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

Detection of SIM-BOX Fraud Using Deep Learning

By:

Haile Welay

Advisor:

Tsegamlak Terefe (Ph.D.)

A Thesis Submitted to
School of Electrical and Computer Engineering

In Partial Fulfillment of the Requirements for the Degree of Masters of Science
in
Telecommunications Engineering

June 26, 2025

Addis Ababa, Ethiopia

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

Detection of SIM-BOX Fraud Using Deep Learning

by:

Haile Welay

Signed by the Examining Committee

Internal Examiner _____ Signature: _____ Date: _____

External Examiner _____ Signature: _____ Date: _____

Advisor Tsegamlak Terefe (Ph.D.) Signature: _____ Date: _____

Co-Advisor _____ Signature: _____ Date: _____

Dean, School of Electrical and Computer Engineering

Declaration

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

Haile Welay Gebremedhin _____

Name Signature

Abstract

In underdeveloped countries, the telecommunications infrastructure is often subsidized by the high cost of incoming international calls. However, this situation has led to an increase in sim box fraud, where attackers use VoIP-GSM gateways, known as "SIM-BOXES," to illegally route international calls through local wired data connections. The research presented here developed models for the classification of Call Detail Records (CDRs) in order to come up with a model that identifies fraudulent subscribers with higher accuracy. Three classification techniques, viz. Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), and autoencoder, combined with three user aggregation datasets (4-hour, daily, and monthly aggregated), were used. These three algorithms, along with the three datasets, were applied in building the models. Results of the work show that LSTM performed better among the three algorithms with an accuracy of 99.81% and a lesser false positive on the monthly aggregated dataset.

Keywords— Deep learning, Bypass Fraud, Fraud Detection, SIM-Boxing

Acknowledgments

I want to start by giving thanks to God for providing me the strength to complete every step. Next, I want to express my sincere gratitude to my adviser, Tsegamlak Terefe (Ph.D.), for his helpful, insightful feedback and encouragement. Additionally, I would like to express my gratitude to my evaluators, Surafel Lemma (Ph.D.), Fitsum Assamnew (Ph.D.), and Sosina Mengistu (Ph.D.), for their comments during the thesis progress presentations.

I want to sincerely thank my esteemed family for their support and for helping me with my thesis work. I also want to express my thanks to my uncle Tsegay G/medhin, my mother Ngisti G/medhin, my wife Nigisti Hagos, my dearest friend Samrawi Hailemariam, and my elder brother Teklay G/mariam for helping me finish this study.

Contents

List of Tables

List of Figures

Acronyms

ANN	Artificial Neural Network
API	Application Programming Interface
AUC	Area Under the Curve
BTS	Base Transceiver Station
CDR	Call Detail Record
CDRs	Call Detail Records
CFCA	Communications Fraud Control Association
CNN	Convolutional Neural Network
CNNs	Convolutional Neural Networks
DB	Database
DBNs	Deep Belief Networks
DL	Deep Learning
DM	Data Mining
FMS	Fraud Management System
GDP	Gross Domestic Product
GPUs	Graphics Processing Units
KNN	K-nearest Neighbor
ML	Machine Learning
NN	Neural Network
RBM	Restricted Boltzmann Machine
RF	Random Forest
ReLU	Rectified Linear Unit
ROC	Receiver Operating Characteristic
SIM-Box	Subscriber Identity Module Box
SVM	Support Vector Machine
TPUs	Tensor Processing Units
TCG	Test Call Generation
VoIP-GSM	Voice over Internet Protocol- Global System for Mobile Communication

1 Introduction

Fraud is one of the most challenging problems in the telecom industry. Due to fraud's dynamic nature and people's exploitation of new technology, the telecom industry's operation is becoming more and more concerning. Due to this, the telecom sector is making every effort to safeguard its services against fraud and reduce income loss.

The Communications Fraud Control Association (CFCA) stated in their 2019 report that telecom operators have lost a mind-boggling amount of $28.69 billion globally through fraud. By 2021, the numbers got worse and reached $39.89 billion [7, 8]. Most of the fraud in the telecommunications sector is coming from the SIM-box fraud operations, which are the biggest culprits. Moreover, via the usage of SIM cards, the fraudsters are not only able to go around the international call rates but they also convert these calls into local ones which is the main reason for the telecom operators' significant revenue losses. Customers can use their mobile service numbers to commit fraud, such as subscriber identity module box (SIM-Box) fraud and roaming fraud, which have a very high impact on the company's revenue loss.

Currently, in Ethio Telecom, customers are forced to subscribe to a limited number of mobile service numbers. However, there is the possibility that a customer can subscribe using counterfeit identities to obtain service numbers that exceed the allowed limit, thereby engaging in fraudulent activities. Detecting SIM-Box fraud is crucial in preventing various fraudulent activities that can result in revenue loss for a telecom company.In the case of Ethio Telecom, the majority of revenue loss can be attributed to SIM-Box fraud. The revenue loss in the Ethiopian telecom sector is increasing, as reported in Table 1 by [1].

The Ethiopian Federal Police have arrested 32 people who are suspected of committing a telecom fraud case that involved the state-owned company Ethio Telecom. The arrest of the suspects was reported by Police TV on the national broadcaster in a news bulletin aired in 2019. The report revealed that the arrested individuals had engaged in an illegal telecom business and employed various gadgets and modems to carry out their activities. The country is said to have

been defrauded of more than 30 million birr (about $1.1 million) because of their fraudulent dealings, the report asserts [1].

Table 1: Revenue loss report due to fraud in Ethio telecom [1]

Year	**Ethio telecom's total revenue loss ($ Million)**
2015/16	35.5
2016/17	52
2017/18	89

The bypass fraud, which ranks as the most expensive fraud worldwide and costs businesses $6 billion a year, has a significant influence on the telecom industry [9].

This research attempts to apply deep learning techniques to address the problem of SIM-Box fraud in light of substantial revenue losses.The subsequent sections will elaborate on the proposed methodology, research objectives, literature review, and the contribution of this study to fraud detection in the telecom industry.

1.1 Problem Statement

Telecommunication fraud poses a significant challenge to the capabilities of telecom service providers. The most damaging type of telecommunication fraud is definitely the interconnection bypass fraud, with SIM box fraud being the most illustrative example. The result of interconnect bypass fraud is so big that it is beyond the imagination of these operators.

"Sim box" is a term used for the illegal act of tricking telecom operators' systems into thinking calls are intended for international destinations. It is achieved by employing SIM box devices, which convert international calls into local calls; hence, they do not pay the higher rates for international calls that are set by the telecom companies. EthioTelecom, being the largest telecommunications provider in Ethiopia, is thus the major party at risk of SIM box fraud and hence losing their money substantially without even knowing. Rule-based systems are an example

of traditional methods that are used for detecting SIM box fraud, but they can hardly find the sophisticated fraud patterns most of the time. That being said, a new detection system that is based on deep learning algorithms should be made to confirm the correct situation of call fraud continuously.

Research that is presently available has utilized machine learning and data mining on call detail records (CDRs) for detection of fraud but such studies are mostly limited to prepaid users and do not consider identifiers such as IMEI and IMSI [2,3,10]. This not only limits the scope of the research but also may lead to the loss of some valuable insights. Our approach which is based on postpaid users and related identifiers as well as a deep learning model allows us to develop a more effective and all-encompassing system for fraud detection and revenue protection.

