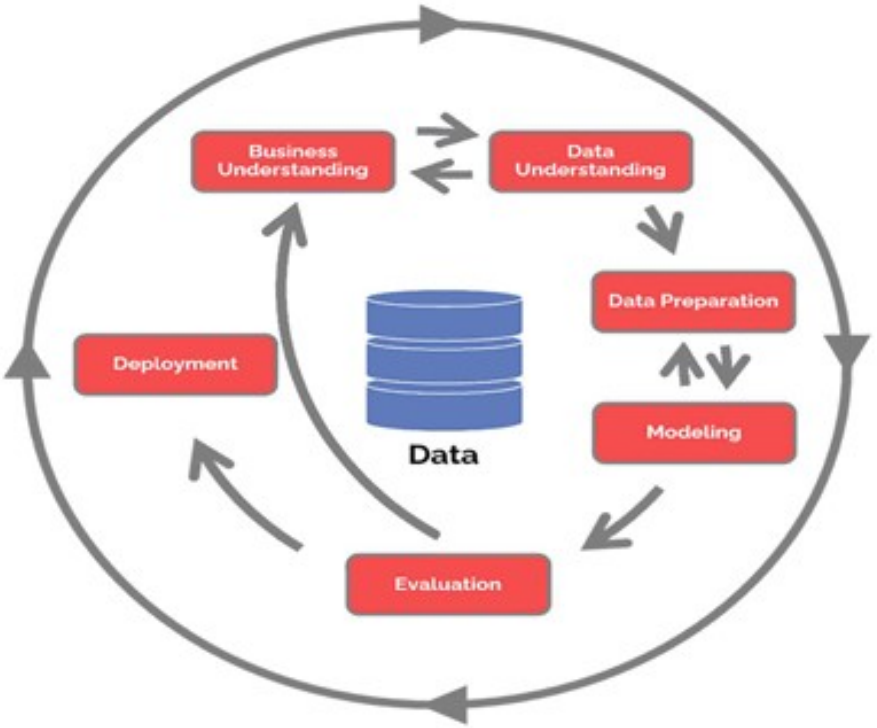

FIGURE 3.1: CRISP-DM Process diagram

- **Data Preparation** ensures that the data is clean and suitable for modeling by cleaning and preprocessing it. This might involve handling missing values, correcting inconsistencies, and formatting data so that it can be effectively analyzed. It might also involve feature engineering: creating new features from existing ones.
- **Modeling** Selects and applies appropriate data mining techniques based on the business problem and data characteristics and then evaluates the performance of different models based on chosen metrics.
- **Evaluation** Assesses the performance of the chosen model on unseen data to ensure its general usability.
- **Deployment** Integrates the chosen model into the business process for practical use.

3.2 Environment setup

To conduct the experiment, we leveraged a robust computing environment built on cloud infrastructure. This environment features a Linux server running Ubuntu 22.04 LTS,

equipped with eight virtual CPUs (vCPUs) for parallel processing, 80GB of RAM to handle large datasets and complex computations, and a 500GB storage node to accommodate the data and experiment results.

This research leverages Jupyter Notebook, a web-based application known for its intuitive interface and strong support for data science tasks. Jupyter Notebook's syntax highlighting simplifies code readability, while its built-in execution capabilities allow for efficient code development and testing. Additionally, we utilized Python as the primary programming language for building the deep learning model. We employed Python's PyTorch framework for deep learning model implementation. PyTorch offers a flexible and efficient platform for building and training deep learning models, and Pandas and NumPy libraries are utilized for data manipulation and analysis. These libraries are powerful tools for data cleaning, transformation, and exploration, which are crucial steps in preparing data for deep learning models.

The core of this study involves evaluating the effectiveness of various classification algorithms for telecom fraud detection. We explored the performance of established models like MLP (Multi-Layer Perceptron) and LSTM (Long Short-Term Memory) networks. We will incorporate BERT, a relatively new and powerful NLP (Natural Language Processing) technique, and rRNN (Recurrent Neural Network). These algorithms have a proven track record of success in sequential data analysis tasks, making them well-suited for analyzing call detail records (CDRs) commonly used in fraud detection. The researchers leveraged Hugging Face, an invaluable platform and community hub dedicated to accelerating advancements in Natural Language Processing[29]. It offers access to an extensive set of NLP datasets available to the public encompassing various tasks and languages, and it is also suitable for the BERT model.

3.3 Evaluation mechanism

To assess the effectiveness of the developed model in identifying SIMBox fraud, we employed accuracy metrics. Accuracy is a metric that measures how often a machine learning model correctly predicts the outcome. It can be calculated by dividing the number of correct predictions by the total number of predictions. Accuracy = correct predicitons/ total predictions When evaluating the accuracy, we looked at correct and

wrong predictions disregarding the class label. Correct predictions include so-called true positives and true negatives. This is how it unpacks for our spam use case example:

- True positive (TP):A call that is actually fraud and is correctly classified by the model as fraud.
- True negative (TN):A call that is actually not fraud and is correctly classified by the model as not fraud. Model errors include so-called false positives and false negatives. In our example:
- False Positive (FP):A call that is actually not Fraud but is incorrectly classified by the model as fraud (a "false alarm").
- False Negative (FN):A call that is actually fraud but is incorrectly classified by the model as not fraud (a "missed fraud").

3.4 Procedure

This research leveraged CDR data from Safaricom Ethiopia, containing both SIMBox fraud and normal call records. To prepare the data for deep learning analysis, we employed a multi-step pre-processing approach. This involves addressing missing values (removal or imputation) and transforming the data into a machine-readable format.The collected dataset has missing values and occasionally an entire feature column with empty columns, depending on the nature of the data(i.e nominal, ordinal) we used imputation technique which uses descriptive statistics methods of mean, mode and median to find the closest possible value for the missing feature by analyzing other feature values using the powerful pandas library as a backbone and keras core when necessary. In the mean time features with more than 50% of missing values are dropped since the data in hand is not conclusive or informative to most ML feature engineeering libraries and methods. For features which are not sensitive to outliers(standard deviation) to outliers can easily be filled using mean and others using mode and in some cases using median values. We used a more comprehensive feature selection method to identify the most significant attributes that differentiate fraudulent and legitimate call behavior. The pre-processed data is then fed into deep learning models trained to analyze call sequences and identify abnormal patterns indicative of fraudulent activities. Finally,

the effectiveness model was evaluated using accuracy metrics on a separate validation dataset

3.5 Data analysis

The analysis began with data pre-processing, cleaning, and feature engineering to ensure the CDR data is suitable for deep learning. This involved handling missing values, formatting timestamps. Next, the pre-processed data was split into training, validation, and testing sets. Deep learning models (potentially including LSTMs, MLP, or BERT, depending on the data characteristics) were trained on the training sets. Hyperparameter tuning was performed on the validation set to optimize model performance. Finally, the performance of the trained model was evaluated on the unseen testing set using accuracy metrics to assess its effectiveness in identifying fraudulent activities within CDR data.

3.6 Ethical concerns

This research prioritized users' privacy by ensuring the protection of their personal and sensitive information throughout the process. To achieve this, a critical step is taken: data anonymization.

The process of data anonymization is converting personally identifiable information (PII) into a format that makes it impossible to identify specific individuals. This means replacing PII data like names, phone numbers, or other details with non-identifiable substitutes. This process ensures that the data reflects patterns and trends without compromising user privacy.

In simpler terms, all collected data undergoes a transformation process where any information that could link a specific person to their activity is removed and replaced with alternative codes. This safeguards user privacy by preventing the identification of individuals or their actions within the anonymized dataset.

- **Data anonymization**: Data anonymization refers to transforming data to prevent individuals from being re-identified, typically achieved by replacing or suppressing sensitive personal information.

- **Confidentiality**: Ensuring the data is managed and kept securely to avoid unwanted access or disclosure.
- **Compliance with regulatory requirements**: Adhering to relevant laws and rules pertaining to privacy and data protection.

Throughout the process, we ensured user data, especially personal and sensitive information, remained strictly protected. This commitment to accountability and responsible data handling practices fosters trust and strengthens the credibility of the research findings.

The process of data anonymization is converting personally identifiable information (PII) in the case of telecom industry features such as phone number, Device ID addresses, names of any sort and other records and transforming into a format that makes it impossible to identify specific individuals. This means 27 replacing PII data like names, phone numbers, or other details with non-identifiable substitutes. This process ensures that the data reflects patterns and trends without compromising user privacy. We used random shuffler algorithm in the pandas library which shuffles the contents of each feature by applying rand() function and giving all feature values as a sample set.