1.2 Objective

1.2.1 General Objective

In general, the aim of this study is to build a model that detects fraudulent activities due to SIMBox fraud using a deep learning approach.

1.2.2 Specific Objective

The specific objectives of the research work are as follows:

- To suggest a SIM-Box fraud detection method that can be used in the real world and improve performance and accuracy.
- To build a model and detect SIM-Box fraud using a deep learning approach
- To evaluate the performance of the model.

1.3 Literature Review

Telecom fraud has been the subject of numerous studies. Different research uses data mining and machine learning (ML) techniques to address the dynamic worldwide issue of telecom fraud.

Researchers in [2] focus on the performance assessment of fraud detection in SIM-Box using machine learning techniques and have followed different methodologies, such as gathering normal and SIM-Box fraud CDR, using support vector machines (SVM) and neural networks (NN) to perform the classification. Consequently, various models of support vector machines and neural networks were constructed. The simulation results show that artificial neural networks (ANN) using the Bayesian regularization technique (99.87%) provided the best precision when compared to SVM (99.24%).

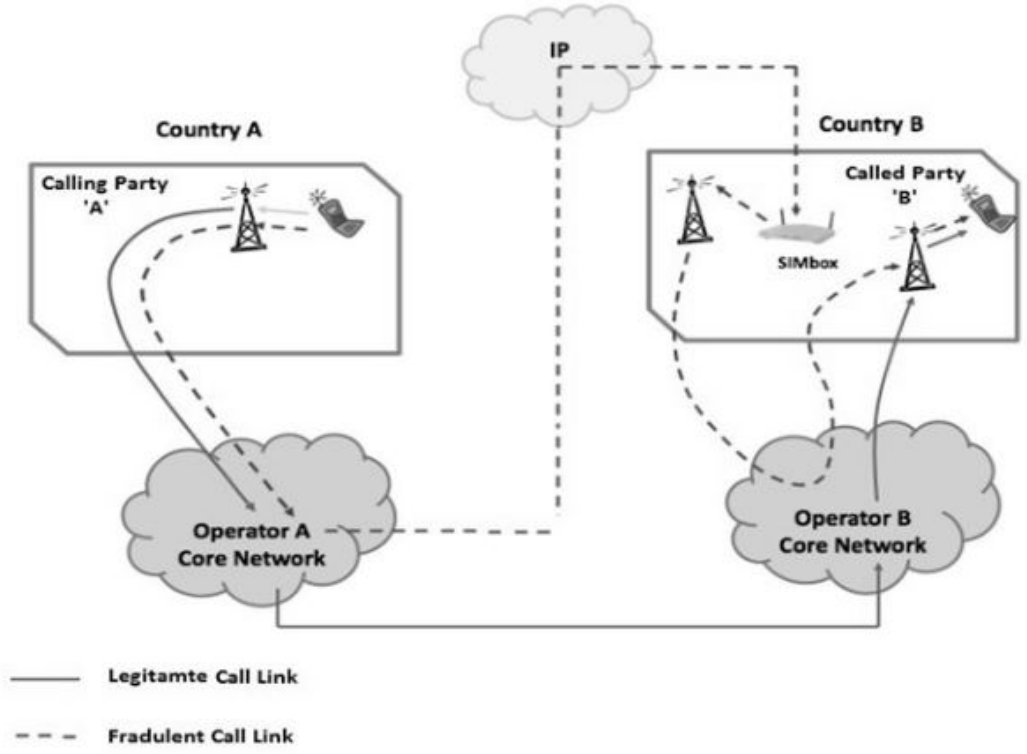

Figure 1: Legitimate and Fraudulent Call Setup [2]

The study in [3] examines data mining methods for SIM-box fraud detection in ethiotelecom. The sample of the data set they used was 20,000 subscribers. Out of the 20,000 subscribers whose CDR data was collected from ethiotelecom, five thousand were identified as fraudulent by the security department, which subsequently took action to stop them. The data mining technique for SIM-box fraud detection was the main focus of the study. The chosen algorithms used in the study include ANN, SVM, and RF. The models for each algorithm were trained and evaluated

at many levels of granularity, such as four hours, a day, and a month. The classification performances of each method varied. Lastly, when considering daily and monthly granularity levels, the RF algorithm model with a 4-hour granularity level outperformed the other two algorithms, SVM and ANN models, in terms of accuracy.

The other research was conducted on Almadar Aljadid operator [4]. The analysis was based on four features: the user numbers making calls but not receiving any, the user numbers receiving texts but not sending any, the user numbers making calls from a fixed location, and the user numbers whose calls lasted longer than the allowed time. In conclusion, it was discovered that Neo4j has the capacity to successfully analyze enormous data files (CDR has 50 million entries), and in this article it was utilized to find SIM boxes associated with SIM cards.

Research was done in Libya on the extent and consequences of bypass fraud, which is the second most costly fraud in the world and costs telecom companies over $6 billion a year.. Since Libya's major mobile operators are state-owned, the revenue they generate might contribute to the country's economic development, and anything that would have an adverse effect on it would lower Libya's GDP. The research was conducted to detect fraudulent SIM boxes using CDR analysis combined with machine learning algorithms. The authors used different algorithms, such as SVM and decision trees (the random forest algorithm), as supervised learning algorithms to detect sim-box fraud, and since the labeled data they used was scarce, they instead used unsupervised learning algorithms to cluster the SIMs in order to get a better algorithm that could improve the designed algorithm. Then they tested and evaluated the performance of the designed algorithm in terms of accuracy and precision. Thus, of the compared algorithms, they found unsupervised learning algorithms to be better than the others in terms of performance and accuracy in detecting SIM-box fraud. Lastly, they said that they had been concerned that the information might be exploited by the scammers, which would limit the amount of information that could be revealed [9].

The author in [11] focuses on the issue of SIM box fraud in the telecom sector, with a particular focus on Ethiopian Telecom. She used Ethio Telecom's call detail records (CDRs) for data analysis, and she used data mining techniques to create models that could identify both legitimate

and fraudulent number usage. Four classification algorithms—decision trees, rule-based induction, neural networks, and hybrid algorithms—were employed. The results demonstrated that the hybrid (J48 and PART) and PART rule-based algorithms outperformed the others, exhibiting high accuracy rates in differentiating between phony and authentic calls.

Another study [10] highlights a case of SIM-box fraud detection, a type of telecom fraud committed by the bypassing of the interconnect. The authors decide to implement machine learning algorithms, namely Random Forest, Artificial Neural Network, and Support Vector Machine, for fraud detection in near real time. In this study, the researchers utilize the sliding window aggregation approach to decrease the detection lag, and they reach a classification accuracy of 96.2%. Experimental results illustrate that Random Forest outperforms in terms of accuracy and detection delay. In this regard, the paper substantially supports the fight against the fraudulent activities in the telecommunications sector with the aid of the machine learning technologies. However, the paper would be more well-rounded if the authors provided a more detailed analysis of the methodology and the experimental setup along with a discussion about the limitations and possible future areas of research

The authors in [12] built a model to classify SIM box fraud subscribers using ANN and SVM. In this study, they examine the results of comparing 240 neural network models produced as a result of experimenting with every possible combination of parameter settings. The models were assessed according to their precision, recall, generalization error, time spent developing the model, and prediction accuracy. Then the ANN outperformed, and they got 98.7% accuracy.

As we have been investigating different literature, many studies have studied different machine learning techniques to detect SIM-Box.

1.4 Methodology

The primary goal of this study is to develop a model for detecting SIM-box fraud using deep learning techniques. To achieve the general and the specific objectives, the following methodology

was followed:

- Problem Understanding: The problem domain was first understood by reviewing relevant academic literature on telecom fraud, with a particular focus on SIM-box fraud detection.
- Data Collection and Preparation: One month of Call Detail Record (CDR) data was collected. The dataset was then explored to understand its structure and content. Based on the understanding and expert input, relevant features were selected for use in the study.
- Data Preprocessing: he selected features were cleaned and transformed through preprocessing steps to make them suitable for training deep learning models. This included handling missing values, encoding categorical variables, and scaling numerical data as necessary.
- Labeling the Dataset: The dataset was labeled by classifying each record as either fraudulent or legitimate
- Model Training and Evaluation Using Python: The Python tool was used to implement and train the selected deep learning algorithms on the prepared dataset. The performance of the classification models was evaluated using a variety of standard evaluation metrics, including
  - Overall Accuracy
  - Receiver Operating Characteristic (ROC) Curve
- Model Testing and Performance Evalution: The trained models were tested, and their performance was assessed using the above evaluation measures to determine how effectively they could detect SIM-box fraud.

Finally, the results were analyzed and discussed in depth. The study concludes with a summary of key findings and offers recommendations for future research.

1.5 Scope

Telecommunication industry is known to be associated with many types of fraud and has a lot of data dealing with such issues. This research, however, has been targeted particularly at identifying SIM-Box fraud only.

1.6 Contribution of the research

This research contribution is quite significant in that it includes important characters like IMEI (device identity) and IMSI (subscriber identity), which are usually not given much attention in the other studies.

1.7 Thesis Organization

This thesis work's following parts are arranged as follows. Chapter 2 provides an overview of telecommunications fraud and addresses SIM-box fraud. In Chapter 3, the basics of deep learning algorithms are discussed.The experimental analysis of this research is covered in Chapter 4. Activities carried out throughout the system model construction process include feature selection and data preprocessing, aggregation methods, algorithm training, and evaluations. The results of the algorithm's model's performance evaluation are discussed in Chapter 5. Chapter 6, the last chapter, contains the conclusion and recommendation.

2 Overview of Telecommunications Fraud

In the telecommunications sector, one of the major risks to revenue generation and service quality is telecommunications fraud. This fraud is defined as either the theft of telecom services or the use of telecom services to commit various forms of fraud [13].In addition, it can be defined as the misuse of telecom products or services with the goal of obtaining money unlawfully from telecom service providers or recipients. As the technology for telecommunications networks continues to advance, criminals' methods of committing telecoms network fraud are getting more realistic and covert. The dynamic nature of telecom fraud means that if fraud actors (fraudsters) believe they will be caught or stopped, they will strive to find a way to get around security measures. Security measures must continuously evolve to stay ahead of these tactics, incorporating advanced technologies such as artificial intelligence and machine learning to detect unusual patterns and behaviors. By doing so, telecom providers can better protect their networks and minimize the financial impact of such fraudulent activities [14]. Because fraudsters are smart enough to continuously look for exploitable weaknesses in the telecom services and networks. Part of their motivation is accounted for by the fact that once an exploit is defined, plenty of potential targets could be available.

The significant loss of money resulting from telecommunications fraud can have an impact on the business and reputation of telecommunications companies [14]. Fraudsters must either lower or avoid paying for the services entirely. Multibillions of US dollars are thought to be lost annually as a result of telecommunications fraud worldwide [15]

2.1 Types of Telecommunication Fraud

According to the study in [13], there are four types of telecommunication fraud: procedural, technical, hacking, and contractual fraud. Premium Rate Service (PRS) fraud and subscription fraud are examples of contractual fraud, which generates income through routine service use without any intention of paying for the usage. When someone commits hacking fraud, they make money by breaking into unprotected systems and taking advantage of whatever capability

is available, such as Private Automatic Branch eXchange (PABX) fraud.Technical fraud refers to any fraud that targets flaws in the mobile system's technology. Initial technical knowledge is usually required for such scams, but once a vulnerability has been discovered, the information is frequently disseminated in a way that non-technical individuals may use. This group includes, for example, technical internal frauds and cloning. However, all frauds in procedural fraud target the flaws in the business processes that allow access to the system and attack the procedures that reduce fraud exposure. Roaming fraud, duplicate voucher IDs, and defective vouchers are a few examples of this category.

Fraud in telecommunications can take several forms and manifest at different levels. Frauds have been classified differently by various authors. For example [16] classified the most prevalent fraud types as subscription fraud and superimposed fraud, while [3] divides them into seven categories: technical, internal fraud, social engineering, fraud based on technological flaws, fraud based on new technology, and superimposed fraud. Likewise, [9] identifies international revenue share fraud, premium rate service fraud, and bypass fraud as the top three forms of telecommunications fraud that result in a notable loss. [3] categorized as technical, procedural, hacking, and contractual frauds.

With the advancement of technology and telecommunications, as well as the magnitude of the telecom sector, which makes it particularly appealing to scammers [17], more sophisticated forms of fraud have supplanted the more conventional ones, which have spread too quickly throughout the world. Fraud types can also be separated into two categories: fraud in new technology and fraud in conventional networks. The first kind of fraud can take numerous forms, such as subscription fraud, which is when someone signs up for a service using a stolen or fictitious identity without committing to paying the bills. And additional kinds, including calling card fraud, roaming fraud, dealer fraud, internal fraud, premium rate service fraud, and SIM cloning.

Numerous types of fraud pose a threat to the telecommunications sector, which is the most prevalent fraud field, according to research [18]. It is estimated that there are about 200 different kinds of telecom fraud. Of these, SIM-BOX fraud and overlaid fraud are the most prevalent types

of telecommunication fraud. Some of the most prevalent types and strategies of fraud in the telecom industry are described here.

2.1.1 Superimposed Fraud

Superimposed fraud occurs when scammers take control of a valid account and utilize services without the required authorization; this is represented in the bill as phantom calls [18]. In these situations, the usual usage of the authorized consumers is superimposed atop the aberrant consumption. Superimposed fraud can be committed in a variety of ways, such as by getting calling card authorization information and cloning a mobile phone.

2.1.2 International Revenue Share Fraud

According to [19], the largest contribution to total fraud losses is International Revenue Share Fraud (IRSF). It takes place when an operator enters into a contract with a third party that will generate calls to premium rate numbers in order to increase traffic. IRSF frequently combines several different fraud techniques. Among the methods are social engineering, phone forwarding and diverting, and taking advantage of roaming SIM cards or dialer malware. Fraudsters profit from the sharing agreements and create high-traffic calls to expensive locations.