Chapter 4

Proposed System

4 Proposed System

This study addresses SIMBox fraud detection for Safaricom Ethiopia, a new player in the Ethiopian market facing unique challenges due to the country's socio-economic and cultural landscape. To tackle this, we propose a contextualized deep learning model specifically designed for Safaricom Ethiopia. We follow the CRISP-DM methodology for data mining, ensuring a structured approach as shown in figure 4.1. Our data (12,274 entries with 80 attributes) reflects a balanced distribution of labeled and unlabeled data collected across different time intervals with Ethiopian call destinations and various data types.

For this thesis work, two separate datasets were acquired from Safaricom Ethiopia that are relevant to the problem at hand. The first one was a labeled SIMBox fraud dataset, which is a set of CDR records that Safaricom Ethiopia has recorded as SIMBox fraud, and the second was an unlabeled normal transaction that can be used to represent non-fraudulent transaction in the modeling phase. Both datasets contain around 80 features, excluding the class label, and around 6136 instances for each category. The CDR includes all of the call's details, including the base station (cell ID), destination location, duration, start and end times, duration of the call, amount charged, and subscriber service numbers and called numbers. Due to the coded nature of the dataset, interpretation of each feature required intense data cross-checking and domain expert consultation.

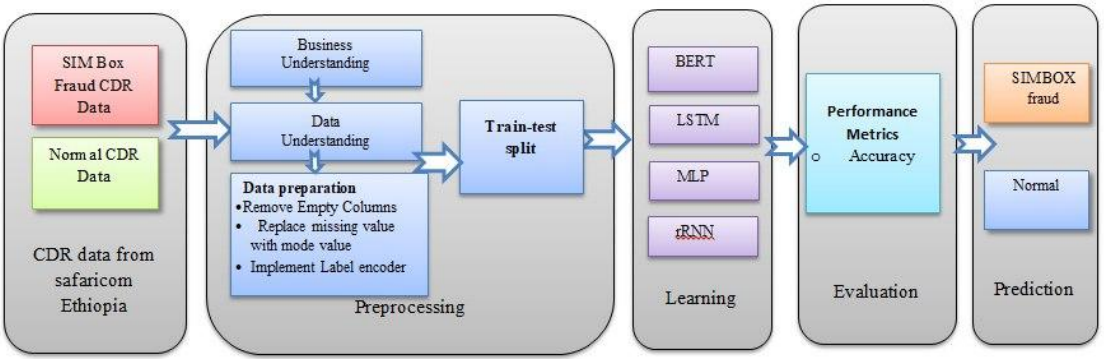

FIGURE 4.1: Proposed System

The Exploratory Data Analysis (EDA) phase helps understand the overall dataset in a more detailed fashion, such as how it looks, which features have missing values, how many missing values, data types of each column, outliers in features, mean, mode, standard deviation, and other statistical descriptions of the datasets. Finding attributes, looking at the data structure that corresponds to them, and assessing how important they are for describing a subscriber's pattern are the main goals of data understanding in the data mining process. Although the class label is balanced, the EDA showed us many inconsistencies in the dataset. There are empty features, outliers, incorrect data entries, and data type mismatches.

Some Machine Learning Operations (MLOps) experts stipulate that about 80% of model building work is spent on data pre-processing and cleaning. Real life data is barely clean and can't be used directly for ML model building, so rigorous data preprocessing tasks are applied. To further analyze the distribution of the data, we identified potential outliers within each feature. These are data points that fall significantly outside the typical range. We'll then calculate and present key statistical measures, including the mean (average), mode (most frequent value), and standard deviation (a measure of spread) for each feature. This comprehensive analysis will provide a rich understanding of the central tendencies and variability within the dataset. Features with missing values filled with mode values and inconsistent data points are replaced with average weights.

Due to computational cost of hardware at our disposal, we shall take the top ranking N elements of the data and based on correlation score assigned to each by the PCA(principal component analysis) algorithm,which then ordered in descending order As shown in **Table** 5.2, and set a threshold of 20 by seeing the PCA score value and with carefully consideration not to drop features that are above the margin of 0.5 which

generally reduces the computing time without serious compromising of the quality of data.

Once clean data is obtained, different modeling approaches are used. It is usual to return to the data preparation stage since different methods may call for different data formats. Choosing modeling techniques, creating tests, constructing the model, and evaluating the model are among the crucial responsibilities. After a detailed review of literature on available deep learning models for CDR based fraud detection, the researchers have chosen four deep learning algorithms model to detect sim-box telecom fraud.

A new transformer-based deep learning model called Bidirectional Encoder Representations from Transformers (BERT) is distinguished by the fact that all output elements are related to all input elements, and the weightings between them are dynamically determined based on their relationship. It is a new transformer-based model that uses joint conditioning of left and right contexts across all layers to pre-train deep two-way representations from unlabeled text[30]. Therefore, by adding only one more output layer to the pre-trained BERT model, state-of-the-art models for various tasks, such as language inference and question answering, can be achieved without requiring significant changes to the task-specific architecture. In contrast, a multi layer perceptron (MLP) is a feed-forward artificial neural network that is completely connected and comprises a minimum of three layers, namely input, output, and a hidden layer. Multilayer Perceptron's (MLPs) provide several benefits, including parallel data processing, fault tolerance, a strong architecture, and the capacity to learn and generalize. Thus, it makes it possible for them to resolve intricate non-linear and multiple input-output relationship issues. Their capacity for non-linear mapping, parallel processing techniques, environmental learning, and consequent environmental adaptation make them valuable in real-world applications as well. Many to one Recurrent Neural Network(M2RNN) is the third DL model used in this thesis. A type of artificial neural network, the Recurrent Neural Networks (RNNs), are made to analyze data sequences for tasks including natural language processing, speech recognition, time series data processing, and other applications. RNNs forecast a layer's output by reusing its saved output from a specific layer and putting it back into the original layer.

A unique RNN-based approach called long short term memory (LSTM) can preserve long-term dependencies in sequential data. LSTMs can process and interpret a variety

of sequential data types, including text, audio, and time series[31]. The Long Short Term Memory (LSTM) is a highly optimized RNN-based neural network with three gates, an input gate, an output gate, and a forget gate. Those three gates guide the entry and exit of data through the cell, and the cell remembers values across random time intervals. A value between 0 and 1 is used by forget gates to specify which data from a previous state should be ignored in light of the current input[32].

Given the dataset's inherently balanced nature, the accuracy score is the primary metric for model evaluation. It effectively captures the model's performance across all classes and provides a comprehensive assessment of its overall correctness, thereby ensuring a robust evaluation of its capabilities.

Chapter 5

Experiments and Analysis

5 Experiments and Analysis

5.1 Data understanding

The data science process begins with meticulous data collection and familiarization. This initial stage lays the groundwork for successful analysis. It involves gathering the initial data, describing data, exploring data, and verifying data quality. Here, the focus is on extracting knowledge from the target data. This entails a comprehensive understanding of the underlying raw data.

5.2 Gathering initial data

In this research, we have acquired two separate datasets from Safaricom Ethiopia that are crucial in identifying SIMBox fraud. The first dataset is labeled and specifically identifies SIMBox fraud instances. The second dataset is unlabeled and represents normal transactions, serving as a benchmark for non-fraudulent behavior during the modeling phases. An extensive examination of the structure and substance of the data is essential.

Our research delves into a rich dataset composed of eighty attributes. Our primary objective at this stage is to discern the most valuable fields within this data and assess the significance of the information they hold. By thoroughly understanding these attributes,

No	Dataset	Number of Instances	Number of Attributes
1	SIMbox dataset	6,136	80
2	Unlabeled dataset	6,138	80

TABLE 5.1: Collected data

```
simbox.columns
```

```
Index(['Unnamed: 0', 'Unnamed: 1', 'Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4',
       'Unnamed: 5', 'Unnamed: 6', 'Unnamed: 7', 'Unnamed: 8', 'Unnamed: 9',
       'Unnamed: 10', 'Unnamed: 11', 'Unnamed: 12', 'Unnamed: 13',
       'Unnamed: 14', 'Unnamed: 15', 'Unnamed: 16', 'Unnamed: 17',
       'Unnamed: 18', 'Unnamed: 19', 'Unnamed: 20', 'Unnamed: 21',
       'Unnamed: 22', 'Unnamed: 23', 'Unnamed: 24', 'Unnamed: 25',
       'Unnamed: 26', 'Unnamed: 27', 'Unnamed: 28', 'Unnamed: 29',
       'Unnamed: 30', 'Unnamed: 31', 'Unnamed: 32', 'Unnamed: 33',
       'Unnamed: 34', 'Unnamed: 35', 'Unnamed: 36', 'Unnamed: 37',
       'Unnamed: 38', 'Unnamed: 39', 'Unnamed: 40', 'Unnamed: 41',
       'Unnamed: 42', 'Unnamed: 43', 'Unnamed: 44', 'Unnamed: 45',
       'Unnamed: 46', 'Unnamed: 47', 'Unnamed: 48', 'Unnamed: 49',
       'Unnamed: 50', 'Unnamed: 51', 'Unnamed: 52', 'Unnamed: 53',
       'Unnamed: 54', 'Unnamed: 55', 'Unnamed: 56', 'Unnamed: 57',
       'Unnamed: 58', 'Unnamed: 59', 'Unnamed: 60', 'Unnamed: 61',
       'Unnamed: 62', 'Unnamed: 63', 'Unnamed: 64', 'Unnamed: 65',
       'Unnamed: 66', 'Unnamed: 67', 'Unnamed: 68', 'Unnamed: 69',
       'Unnamed: 70', 'Unnamed: 71', 'Unnamed: 72', 'Unnamed: 73',
       'Unnamed: 74', 'Unnamed: 75', 'Unnamed: 76', 'Unnamed: 77',
       'Unnamed: 78', 'Unnamed: 79'],
      dtype='object')
```

FIGURE 5.1: Describing number of attributes

we can identify the most impactful features for building a robust model capable of detecting SIMBox fraud.