2.1.3 Interconnect Bypass Fraud

Unauthorized access to another carrier's network is known as interconnect bypass fraud. This kind of fraud is also known as SIM boxing, Global System for Mobile Communications (GSM) gateway fraud, or interconnect fraud. In this case, the fraudsters must have access to cutting-edge technology like VoIP, which can effectively circumvent the standard international call payment system by making foreign calls seem less expensive than domestic ones. Usually, the scammers would offer long-distance calling cards for sale. Operators have the ability to switch the call so that it seems to be a domestic call when clients dial the number on the cards [9].

2.1.4 Subscription Fraud

Subscription fraud is one of the most common telecom frauds in the industry. The fraudulent use of telecom services or the use of such services for further fraudulent operations is made possible

by the use of false identities and/or identity theft at the point of sale. Fraudsters can access telecom services, such as phone, data, and Short Message Services (SMS), as well as mobile financial services like mobile banking and mobile payments, after creating a phony subscription [20]. The motivation could be as simple as trying to take advantage of a known weakness or being opportunistic. However, organized criminals now control this, creating numerous false identities over an extended period of time. Additionally, fraudsters have developed a thorough understanding of fraud systems and are constantly testing the thresholds to take advantage of the weaknesses in the systems.

2.1.5 Subscriber Identity Module Box Fraud

In ethiotelecom, one of the most prevalent forms of fraud that impacts network revenue and quality of service is Subscriber Identity Module Box (SIMBox) fraud. With the advent of Voice over Internet Protocol (VoIP) technologies, SIMBox fraud—a sort of bypass fraud—has become more prevalent in international calls [3]. Global voice calls are intercepted by SIMBox scammers, who then use the Internet to send them to a mobile device, where they are converted to local calls at the target network. SIMBox devices use a variety of SIM cards from both domestic and international carriers. Until the operator finds and shuts them down, fraudsters can use a fresh set of SIM card numbers whenever they identify and deactivate fake service numbers [21].

2.1.6 SIM-Box Fraud Scenario

Every time a subscriber calls a foreign location, the call travels via several different companies [3]. The logical first step in explaining SIM-box fraud is to outline the legal international call route, followed by a discussion of the fraud bypass scenario. Pretend that subscribers A and B occupy separate countries, A and B, respectively.

- Through the mobile operator, subscriber A calls subscriber B and receives payment from the service provider.
- Subscriber A generated the call and routed it to the international gateway in nation A.
- The received call is forwarded to a temporary operator by country A's domestic international gateway, which also covers the cost.

- Once this call has been routed to a destination international gateway (country B), the temporary operator pays a toll to the destination international operator.
- Finlay, subscriber B in country B receivers a international call from abroad but with a local number, it may Amazed.

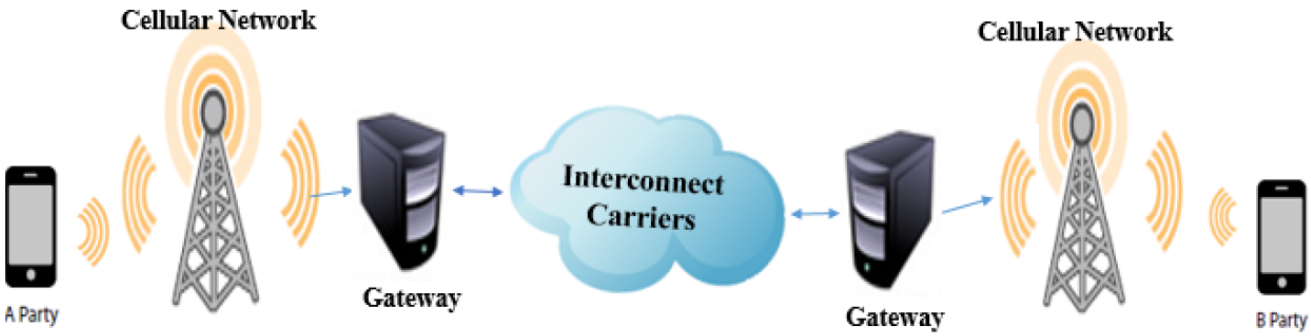

Figure 2: Legitimate Route of International Call, adopt from [3, 4]

- Using the domestic mobile operator network, subscriber A calls subscriber B and pays for the call.
- In country A, the call that subscriber A generated was routed to the home international gateway.
- The received call is routed to a temporary operator and paid for by country A's domestic international gateway.
- After that, the temporary operator uses VoIP to route this call to a SIM box located in nation B and pays a toll to the SIM boxer.
- Using its local SIM card, the SIM-Box then makes a separate call to subscriber B on country B's network; this makes it appear to be a local call, and it saves money by only charging for the local call rather than interconnecting.
- Finlay, subscriber B in nation B may be amazed when it receives an international call from overseas with a local number.

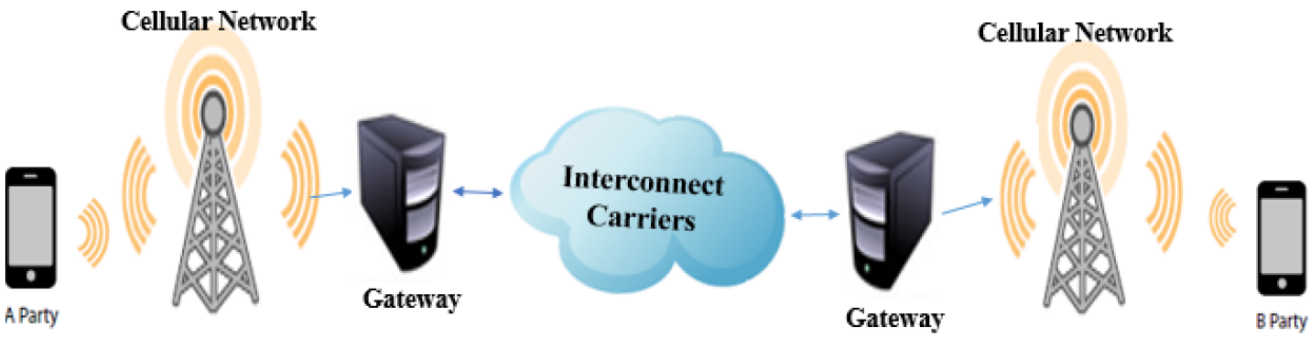

Figure 3: SIM-Box Fraud Rout of International Call, adopt from [3, 4]

3 Deep Learning Algorithm

It is challenging for systems that depend on hard-coded knowledge to solve dynamic problems. It is suggested that an artificial intelligence (AI) system learn on the fly by seeing patterns in raw data in order to get around this problem.Artificial intelligence (AI) is the replication of human intellect in robots that are designed to mimic human thought processes and behaviors [22]. ML is the process of implementing AI by extracting features from raw data. Computers can already solve problems using real-world knowledge and draw seemingly subjective conclusions thanks to machine learning (ML). Machine learning (ML) is the study of algorithms that automatically get better with experience. This is achieved through an iterative process that uses learning to improve the algorithms' conclusions. It is a wise and practical move to create AI systems that can operate in challenging real-world environments.