We were working with two datasets related to SIMBox fraud detection: a labeled dataset and an unlabeled dataset. as shown in the Table 5.1, the labeled dataset contains 6,136 data points, each with 80 attributes.

These attributes represent various features extracted from call detail records (CDRs) that can be used to identify fraudulent activity. However, As shown in Figure 5.1, the labels associated with these attributes in the SIMBox fraud dataset are not very descriptive.

```
normal.describe()
```

	ENTITY	STARTTIME	STARTTIME_DAY	STARTTIME_HOUR	DURATION	MSISDN	MSRN	CALL_TYPE	CALL_CATEGORY	
count	6.138000e+03	6138	6138.0	6138.0	6138.000000	6.138000e+03	2.870000e+02	6138.000000	6138.000000	5
mean	5.805125e+14	2024-03-20 08:53:48.825349888	20.0	8.0	12.933855	2.517281e+11	2.517000e+11	212.432877	0.942489	6
min	2.517010e+11	2024-03-20 08:50:46	20.0	8.0	0.000000	2.517002e+11	2.517000e+11	1.000000	0.000000	6
25%	6.360201e+14	2024-03-20 08:52:01	20.0	8.0	0.000000	2.517100e+11	2.517000e+11	3.000000	0.000000	6
50%	6.360201e+14	2024-03-20 08:53:31	20.0	8.0	0.000000	2.517135e+11	2.517000e+11	303.000000	1.000000	6
75%	6.360201e+14	2024-03-20 08:55:20	20.0	8.0	5.000000	2.517165e+11	2.517000e+11	303.000000	1.000000	6
max	6.450200e+14	2024-03-20 08:59:23	20.0	8.0	424.000000	2.609662e+11	2.517000e+11	802.000000	3.000000	6
std	1.795022e+14	NaN	0.0	0.0	33.505999	2.243216e+08	5.036742e+03	252.222540	0.701577	2

FIGURE 5.2: describing SIMBox data

5.3 Describing Initial data

In contrast, the unlabeled dataset, also containing 6,138 data points with 80 attributes, has features with clear and descriptive names. This distinction in labeling detail will likely require different approaches when working with each dataset.

This section explores the dataset in greater detail, giving a thorough explanation of its organization and characteristics. We'll begin by examining the format of the data, offering insights into how it's visually organized. Next, we'll identify features (columns) that contain missing value. Additionally, we'll delve into the data types associated with each column, ensuring a clear understanding of the kind of information they hold (e.g., numbers, text).

To further analyze the data's distribution, we'll identify potential outliers within each feature. These are data points that fall significantly outside the typical range. As shown in **Figure** 5.2, we will then calculate and present key statistical measures, including the mean (average), mode (most frequent value), and standard deviation (measure of spread) for each feature. This comprehensive analysis will provide a rich understanding of the central tendencies and variability within the dataset.

There is a data quality concern within our SIMBox dataset. A significant number of attributes seem to be missing values. These attributes, identified by their common name (unamed 20, 37, 40, etc.), are completely empty in the SIMBox data. Additionally, the normal dataset is also missing values for several attributes, including SESSION ID, TARIFFID, and RULEBASENAME.

Interestingly, in both datasets, most of the data types appear to be in objects, with a few exceptions of integers and date entries Based on this observation and domain knowledge of database experts at Safaricom Ethiopia and explorative data analysis(EDA) are the two mechanisms that are implemented to justify the correctness of the mapping of columns. we propose copying the attribute names from the normal dataset to the SIMBox dataset. This approach assumes that both datasets share similar structures and the missing attributes in the SIMBox data might have corresponding entries in the normal dataset.

we encountered a data quality issue where 18 attributes were empty in both datasets we planned to merge. To address this, we took several steps. First, I added new columns labeled 'SIMBox' and 'normal' to indicate whether each data point belonged to the SIMBox fraud category or the normal category. This provided labels for classification. Next, I merged the two datasets into a single, unified dataset. Finally, to ensure data quality and focus on relevant features, we identified and dropped the 18 columns that were consistently empty across both datasets. This resulted in a clean dataset with 12,274 rows (data points) and 63 usable columns (features) ready for further analysis. Identifying key features with the highest decisive scores for normal and SIMBox call data classification is crucial for the model's effectiveness. As shown in**Figure** 5.3, both datasets might appear visually similar.

As shown in **Figure** 5.4 the context of analyzing call data records (CDRs), this might signify that a higher volume of calls is being placed to destinations within Ethiopia compared to other country's. This indicates that the data relevant to our research

The other thing, as shown **Figure** 5.5 time related feature how to describe time -series data in the graph, the variety of collected CDR data in the different time intervals in number of records is valuable for our paper.

5.4 Data preparation

The unstructured and disorganized nature of real-world data makes it unsuitable for direct use in machine learning models. Machine learning's main challenge is determining a representative set of characteristics to construct a classification model for a particular job. Recent studies have demonstrated the impact of duplicate and unnecessary training

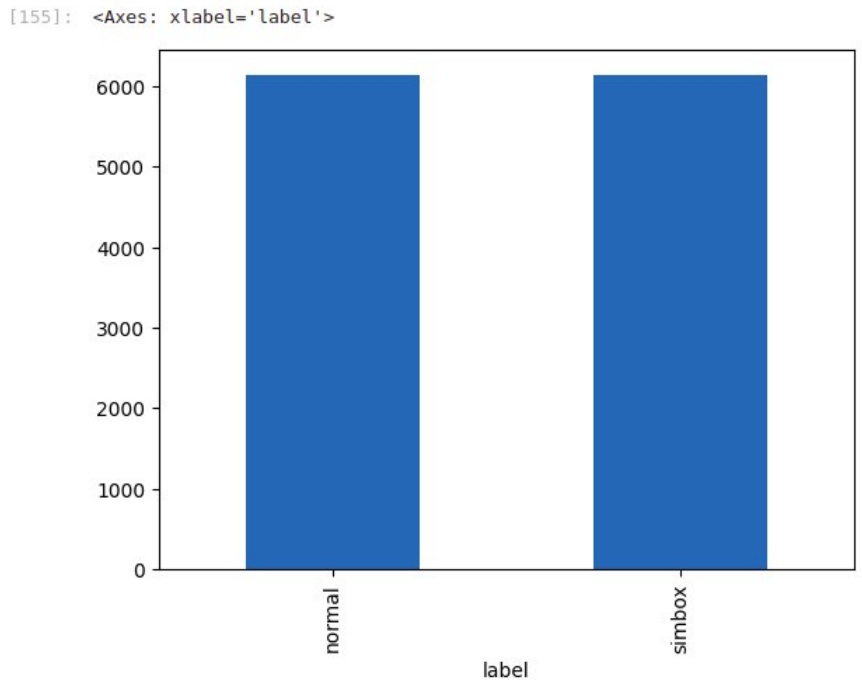

FIGURE 5.3: count the data categorized to normal and SIMBox

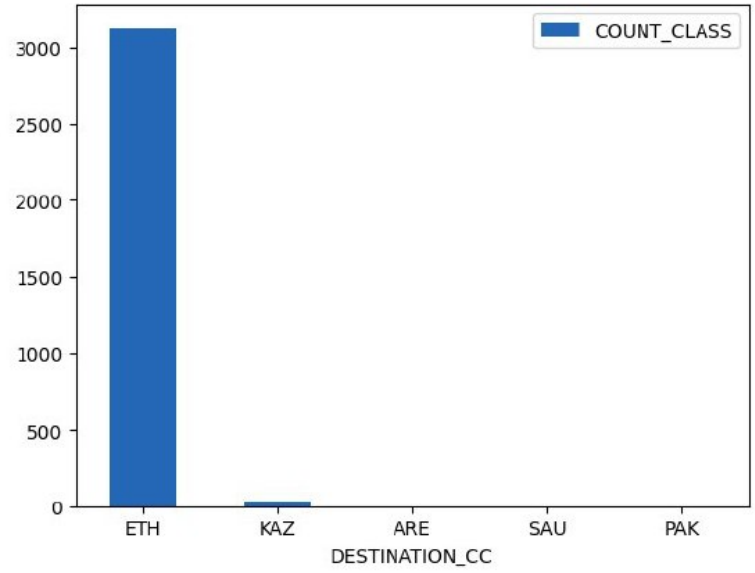

FIGURE 5.4: The most call destination

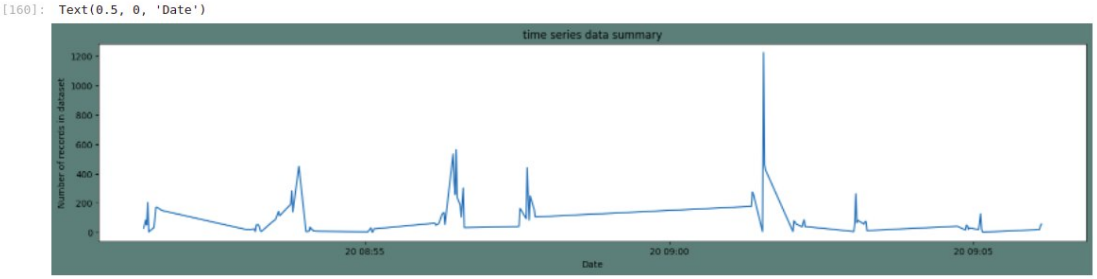

FIGURE 5.5: Time series data description

```
for column in final.columns:
    final[column] = final[column].fillna(final[column].mode()[0])
```

FIGURE 5.6: Replace missing value with mode value

data on machine learning algorithms. Learning becomes more challenging during the training phase if there is an excessive amount of redundant and irrelevant information available or if the data is noisy and inaccurate[33]. Processing raw data is necessary for data preparation because it allows machine learning algorithms to produce a structured representation of the information that is latent in the data. It defines, processes, and makes suitable data mining techniques. It is a crucial initial step in the data mining process and affects the outcome of the complete procedure.

This phase, crucial for successful modeling, tackles these challenges. We meticulously clean the data to address inconsistencies, missing values, and formatting errors. Next, we select relevant features that hold the most significant information for the modeling task. Feature engineering might also be employed, where we create new informative features by combining existing ones. Once cleaned and refined, the data from various sources is integrated into a cohesive dataset. Finally, the data is formatted into a structure that the chosen machine learning model can readily understand and process.

Let's address the remaining missing values in our dataset by imputing them with the most frequent value (mode) within each column. We can achieve this using a loop that iterates through each column in the final dataset. As shown in**Figure** 5.6 inside the loop, we can calculate the mode for the current column and then replace all missing values within that column with the calculated mode.

In order to set up the data for the machine learning model to analyze, for the model to understand, all categorical and string data must be converted into a numerical format.

```
from sklearn.preprocessing import LabelEncoder
import pandas as pd
le = LabelEncoder()
final.apply(LabelEncoder().fit_transform)
```

	SOURCE_TYPE	SOURCE_FILE	ENTITY	STARTTIME	STARTTIME_DAY	STARTTIME_HOUR	DURATION	DESTINATION	DESTINATION_CC	MSISDN	MSRN	CALL_
0	0	74	1225	3	0	0	60	1183	1	4863	262	
1	0	74	5117	5	0	0	0	2446	1	1616	15	
2	0	74	5483	8	0	0	0	2446	1	1870	15	
3	0	74	6072	18	0	0	0	2441	1	1252	15	
4	0	74	2418	20	0	0	0	4	1	527	15	
...	...	...	...	...	...	...	...	...	...	...	...	
6133	1	64	3548	337	0	0	0	0	1	2629	15	
6134	1	61	2659	342	0	0	0	0	1	1368	15	
6135	1	61	2659	342	0	0	0	0	1	1368	15	
6136	1	73	6376	345	0	0	0	0	1	3958	15	
6137	1	73	6376	345	0	0	0	0	1	3958	15	

12274 rows × 63 columns

FIGURE 5.7: Transform all string and other data type to encoded format

This process is called encoding. we utilized the LabelEncoder function to achieve this. As shown in **Figure** 5.7 LabelEncoder assigns a unique integer value to each unique category within the data. For example, a category like **'SIMbox'** might be encoded as the number 0, while **'normal'** might be encoded as 1. This transformation ensures that the model treats each category consistently and allows it to learn relationships between these categories and the target variable we're trying to predict.

Let's delve into identifying the most influential features that contribute to the class label in our data. To achieve this, we'll utilize Principal Component Analysis (PCA). However, before applying PCA, it's crucial to standardize our data using a standard scalar function. By ensuring that all of the features are on the same scale, standardization keeps the analysis from being dominated by features with higher values.

PCA is a powerful statistical technique that simplifies complicated datasets by transforming them into a new set of uncorrelated variables called principal components. These components are ordered based on the amount of variance (information) they capture from the original features. The first principal component captures the most significant variations within the data, while subsequent components account for progressively less variability.