One subset of machine learning called deep learning has the ability to learn to represent the world as a nested hierarchy of concepts with greater flexibility. One effective method for gleaning significant features from the available raw data is deep learning. Most significantly, it carries out a hierarchical feature extraction and is independent of manually created features such as local binary patterns, a gradient histogram, etc. Layer-wise feature learning means that it learns low-level features in the first levels and then begins to learn a more abstract representation of the input as it advances up the hierarchy.However, machine learning (ML) is not an effective way to extract significant characteristics from the raw data, in contrast to deep learning. For optimal

performance, it uses hand-crafted features as an input [23].

Deep learning offers practical tools for processing large volumes of data and producing insightful predictions in scientific domains; it has greatly benefited other sciences. This type of machine learning, which is based on the idea of artificial neural networks (ANNs), learns representations from data with a focus on learning successive layers of increasingly meaningful representations. In essence, a neural network consists of three layers: an input layer, many hidden layers, and an output layer.The depth of the model is the number of layers that contribute to the data. Deep learning algorithms come in a variety of forms, including CNN, autoencoder, recurrent neural network, and long short-term memory network [23]. To complete particular tasks, each of these algorithms makes use of a different kind of neural network. Deep learning algorithms, which can operate with nearly any type of data, require a lot of information and processing capacity to solve complex problems.

3.1 Convolutional Neural Networks(CNN)

One of the most popular kinds of deep neural networks used in computer vision is the convolutional neural network (CNN), also known as ConvNet. It uses a mathematical operation called convolution in place of general matrix multiplication in at least one of its layers to analyze visual images by processing data with a grid-like topology [15, 20].Convolutional networks, which are just neural networks, have shown remarkable success in real-world applications. CNNs have multiple layers and use images as input, unlike simple neural networks. Neuroscientists can examine a model of visual processing provided by contemporary convolutional networks for object recognition. Numerous statistical characteristics of natural images are translation-invariant. Major classification issues where features are automatically learned from low level to high level on ever-increasing layers of the network are the primary applications for CNN [24, 25]. The CNN method operates in three basic stages: numerous pooling operations, non-linear operations, and multiple convolutions [24]. To create a collection of linear activations, several convolutions are carried out employing filters concurrently in the first stage. The second step uses a nonlinear activation function (such as a rectified linear one) that returns the input directly if it is positive and zero otherwise. However, in the third stage, the feature map's size is decreased using a

pooling technique. There are several pooling functions, including weighted average pooling, max pooling, and average pooling.In contrast to average pooling, which takes the average of all the values in a rectangular window, max pooling locates and takes the maximum value within the panel. CNN's final pooling layer's output is fed into the fully connected (FC) layer, which ensures that every node there is connected to every other node in the layer above. Classifying the input data into distinct classes is done using the FC layer.

CNNs, on the other hand, are trained using a method known as backpropagation, which has four steps: weight update, loss function, and forward and backward pass [24]. Random initialization is used for the filter weights. The training images are sent to the network during the forward pass. The loss function is used to calculate the error rate, which is then used to determine the backpass and weight update phases based on the comparison between the network output and the desired output. Up to convergence, the backpropagation operation can be carried out repeatedly.

CNN's architecture makes it unique since it requires little pre-processing of the input image and can handle segmentation, feature extraction, and classification all in one processing module. Efficiency in pattern recognition tasks is thought to be achieved with minimal domain knowledge of the problem. CNN is used in many different applications, including face detection, face recognition, object recognition, gender recognition, character recognition, and texture recognition [20].CNNs use a variety of connections and layers, including convolution, pooling, and fully connected layers, and they implement regularization in some way [25]. The purpose of regularization is to keep the model training from overfitting. It can be enhanced by modifying a weight decay coefficient or by implementing a regularization technique like data augmentation or dropout. The general CNN architecture is displayed in Figure 4.

3.1.1 Convolution Layer

The primary component of a CNN is the convolution layer, which consists of a number of filters or learnable kernels designed to extract unique features from the input. A feature map is computed by each kernel in this layer. Corners, edges, lines, and textures are examples of low-level relevant features that are extracted by the first convolutional layer, whereas higher-level features are

extracted by the second convolutional layer. The last layer extracts the input data's highest-level features.In order to identify patterns in the image and produce a particular feature map of the filter size, the convolution layer employs an independent matrix filter that carries out the convolution process. To obtain a rectified feature map of the input picture, the convolution layer is subjected to an activation function known as the Rectified Linear Unit (ReLU) [20].

$$S(t) = (x * w)(t) \tag{1}$$

where x is the input, w is the filter kernel, and S is the output called feature map or kernel map

3.1.2 Pooling Layer

Each convolution block is followed by a pooling layer, which lowers the resolution of the prior feature maps by compressing data and increasing the network's computational complexity. To guarantee that the network concentrates on the most significant patterns, it adapts the features to be resilient to disorder, noise, and other minor fluctuations. Generally speaking, a pooling layer lowers the dimensionality of the feature maps used in the following layers and creates down-sampled representations of the input map [25]. The most popular method for the pooling layer is max pooling, which sub-samples the input image to make the representation roughly invariant to slight input translation.

3.1.3 Fully Connected Layer

The final step of the CNN topology, which consists of a generic multi-layer network, is the fully linked layer. FC one-dimensional levels to all activations in the preceding layer make up the final few higher layers. We must use an FC neural network to categorize the data into different classes once the convolution and pooling layers have completed feature extraction. The FC layer, which makes up the final few layers of the network, is merely a feedforward neural network. The final pooling layer's flattened output serves as its input [20].

3.2 Long Short-Term Memory (LSTM)

Long-term dependencies can be monitored by recurrent neural networks, or LSTMs. They are therefore excellent for developing models that rely on context and previous states and for learning

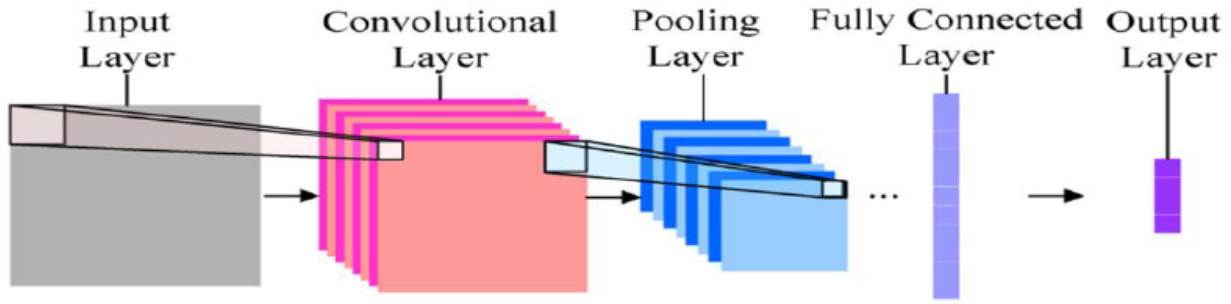

Figure 4: General CNN Architecture

from sequence input data. Previous states' relevant information is retained in the LSTM cell block. The input, forget, and output gates determine what enters the cell, what stays in the cell, and the cell values that are utilized to compute the LSTM block's output for automatic feature extractions, respectively.LSTM techniques have also been used in a variety of applications for unsupervised learning. The primary drawbacks of unsupervised learning are its processing complexity and inability to yield precise information about data sorting. Among the most widely used methods for unsupervised learning is clustering [26].

A recurrent neural network (RNN), also called long short-term memory (LSTM), was developed to address the vanishing gradient issue [5]. Over time, this neural network can identify patterns in the sequence of inputs that come into the units. It is therefore a suitable option for time series forecasting. LSTM units come in a wide variety of structures. A memory cell, an input gate, an output gate, and a forget gate are all found in a typical LSTM unit. In Fig 5, an LSTM unit is displayed. Every gate is managed by sigmoid-equipped blocks.

Every gate is managed by blocks that use sigmoid as an activation function. The output from 0 to 1 is used by these blocks to open or close the gates. The input gate regulates how strongly a new value enters the cell. The forget gate determines whether or not the prior value ought to stay in the cell. The output gate, which comes last, modifies how current values affect the output of the cells. The unit's activation function is tanh. The LSTM neural network is trained in this study using backpropagation across time.

A feedforward neural network can be used to create an autoencoder. But in order to account for temporal data, we will construct an autoencoder using LSTM layers. In contrast to feedforward

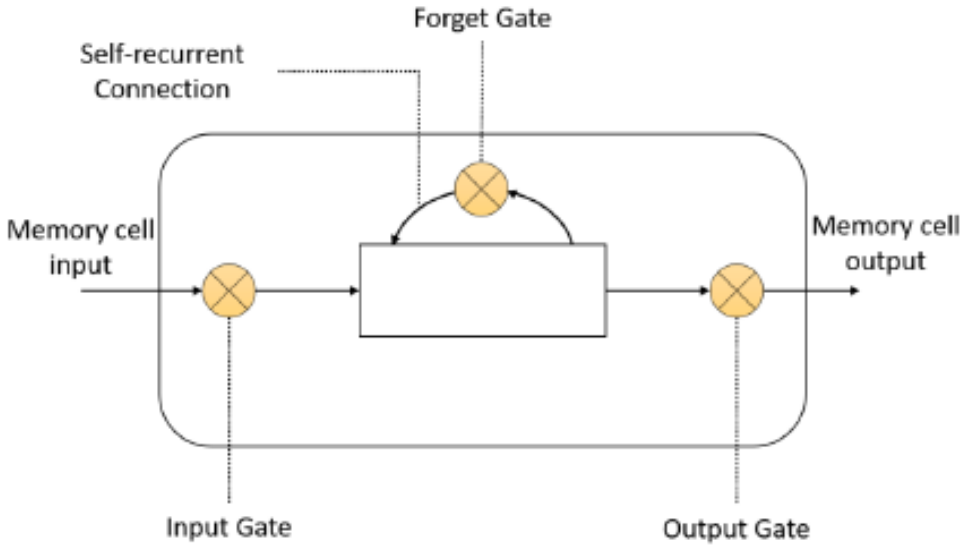

Figure 5: Diagram of an LSTM unit and its inputs and outputs [5]

neural networks, we feed data into LSTM one number at a time in a consequential manner. The neural network stores some knowledge about past time series values since each LSTM unit is a generalization of an RNN unit [6].

LSTM units have three gates.

- the input gate, which is in charge of obtaining new information;
- the output gate, which determines how much information we output
- the forget gate, which maintains the fraction of information from earlier states.

Neural networks process the input sequence one number at a time. One neuron makes up the dense layer in this situation. In order for the output neuron to receive the information from the previous input, the neural network continues to operate during N+K time steps.

3.3 Autoencoder

Using an unsupervised technique, an autoencoder is a neural network that learns the representation in the input data set to reduce dimensionality and reproduce the original data set. The

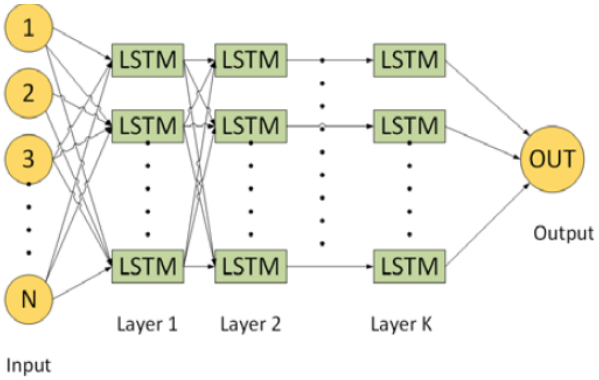

Figure 6: Neural network architecture based on LSTM layers [6]

application of backpropagation serves as the foundation for the learning algorithm [27]

The concept of principal component analysis (PCA) is expanded upon by autoencoders. PCA creates a linear representation from multi-dimensional data. Conversely, autoencoders are able to generate nonlinear representations. A set of linear variables is identified via PCA in the directions with the highest variance. A smaller (i.e., less than m) dimensional space is created by representing the p-dimensional input data points as m orthogonal directions, so that $m \leq p$ information in the appropriate orthogonal directions is omitted when the original data points are projected onto the principal directions. PCA searches for the linear function with the highest variance and places greater emphasis on variances than covariances and correlations [27]. A simplified illustration of how autoencoders might learn to replicate input data in the output layer by reducing its dimension is shown in Figure 7.

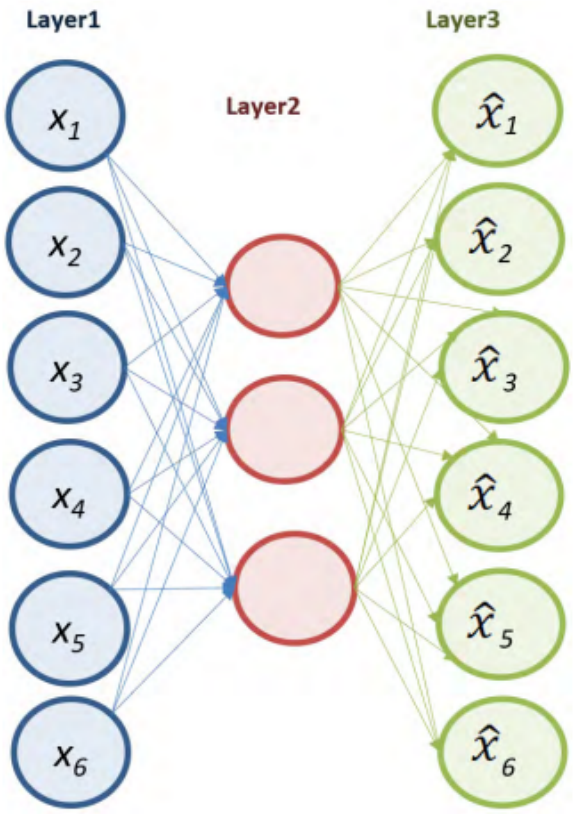

Figure 7: Autoencoder nodes

4 Experimental Analysis

This chapter covers the entire experimental procedure used in this study. Figure 8 illustrates the model's experimental procedure; the primary activities carried out to identify SIM-box fraud are data collecting, data pre-processing, and classification. The upcoming sections contain a description of the tasks completed under these modules.