By employing PCA, we aim to reduce the overall dimensionality of the data while retaining the most critical patterns and relationships between the features. As shown in **Figure** 5.8, this dimensionality reduction not only improves computational efficiency

```
from sklearn.decomposition import PCA

pca = PCA(n_components = 20)
components = pca.fit_transform(x)
components_df = pd.DataFrame(components)
df_pca_loadings = pd.DataFrame(pca.components_)
df_pca_loadings.head()
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	
0	0.268306	-0.270350	0.034184	-0.086997	0.000000e+00	1.058791e-22	-0.118353	-0.237553	-0.014466	-0.003430	-0.047198	-0.171735	-0.272785	-0.006500	-0.0
1	-0.131638	0.101026	-0.016423	0.265433	1.734723e-18	-0.000000e+00	0.070893	-0.040647	-0.002267	-0.007611	0.033219	-0.041869	0.039372	-0.002286	-0.0
2	0.067597	-0.060948	-0.016256	-0.135168	1.734723e-18	-0.000000e+00	0.211550	-0.226434	-0.025761	-0.022320	0.097226	-0.286823	0.088828	-0.018316	-0.0
3	-0.017023	0.013903	-0.438094	-0.221458	-0.000000e+00	-0.000000e+00	0.201425	0.003371	-0.015116	-0.062027	0.050557	0.214452	0.099297	-0.297056	-0.0
4	-0.114556	0.100938	0.306944	-0.282455	-6.938894e-17	5.551115e-17	-0.098881	0.069206	0.026616	0.101943	0.006849	-0.007632	-0.003107	0.274819	0.0

FIGURE 5.8: Feature Selection with Correlation Analysis

but also helps us identify the top 20 features that have the strongest influence on the class label we're trying to predict.

For comparison purposes, let's explore using the filter method for feature selection. This method, as the name suggests, involves filtering out irrelevant features and keeping only a subset of the most decisive ones. The model is then built using this reduced feature set. A popular filtering method uses the correlation matrix—Pearson correlation in particular. This coefficient measures the linear bond between two variables. It ranges from -1 to 1, where: a value close to 0 indicates weak or no correlation. Values closer to 1 indicate a powerful positive correlation, meaning both variables tend to move in the same direction. Values closer to -1 indicate a powerful negative relationship, meaning the variables move in opposite directions. In our case, we will first visualize the correlation of the autonomous elements (features) to the selected elements using a Pearson correlation heatmap. This heatmap will help us identify features with a strong correlation (absolute value above 0.2) with the target variable. These features are likely to be the most informative for predicting, and we will select them for model building. As shown in **Table** 5.2, our analysis reveals 18 features exceeding the predefined threshold for a decisive score. These features likely have a significant influence on predicting or classifying the data.

5.5 Modeling

Arranging Once clean data is obtained, different modeling approaches are used. Each method may require specific data formats, so looping back to the data preparation phase is not uncommon. The crucial tasks are choosing modeling techniques, creating tests,

No	Index	Score
17	STORED	0.798531
16	ZERO˙FLAG˙VOLUME	0.225602
15	DATA˙FLOW˙TIME	0.873671
14	CALL˙CLASS	0.306370
13	VOLUME	0.223751
12	IMEI	0.208157
11	CALLED˙NUMBER	0.237228
10	SERVICE˙STRING	0.210570
9	SERVED˙GGSN˙ADDRESS	0.320313
8	CAUSE˙FOR˙TERM	0.548444
7	EDR˙ID	0.573148
6	CALL˙REFERENCE	0.281299
5	CALL˙CATEGORY	0.287488
4	CALL˙TYPE	0.206010
3	DESTINATION	0.283134
2	STARTTIME	0.852266
1	SOURCE˙FILE	0.442304
0	SOURCE˙TYPE	0.492017

Table 5.2: top decisive features for class labeling

constructing the model, evaluating the mode, etc. After a detailed review of literature on available deep learning models for CDR based fraud detection, the researchers have chosen the following deep learning algorithms for modeling SIMBox fraud detection.

A transformer-based deep learning model known as Bidirectional Encoder Representations from Transformers, or BERT for short, is characterized by each output element being connected to every input element and having weightings between them that are dynamically determined based on that relationship.

This unique transformer-based approach uses simultaneous left and right context training across all layers to pre-train deep bidirectional representations from unlabeled text.

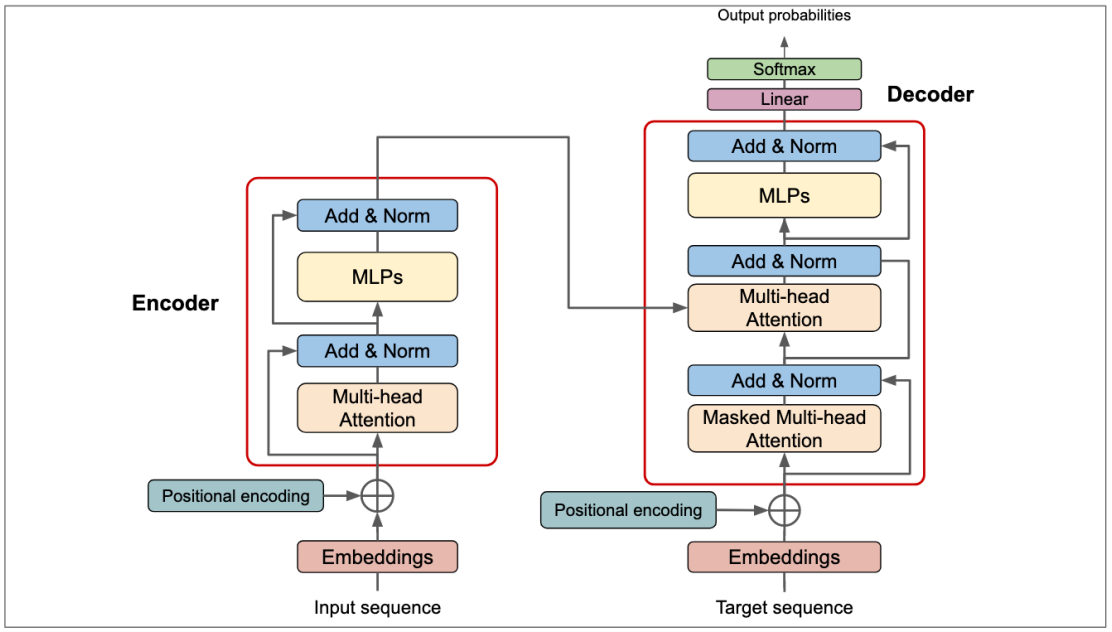

FIGURE 5.9: Transformer based Model architecture[34]

Therefore, the pre-trained BERT model can be further enhanced with just one more output layer to build state-of-the-art models for multiple tasks, such as language inference and question-answering, without requiring large task-specific architecture adjustments.

As shown in **Figure** 5.9, the two main stages of the model's design are pre-training and fine-tuning. Pre-training involves training the model on unlabeled data from various pre-training assignments. Subsequently, labeled data from the downstream tasks is used to fine-tune all of the parameters in the BERT model after it has been initialized with the pre-trained parameters. Despite being started with identical pre-trained parameters, each downstream task has its own fine-tuned models.

One of the most distinctive features of BERT is its integrated architecture, which is used throughout various jobs. There is negligible variation between the pre-trained architecture and the corresponding downstream architecture. **Multi layer perceptron(MLP):-** A multilayer perceptron (MLP), in contrast, is a feed-forward artificial neural network that is completely connected and comprises a minimum of three layers as shown **Figure** 5.11 (input, output, and at least one hidden layer).

The most basic type of artificial neural network is a perceptron, which is made up of a single neuron that can process numerous inputs and generate a single output. In this architecture, classes that are linearly separable are classified using perceptrons. As can

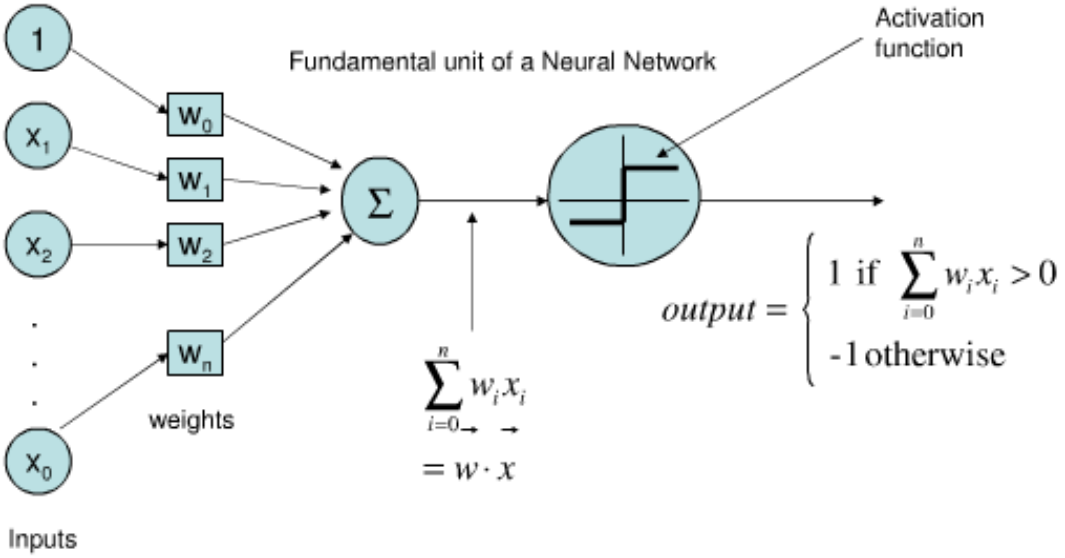

FIGURE 5.10: Fundamental Unit of neural network[35]

be seen in the above image, it receives a vector of real-valued inputs, uses the chosen function to calculate a linear combination of these inputs, and outputs 1 if the result is greater than a threshold and -1 otherwise. Finding a weight vector that influences the perceptron to produce the proper output for each training example is the precise learning problem in this design. Running the perceptron algorithm repeatedly over the training set until it generates an assumed vector that is correct on all training sets is the most often used technique for utilizing the algorithm to learn from a batch of training samples. Next, the labels on the test set are predicted using this prediction rule.

$$\sum_{j=1}^{m} Wj.Xj = W1.X1 + ... + Wm.Xm$$

The weighted sum (WS) in this design is computed using the following equation, which is then evaluated and sent to an activation function (such as the sigmoid) for comparison with a predefined threshold Q(theta). The perceptron fires and outputs 1 if the weighted total is greater than the threshold Q; if not, it outputs 0 (-1). While many different activation functions may be employed with the perceptron, the most commonly utilized ones are the step, sign, linear, and sigmoid functions.

Step

$$f(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x < 0 \end{cases}$$

Sign

$$Sign(f(x)) = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{if } x < 0 \end{cases}$$

Linear

$$f(x) = x$$

Sigmoid

$$f(x) = \frac{1}{1 + e^{-x}}$$

At a threshold Q = 0, the aforementioned activation functions are activated. Nonetheless, having a threshold other than 0 is more practical. To do this, a bias Q is added to the inputs of the perceptron. Q's primary function is to modify the activation threshold by shifting the threshold function to the right or left. Finding the ideal weights and bias value at which the perceptron fires is the goal of training the perceptron.

Perceptrons are universally useful for solving any classification problem for classes that are linearly separable. A single-layer perceptron may not be able to solve the problem if it is presented with two nonlinearly separable classes. The multilayer Perceptron (MLP) is used to solve such a nonlinearly separable problem. Each perceptron receives a set of inputs from other perceptrons in a multilayer perceptron (MLP) neural network. Whether the weighted sum of the inputs is greater than a predetermined threshold determines whether the perceptron fires or not. On the other hand, nodes in the hidden layer help information move from the input to the output layers. Weight variables connected to each connector regulate the flow. Nodes that reflect the system's classification choice make up the output layer. To ascertain the output and categorize each case, the values of the output nodes are compared with limits[37].

Some advantages of Multi layer Perceptron (MLP) are their robust architecture, fault tolerance, it is the inherent ability to learn and generalize adaptability, and parallel data processing. As such, it makes it possible to resolve intricate non-linear and multiple

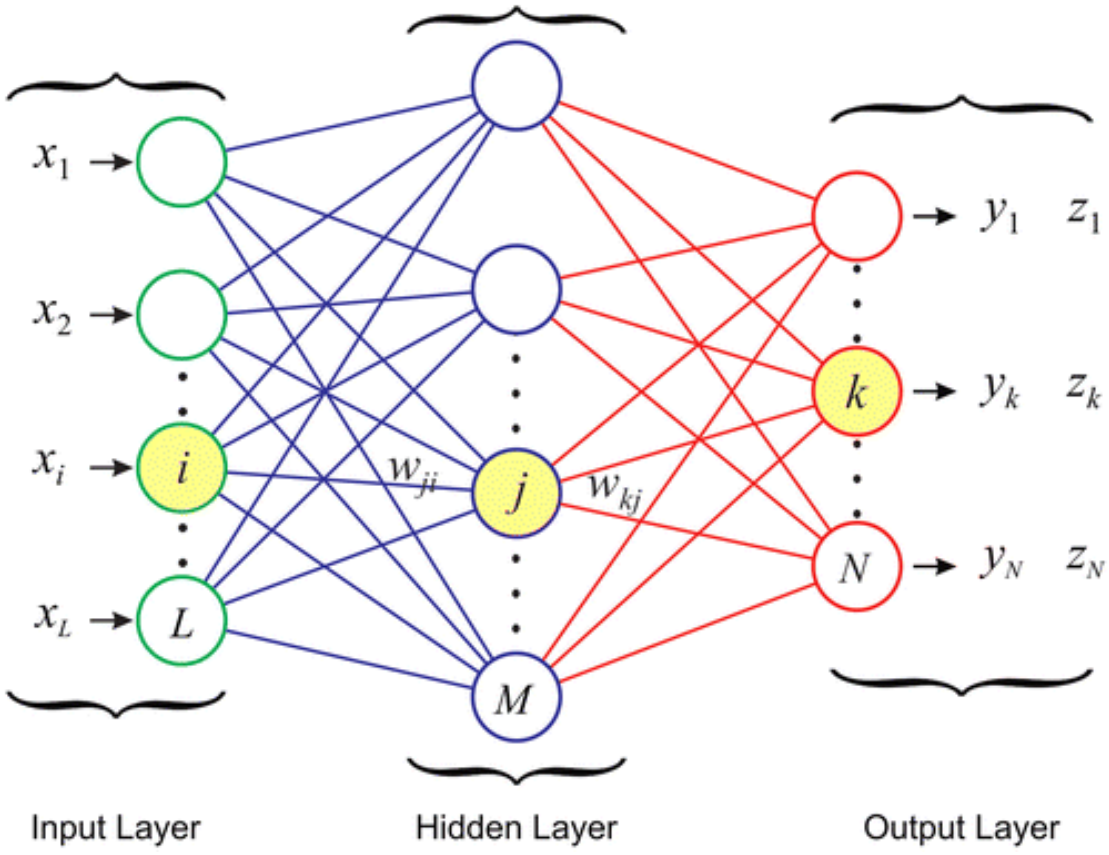

FIGURE 5.11: Multi layer Perceptron[36]