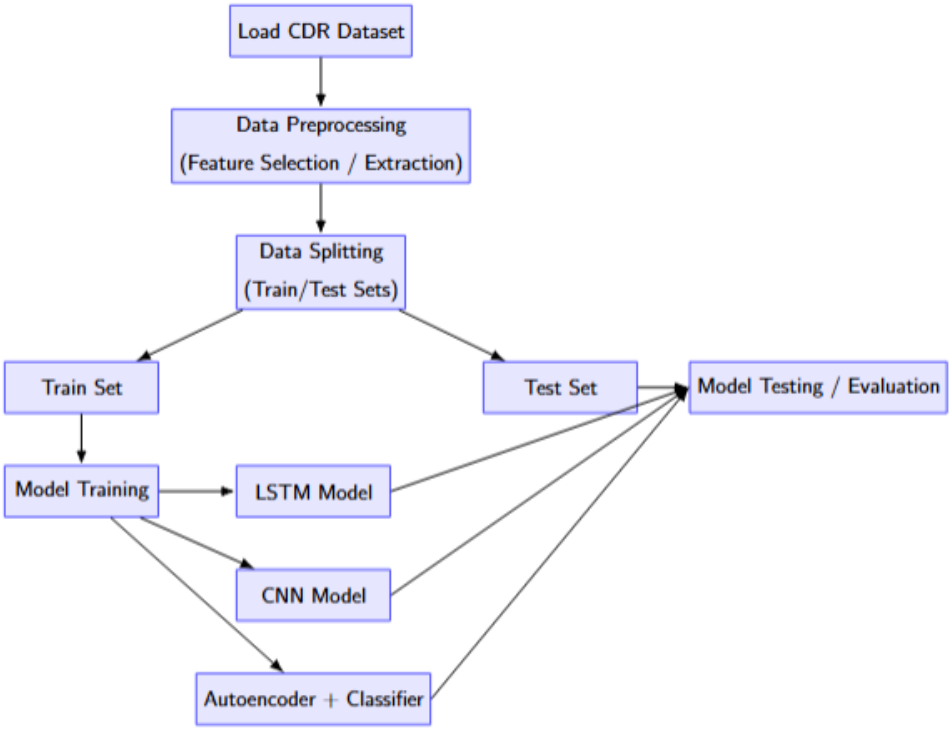

Figure 8: System Mode l [6]

4.1 Data Collection

ethiotelecom Currently, FMS analyzes and detects telecom frauds using customer data records (CDRs); the experiment in this study also leverages the same CDR source. Approximately 60 million CDR records are dumped to our database server daily on average, with raw CDR data being deposited there every five minutes.A separate storage location is necessary due to the amount of the CDR data.

4.2 Understanding the Data

The first step in extracting knowledge from the target material is to comprehend it. Identifying fields, analyzing the values they contain, and determining their significance for this research are all part of this process. In this procedure, domain experts work together to carefully analyze the data and its structure. Data relationships with certain data-mining techniques and the problem at hand were assessed. Ethiopia Telecom, the country's telecom service provider, provided the dataset used in this thesis. It includes customer profile information and CDR dataset.

The collected Call Detail Record (CDR) dataset contains 33 fields, as detailed in Table 2. Most fields are primarily used for billing, including `CHARGE`, `Call_Fee`, `Account_Item_ID`, `Rate_ID`, `Billing_Date`, `Billing_Offering_ID`, and `Billing_Cycle_ID`. Internet usage records are represented through fields capturing upload and download volumes.

To distinguish between different service types (voice, SMSnd data each record is uniquely identified using `CDR_ID` and `RE_ID`. Additionally, `CDR_TYPE` categorizes call types as Mobmobile originatingobmobile terminatingr Forforwarded.cation information such as the cell or district of the calling and called party is captured in `CELL_A` and `CELL_B`. For privacy reasons, sensitive fields, including phone numbers and billing identifiers, have been hashed to protect user confidentiality.

Table 2: CDR Fields Description

NO	Field Name	Description
1	CDR_ID	CDR Sequence Number
2	RE_ID	CDR type ID for voice, SMS and Data
3	BILLING_NBR	Billing Number
4	CDR_TYPE	Call type ID
5	CALLING_NBR	Calling Number
6	CALLED_NBR	Called Number
7	CALLING_IMEI	Calling IMEI
8	CALLING_IMSI	Calling IMSI
9	THIRD_NBR	Third Party Number
10	START_TIME	Call start time
11	END_TIME	Call end time
12	DURATION	Call duration
13	CALL_FEE	Call fee
14	CALLED_COUNTRY	Called country
15	CALLING_CARRIER	Calling carrier
16	CALLED_CARRIER	Called carrier
17	CELL_A	Calling district
18	CELL_B	Called district
19	STATE_DATE	State date
20	CALLING_SUB_ID	Calling subscriber ID
21	BILLING_CYCLE_ID	Billing cycle ID
22	CHARGE1	Charge amount
23	CHARGE2	Charge amount
24	PRICE_ID1	Rate ID
25	ACCT_ITEM_ID1	Account item ID
26	TRAFFIC_UP	Upload traffic

Table 3 – Continued from previous page

NO	Field Name	Description
27	TRAFFIC_DOWN	Download traffic
28	BILLING_OFFERING_ID	Billing offering ID
29	ERROR_CDR_TYPE	Error CDR Indicator
30	CALL_FORWARD_INDICATOR	Call Forward Indicator
31	HOT_LINE_INDICATOR	Hot Line Indicator (voice mail)
32	CALLING_TRUNK_ID	Calling Trunk ID
33	CALLED_TRUNK_ID	Called Trunk ID

4.3 Tools Selection

To build a model, we employed various open-source libraries and tools, including TensorFlow, Keras, and Python. The following subsections provide a quick overview of these tools, which are openly accessible to all developers and researchers.

4.3.1 TensorFlow

TensorFlow is an open-source machine learning software toolkit that offers a number of Python model development and training methods [28]. It was created for internal usage by the Google Brain Team and is useful for a variety of applications, with a focus on deep neural network training and inference. With features like the Model Sub-classing Application Programming Interface (API) and Keras functional API for the design of complex topologies, TensorFlow provides flexibility and control while assisting in the development and training of state-of-the-art models with good performance. Its adaptable ecosystem of libraries, tools, and community resources enables developers to create and implement ML applications with ease and allows researchers to push the boundaries of machine learning.

4.3.2 Keras

The most popular open-source software library for deep learning frameworks, Keras serves as an interface for the TensorFlow library and offers a Python interface for ANNs [29]. Keras prioritizes usability, modularity, and extensibility in order to facilitate rapid deep neural network research. This API, which was created for people rather than computers, adheres to best practices for lowering cognitive load by providing straightforward and consistent APIs, minimizing the number of user activities needed for typical use cases, and displaying error messages that are easy to understand and actionable. Built on top of TensorFlow, Keras is a robust industry framework that can expand to massive GPU or TPU clusters.

4.3.3 Python

Python is a high-level, interpreted, general-purpose, object-oriented programming language with a large scientific library [30] that is used for complex data analysis, machine learning, and web development. Python's object-oriented approach and language characteristics are designed to assist programmers in writing logical, understandable code for both small and big projects. It supports several programming paradigms, including structured programming, especially procedural, functional, and object-oriented programming, and is garbage-collected and dynamically typed. It is frequently referred to as a "batteries included" language because of its extensive standard library [31].