input-output relationship issues. Its capacity for non-linear mapping, parallel processing methods, learning capacity, and ensuing environmental adaptability also make it valuable in real-world applications[37]. M2RNN, or multiple-to-one recurrent neural network; Recurrent neural networks (RNNs) are a form of artificial neural network designed to process sequences of data for various tasks such as voice, natural language, time series data, and more. RNNs work by storing a layer's output and sending it back into the input layer in order to forecast the layer's output[37][38].

Generally speaking, recurrent neural networks (RNNs) are a kind of neural network design employed to find patterns in a data series. RNNs are typically used in speech recognition, picture description or video tagging, language modeling, text generation, and other fields. The way information travels through a recurrent neural network is different from that of feedforward neural networks, like Multi-Layer Perceptrons (MLPs). A feedforward network transmits data through its network without cycles, whereas an RNN transmits data back into itself and has cycles. This allows them to expand the capabilities of feedforward networks beyond just considering the current input Xt to

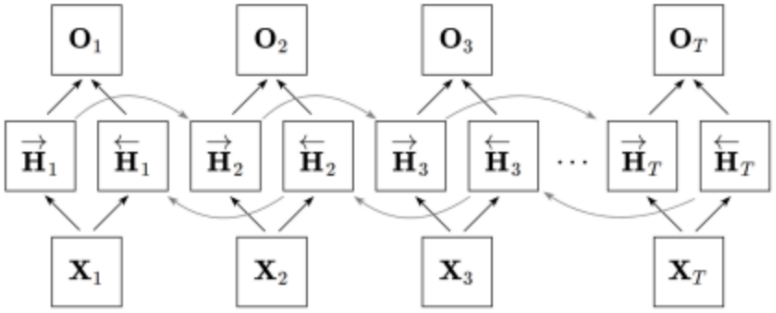

FIGURE 5.12: Architecture of rRNN[38]

include past inputs X0:t1[38].

The hidden state and the input at time step t, respectively, can be denoted with the mathematical notation Ht R n×h and Xt R n×d, where n is the representation of the number of samples, d is the number of inputs of each sample, and h is the number of hidden units, to represent the process of transmitting data to the hidden layer from the input layer. Thus, we employ a bias parameter, bh R 1×h, a weight matrix, Wxh R d×h, and a hidden-state-to-hidden-state matrix, Whh R h×h. All of this data eventually flows to an activation function , which often prepares the gradients for use in backpropagation and is a logistic sigmoid or tanh function. Equstion 1 is the hidden variable, and Equation 2 is the output variable when all of these notations are added together. Ht = h (Ht1Whh + bh + XtWxh)As shown **Figure** 5.12

From what we have observed thus far, we can confidently anticipate the subsequent sequence element using our existing models. But in certain situations, we might want to complete a statement where there is a gap and the sentence's remaining portion provides important information. To perform successfully on this kind of work, consideration of these knowledge is required. We would like to add a look-ahead property for sequences on a broader scale.

5.6 Evaluation

Accuracy has been one of the commonly used evaluation criteria for various machine learning and deep learning algorithms for a long time. However, when working with unbalanced class datasets in situations where there is a notable disparity between the

		Predicted	
		Positive	**Negative**
Actual	**Positive**	TP	FN
	Negative	FP	TN

FIGURE 5.13: Confusion Matrix Value Mapping

amount of positive (fraud) and negative (non-fraud) labels, accuracy alone may not always provide the whole picture. This metric is associated with the number of positive predicted true or false, or True Positive and False Positive (TP / FP), as well as the number of negative predicted true or false, or True Negative and False Negative (TN / FN). One way to gauge how well machine learning performs in categorization is by using a confusion matrix, as shown in Table 5.13below.

- **True Positive (TP):** Real SIM-Box cases that fell under the SIM-Box case classification.
- **False Positive (FP):** Regular clients who were categorized as SIM-Box cases.
- **False Negative (FN):** Real SIM-Box cases that were categorized as Regular clients.
- **True Negative (TN):** Normal cases that met the criteria for classification as normal.

The classifier's correctly predicted results are represented by the values TP and TN of the confusion matrix, while the wrongly predicted results are represented by the values FP and FN. The following detailed discussion was held over the suggested performance assessment measures.

The main evaluation criterion for experiments in this thesis is classification accuracy. The percentage of TP and TN to all analyzed cases is used to compute accuracy. Mathematically, the following can be stated

$$Accuracy = TP + TN/TP + FP + FN + TN$$

Chapter 6

Result and Discussion

6 Result and Discussion

The ability to detect telecom fraud has improved over time. However, as technology is always developing, there seems to be a constant process of progress because fraudsters may always find a way around the restrictions. Our goal in this thesis has been to identify CDR attributes that have the strongest influence on classifying fraudulent activity and develop a model for SIMBox fraud detection for newly established Safaricom Ethiopia that enables the operators to solve one of the most serious fraud types.

We employed Principal Component Analysis (PCA) to efficiently analyze the data. With PCA, we can reduce the number of features while retaining the most critical information relevant to predicting the class label. Through this process, we identified the top 20 features with the strongest influence on our prediction. To identify the most influential features, we employed the filter function in combination with Pearson correlation, and the method evaluates the linear connection between two variables by computing a coefficient that falls between -1 and 1, providing a robust measure of the correlation between variables.

The other thing in our analysis revealed that the Recurrent Neural Network (RNN) achieved the highest overall accuracy, reaching 99.7% in identifying fraudulent activities within the CDR data. This suggests that the RNN effectively captured the sequential nature of call data, allowing it to recognize patterns indicative of SIM box fraud.

	#	Model Name	Hidden Layers	Accuracy	Rank
0	1	BERT	2	98.6	3
1	2	MLP	2	97.6	4
2	3	LSTM	2	99.1	2
3	4	rRNN	2	99.7	1

FIGURE 6.1: Selected model Accuracy

Following closely behind was the Long Short-Term Memory (LSTM) model with an accuracy of 99.1%. LSTMs are known for their ability to learn long-term dependencies within sequences, which might explain their strong performance in this context. The model of natural language processing's accuracy, BERT, was 98.6%.

The Multi-Layer Perceptron (MLP) exhibited the lowest accuracy of 97.6% among the tested models. MLPs are generally less adept at handling sequential data compared to RNNs and LSTMs. This emphasizes the crucial importance of selecting models appropriate for the particular features of the data being examined.

Since we used a balanced set of class labels for SIMBox fraud and normal datasets, the accuracy score is efficient enough to evaluate the selected models. Although BERT has developed quite a reputation lately for solving complex language processing tasks with its in built capability of dynamic computation to identify discern context for better results in search queries and its concealed weapon on adaptive nature, the researcher's hypothesis of BERT having a chance to out score other deep learning algorithms has proven to be wrong.

Considering computational resource consumption, the researchers didn't observe any major differences. The main reasons could be the researchers applying similar code implementations for each with 2 hidden layers with similar activation functions whenever needed. On the other hand, the dataset used in the experimentation is much less complex in nature which has significantly reduced the time needed for the model to complete the learning process. Finally, we applied a comprehensive data pre-processing and feature

extraction tasks which, in the researcher's opinion, has simplified the model building process.

Chapter 7

Conclusion and Future work

7 Conclusion and Future work

7.1 Conclusion

Tariffs from international calls are used by telecommunications companies in emerging nations to offset part of their expansion costs. Nevertheless, SIM-Box scammers take advantage of this situation by offering callers lower costs while stealing money from the carriers. They can divert an international call and have it returned as a local call because of VoIP technology, SIM-Box, and local SIM cards. Operator revenue loss can be minimized by implementing a technique that identifies SIM-Box frauds early on and prevents the fraudsters from generating money.

Safaricom Ethiopia has become the second service provider to operate in Ethiopia. Although Safaricom is a highly experienced telecom service company, due to various unforeseen factors such as ethnolinguistic, cultural, and socio-economic distinct to Ethiopia, they will certainly be tested to its limits with a plethora of fraudulent activities which necessitates a proactive measure by building more contextualized fraud prediction model for Safaricom Ethiopia specifically.

In this research, we have collected fraudulent CDR and normal CDR logs from Safaricom Ethiopia and analyzed the data after rigorous data preparation and pre-processing steps relevant features are to distinguish SIM-Box fraud from legitimate subscribers. More

focus should be placed on the CDR features that display the helpful user profile. For the study to be successful, a thorough understanding of the fraud type's behavior is essential. Data from mobile devices is being created more quickly than ever before, and this data contains important information regarding scams.

Using principal component analysis(PCA) techniques, we have identified top ranking features using 0.2 as a threshold after performing a detailed survey of current and relevant literature in the domain. We have identified 4 deep learning algorithms of BERT, MLP, LSTM, and rRNN (classical) for model building. The results show that the rRNN algorithm showed the highest accuracy of 99.7% followed by LSTM, BERT, and MLP at 99.1%, 98.6%, and 96.7% of accuracy respectively.

7.2 Future Work

The study's findings have yielded encouraging outcomes. To detect interconnect bypass fraud in almost real time and lower the false alarm rate, more model improvement is required.

- If the subscriber data is accessible, further refining its features—such as adding the customer's demographics—might enhance the technique's accuracy and effectiveness. The CDR data belonging to the legitimate subscriber may contain scammers.
- Other telecom fraud types, such as International Revenue Sharing Fraud, subscription Fraud, Wangiri fraud, etc., are more common and if the data is available, those kinds of frauds must be addressed in future works.
- Although the research emphasises on application of CDR log data for SIMBox fraud detection for Safaricom Ethiopia, the researchers propose to fine tune the model with other datasets and use for realtime application of various telecom fraud related detection tasks. Hosting platform for BERT models(i.e huggingface) can be uitilized for this proposal in a near future.

  There are many different kinds of fraud in the telecom sector, and as some of them are facilitators or often linked to SIMBox fraud,It is recommended that comparable research be done on the other fraud categories.

References

[1] João Vitor and Cepêda De Sousa. Telecommunication fraud detection using data mining techniques, 2014. URL https://paginas.fe.up.pt/~ee09299/documents/ReportforJuryEvaluationCepeda.pdf.

[2] P D K E Wickramasinghe, K.G.D.C Kehelwala, H.M.N.D Bandara, R.A Yasaratne, P De Almeida, and I.K.K.S Ilesinghe. Real-Time Grey Call Detection System Using Complex Event Processing. *Annual Conference IET Sri Lanka*, pages 52–60, 2015.

[3] Ivan Krasic and Stipe Celar. Telecom fraud detection with machine learning on imbalanced dataset. pages 1–6, 09 2022. doi: 10.23919/SoftCOM55329.2022.9911518.

[4] Partial Fulfillment and Telecommunication Engineering. International Revenue Sharing Fraud (IRSF) Detection Using Data Mining Techniques : The Case of ethio telecom. 2022.

[5] Judy Biljon and Paula Kotzé. Cultural factors in a mobile phone adoption and usage model. *Journal of Universal Computer Science*, 14, 01 2008.

[6] Joshua Evan Blumenstock, Dan Gillick, and Nathan Eagle. Who's calling? demographics of mobile phone use in rwanda. In *2010 AAAI Spring Symposium Series*, 2010.

[7] Joshua Blumenstock and Nathan Eagle. Mobile divides: gender, socioeconomic status, and mobile phone use in rwanda. In *Proceedings of the 4th ACM/IEEE international conference on information and communication technologies and development*, pages 1–10, 2010.

[8] Frehiwot Mola. Analysis and Detection Mechanisms of SIM Box Fraud in The Case of Ethio Telecom Thesis. *AAU repository*, pages 1–76, 2017. URL http://etd.aau.edu.et/bitstream/handle/123456789/18345/FrehiwotMola.pdf?sequence=1&isAllowed=y.

[9] YESHINEGUS GETANEH. Predictive modeling for fraud detection in telecommunications. *The Lancet Neurology*, 12(9):854. ISSN 14744422. doi: 10.1016/S1474-4422(13)70208-4.

[10] Clifton Phua, Vincent Lee, Kate Smith, and Ross Gayler. A Comprehensive Survey of Data Mining-based Fraud Detection Research. 2010. doi: 10.1016/j.chb.2012.01.002. URL http://arxiv.org/abs/1009.6119%0Ahttp://dx.doi.org/10.1016/j.chb.2012.01.002.

[11] Olusola Adeniyi Abidogun. Data mining, fraud detection and mobile telecommunications: call pattern analysis with unsupervised neural networks. 2005. URL http://etd.uwc.ac.za/xmlui/handle/11394/249.

[12] Constantinos S Hilas and Paris As Mastorocostas. Knowledge-Based Systems An application of supervised and unsupervised learning approaches to telecommunications fraud detection q. 21:721–726, 2008. doi: 10.1016/j.knosys.2008.03.026.

[13] Pablo A. Estévez, Claudio M. Held, and Claudio A. Perez. Subscription fraud prevention in telecommunications using fuzzy rules and neural networks. *Expert Systems with Applications*, 31(2):337–344, 2006. ISSN 09574174. doi: 10.1016/j.eswa.2005.09.028.

[14] Mohammad Iquebal Akhter and Mohammad Gulam Ahamad. Detecting Telecommunication Fraud using Neural Networks through Data Mining. 3(3):1–5, 2012.

[15] John Shawe-Taylor, Keith Howker, and Peter Burge. Detection of fraud in mobile telecommunications. *Information Security Technical Report*, 4(1):8–9, 1999. ISSN 13634127. doi: 10.1016/s1363-4127(99)80026-0.

[16] Bpm services for fraud detection and prevention — infosys bom. URL https://www.infosysbpm.com/industries/financial-services/service-offerings/fraud-detection-and-prevention.html.

[17] Moritz Von Mörner. ScienceDirect Application of Call Detail Records - Chances and Obstacles. *Transportation Research Procedia*, 25:2238–2246, 2017. ISSN 2352-1465. doi: 10.1016/j.trpro.2017.05.429. URL http://dx.doi.org/10.1016/j.trpro.2017.05.429.

[18] Gebremeskel Aregay. Subscription Fraud Prevention in Telecommunication using Deep Learning Approach: the case of ethio telecom. 2021.

[19] Donald J. Norris. Machine Learning: Deep Learning. In *Beginning Artificial Intelligence with the Raspberry Pi*, pages 211–247. 2017. doi: 10.1007/978-1-4842-2743-5_8. URL https://www.datacamp.com/tutorial/machine-deep-learning.

[20] Yashpal Singh and Alok Singh Chauhan. Neural networks in data mining. 2009.

[21] Hojjat Salehinejad, Sharan Sankar, Joseph Barfett, Errol Colak, and Shahrokh Valaee. Recent Advances in Recurrent Neural Networks. pages 1–21.

[22] TESFAY HADDISH. CONSTRUCTING PREDICTIVE MODEL FOR SUBSCRIPTION FRAUD DETECTION USING DATA MINING TECHNIQUES:THE CASE OF ETHIO-TELECOM. (June):2–4, 2012.

[23] Hailemeskel G/Tsadik. Constructing Subscription Fraud Detection Model Using Machine Learning Algorithms: the Case of Ethio Telecom. (January), 2021. URL http://etd.aau.edu.et/handle/12345678/25498.

[24] Getahun Surafel. Customer clustering for a mobile telecommunications company based on call detail records. 2021.

[25] Derebe Tekeste. A Comparative Analysis of Machine Learning Algorithms for Subscription fraud Detection : The case of ethio telecom. 2020.

[26] Fitsum Tesfaye. Near-real time sim-box fraud detection using machine learning in the case of ethio telecom. *Addis Ababa University*, 2020.

[27] Misrak Birhanu. *Near Real-time SIM-box Fraud Detection in Telecommunication System Using Machine Learning Approach in the Case of Ethio Telecom.* PhD thesis, St. Mary's University, 2024.

[28] Hagos Kahsu. *Sim-box fraud detection using data mining techniques: The case of ethio telecom*. PhD thesis, Ph. D. Dissertation. School of Electrical and Computer Engineering Addis . . . , 2018.

[29] Hugging Face. Hugging face – on a mission to solve nlp, one commit at a time., 2024. URL https://huggingface.co/.

[30] Kulvinder Panesar. Conversational artificial intelligence - demystifying statistical vs linguistic nlp solutions. *Journal of Computer-Assisted Linguistic Research*, 4:47, 05 2020. doi: 10.4995/jclr.2020.12932.

[31] Introduction to long short-term memory(lstm) — simplilearn. URL https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/lstm.

[32] Wikiwand - long short-term memory. URL https://www.wikiwand.com/en/Long$_{s}$hort-term$_{m}$emory.

[33] object Object. A new balance for efficiency and accuracy of feature selection for high-dimensional datasets. *core.ac.uk*. URL https://core.ac.uk/download/268070310.pdf.

[34] Rupali Goyal, Parteek Kumar, and Varinder Singh. A systematic survey on automated text generation tools and techniques: application, evaluation, and challenges. 04 2023. doi: 10.1007/s11042-023-15224-0.

[35] Kritika Verma and Pradeep Singh. An insight to soft computing based defect prediction techniques in software. *International Journal of Modern Education and Computer Science*, 7:52–58, 09 2015. doi: 10.5815/ijmecs.2015.09.07.

[36] Yong Fang, Yanhua Sun, Lu Zhang, Gengxin Chen, Mei Du, and Yunxia Guo. Stochastic simulation of typhoon in northwest pacific basin based on machine learning. *Computational Intelligence and Neuroscience*, 2022:1–16, 02 2022. doi: 10.1155/2022/6760944.

[37] Manjot Kaur and Aakash Mohta. A review of deep learning with recurrent neural network, 11 2019. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8987837.

[38] Robin M. Schmidt. Recurrent neural networks (rnns): A gentle introduction and overview, 2019.

Appendix A

BERT

from tensorflow . keras import Input

```
from transformers import AutoTokenizer, AutoModelForCausalLM
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, precision_score, recall_score
from sklearn.preprocessing import LabelEncoder

model_id = "NexaAIDev/Octopus-v2"
tokenizer = AutoTokenizer.from_pretrained(model_id)
model = GemmaForCausalLM.from_pretrained(
tokenizer = AutoTokenizer.from_pretrained("databricks/dbrx-instruct", trust_remote_code=True, to
model = AutoModelForCausalLM.from_pretrained("databricks/dbrx-instruct", device_map="auto", torc

input_text = "What ?"
messages = [{"role": "user", "content": input_text}]
input_ids = tokenizer.apply_chat_template(messages, return_dict=True, tokenize=True, add_generat

model_id = "NexaAIDev/Octopus-v2"
tokenizer = AutoTokenizer.from_pretrained(model_id)
model = GemmaForCausalLM.from_pretrained(outputs = model.generate(**input_ids, max_new_tokens=20
model_id = "NexaAIDev/Octopus-v2"
tokenizer = AutoTokenizer.from_pretrained(model_id)
model = GemmaForCausalLM.from_pretrained(

def inference(input_text):
    start_time = time.time()
    input_ids = tokenizer(input_text, return_tensors="pt").to(model.device)
    input_length = input_ids["input_ids"].shape[1]
    outputs = model.generate(
        input_ids=input_ids["input_ids"],
```

```
        max_length=1024,
        do_sample=False)
    generated_sequence = outputs[:, input_length:].tolist()
    res = tokenizer.decode(generated_sequence[0])
    end_time = time.time()
    return {"output": res, "latency": end_time - start_time}
```

Appendix B

MLP Algorithm script

```
# define the keras model
import numpy as np
from keras.layers import Dense, LSTM, Dropout, GRU, Bidirectional
import math
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split

from sklearn.metrics import mean_squared_error
from tensorflow.keras.optimizers import SGD
#X = np.asarray(X).astype(np.float32)
#y = np.asarray(y).astype(np.float32)
model = Sequential()
model.add(Dense(24, input_shape=(34,), activation='sigmoid'))
model.add(Dense(16, activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(8, activation='relu'))

model.add(Dense(1, activation='sigmoid'))

# compile the keras model
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
# fit the keras model on the dataset
model.fit(X, y, epochs=150, batch_size=20)
# evaluate the keras model
_, accuracy = model.evaluate(X, y)
print('Accuracy: %.5f' % (accuracy*100))
```

Appendix C

LSTM Algorithm script

```
from random import randint
from numpy import array
from numpy import argmax
import keras.backend as K
from tensorflow.keras import models
from numpy import array_equal
import numpy as np
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.layers import LSTM, Bidirectional
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras import Input
from tensorflow.keras.layers import TimeDistributed
from tensorflow.keras.layers import RepeatVector
X=data.drop('label',axis=1)
y=data['label']
X.shape
a = np.asarray(data)
a = a.reshape(1,12274, 63)
a.shape

timesteps=40          # dimensionality of the input sequence
features=3            # dimensionality of each input representation in the sequence
LSTMoutputDimension = 2 # dimensionality of the LSTM outputs (Hidden & Cell states)


input = Input(shape=(timesteps, features))
output= LSTM(LSTMoutputDimension)(input)
model_LSTM = Model(inputs=input, outputs=output)

model_LSTM.summary(s=['accuracy'])
```

```
# Train the model
model.fit(X_train, y_train, epochs=10, batch_size=32, validation_data=(X_test, y_test))

# Evaluate the model
loss, accuracy = model.evaluate(X_test, y_test)
print("Test Loss:", loss)
print("Test Accuracy:", accuracy)

# Make predictions on new data
predictions = model.predict(X_new)
```

Appendix D

rRNN Algorithm script

```
from keras.layers import Dense, LSTM, Dropout, GRU, Bidirectional
import math
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split

from tensorflow.keras.optimizers import SGD
#X = np.asarray(X).astype(np.float32)
#y = np.asarray(y).astype(np.float32)
model = Sequential()
model.add(Dense(24, input_shape=(34,), activation='sigmoid'))
model.add(Dense(16, activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(8, activation='relu'))

model.add(Dense(1, activation='sigmoid'))

# compile the keras model
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
# fit the keras model on the dataset
model.fit(X, y, epochs=150, batch_size=20)
# evaluate the keras model
_, accuracy = model.evaluate(X, y)
print('Accuracy: %.5f' % (accuracy*100))
```

Assurance

This thesis is submitted as a partial fulfillment of the Master of Science in Cyber Security (School of Information Technology and Engineering), Addis Ababa Institute of Technology, Addis Ababa University. The author carried out the work included in this thesis. Wherever other people's contributions are involved, every effort is made to make this clear, with an appropriate citation from the source.

Fikirte Endalew

Signature

.....

Date: June, 2024