4.4 Field Selection

Some of the fields in the collected CDR are useless for the proposed thesis work, and others have duplicate and empty data, as covered in depth in subsection Section 4.2 The total fields are 33, and the most important ones for the research are selected. The selected fields are described in Table 3. Additionally, since SIM-box fraud depends on mobile phone subscribers, only mobile phone users' CDRs are taken into account.

Table 4: Selected CDR Fields Description and Feature Contribution to SIM Box Fraud Detection

Feature	**Contribution (%)**	**Description**
CALLING_IMSI	14.370270	International Mobile Subscriber Identity (IMSI) of the calling device. Identifies the subscriber's SIM card.
CALLING_NUMBER	12.674316	Phone number of the caller. Used to trace origin and detect spoofed numbers.
CALLED_NUMBER	12.200631	Phone number of the recipient. Helps identify high-risk destination patterns.
CALL_DURATION	12.015202	Length of the call (seconds). Short durations may indicate fraud probes.
CALL_START_TIME	10.641217	Timestamp when the call was initiated. Identifies abnormal call timing patterns.
CALL_END_TIME	10.628406	Timestamp when the call ended. Used with start time to calculate duration.
CALLING_DISTRICT	9.963751	Geographic district of the caller. Detects unusual location-based activities.
CALLED_DISTRICT	9.801298	Geographic district of the recipient. Flags cross-border fraud patterns.
CALLING_IMEI	4.493116	International Mobile Equipment Identity (IMEI) of the calling device. Identifies compromised devices.
DOWNLOAD_TRAFFIC	1.614591	Volume of data downloaded (bytes). Anomalies may indicate SIM box misuse.
UPLOAD_TRAFFIC	1.597202	Volume of data uploaded (bytes). Less relevant for voice-focused SIM box fraud.

4.5 Sampling Selection

Sampling is one of the first required tasks to be completed before moving on to the data preparation stage. While DL employs a supervised data type—that is, labeled data with two classes—ML is applied/used to detect SIM-box fraud. The data type can be either numeric or nominal.

Table 5: Class Label [1]

Subscriber Type	Class Label
legitimate customer	0
fraudulent customer	1

. Fraudulent customer numbers are provided from the Ethio Telecom security department; there are a lot of SIM-box fraudulent numbers detected every day. The company provides 5750 SIM-box fraudulent numbers that are detected and suspended by FMS within the CDR data collection period. The ratio of fraudulent and legitimate numbers must be proportional. Most researchers proposed the ratio to be 25% fraudulent numbers and 75% legitimate numbers.
Currently, Ethio Telecom is catering to around 80 million active mobile subscribers, who are a huge number compared to the sample size used in this study. From these subscribers, 17,250 valid (non-fraudulent) subscriber numbers were randomly selected using Simple Random Sampling (SRS), ensuring that each customer had an equal chance of selection. The study has used 23,000 sample subscriber numbers in total, consisting of both valid and fraudulent users.

Also, 104,600 call detail records (CDRs) were produced for these subscribers on which the analysis is based. The detailed distribution of normal and fraudulent subscribers is presented in Table 6. Deep learning, or state-of-the-art methods for SIM-box fraud detection, typically uses customer usage data, or CDR, which is very useful for learning about customer behavior.

Table 6: Subscriber Sample Number

Subscriber Type	No of Sample	No of Total Records	Class Label
Fraudulent Customer	5750	261,000	1
Legitimate Customer	17250	784,500	0

4.6 Data Preprocessing

Preprocessing the dataset is an essential step in the implementation phase since it guarantees that the data used to train deep learning models is accurate and relevant.

In general, raw data is noisy, inconsistent, and incomplete; it takes effort to make it meaningful for deep learning. A crucial step in preparing the raw data for deep learning is data preparation.

4.6.1 Conversionton CSV Format

The first step in the preprocessing process was to convert the raw data, which was initially supplied in text files in TSV format, into CSV format. Given that CSV files are easy to handle and extensively supported by data analysis tools and libraries, this conversion was essential. The conversion procedure made sure that every item of data was correctly divided into separate columns, making manipulation and analysis easier in later stages. For appropriate data modification and analysis, the CSV files were subsequently represented as DataFrames in Python.

4.6.2 Data Cleaning

Data cleaning is a process of getting a relevant data source ready for deep learning. Remove or fill in the data's empty values, preserve data integrity, eliminate noisy data, and eliminate redundant values by retaining a single record so as not to skew deep learning. It takes time and careful attention to prevent the formation of unnecessary data at the end of the data cleansing procedure.

We cannot guarantee that data will be collected correctly; errors will occur, and we must address data quality concerns. Like any big data collection, the data used for this thesis contains certain flaws, including duplicate records, missing values, and incomplete information. To begin

cleaning this data, only calls made by mobile devices were chosen, and records including subscriber numbers other than mobile ones were eliminated. Data that has any missing values in any of its fields is also not included in the target data.Records with missing or incorrect values are eliminated (for example, if the calling number is longer than 12 characters). Records that do not have the country code (251) prefix are modified by appending a prefix to them. Repetitive records are eliminated by retaining only one copy of each record. According to the desired machine learning methodologies, the target data's validity and quality are also examined.

4.6.3 Data Aggregation

One kind of data preprocessing operation utilized on the gathered CDR data to help in providing the complete individual user information is data aggregation. One record under the raw CDR displays the activities of a single user, but it is challenging to comprehend user behavior from a single CDR record. A comprehensive view of user behavior requires the aggregation of a collective CDR record. A cumulative result from every single user over a specified period of time is called an aggregate. The behavior of the research determines the aggregation's temporal span. Understanding consumers' usage habits during the specified time period and detecting SIM-box fraud using usage data are the main goals of this study.

The SIM-box fraud may be undetected by an inappropriate pattern when the aggregation time span is shorter. However, detecting SIM-box fraud in near real time may not be possible with a high aggregation time span. To identify SIM-box fraud, a study [3] is carried out utilizing CDR data from Ethiopian Telecom. The three granularity levels it employs are 4-hour, 1-day, and 1-month. When compared to the other granularity levels, the research's cumulative findings indicate that the minimum granularity level (4 hours) performs better. Furthermore, the FMS ethiotelecom currently employs a 4-hour granularity level as the minimum aggregate time period.

The following actions have been performed on the sampled records covered in detail in Subsection 4.6 in order to prepare the input data for the desired deep learning algorithms. To generate derived aggregated attributes, which are described in Table 7, in a month, a Day and a 4-hour span of time, we followed the following steps.

- CDRs are aggregated at the subscriber level.
- Voice, SMS, and data usage are combined into one instance per subscriber per aggregation window.
- A class label is added for model training.

Table 7: Derived Attribute and Descriptions

Attribute	**Description**
MSISDN	Unique mobile subscriber ID
TOTAL_CALLS	Total calls made
TOTAL_DURATION	Total call duration
SMS_COUNT	Number of SMS sent
DATA_USAGE_MB	Data usage in MB
TOTAL_CELL	Total cell connections
TOTAL_DIS_CEL	Disconnected cell sessions
TOTAL_DIS_OUT	Outgoing disconnections
CALL_GAP	Avg. time gap between calls
CALL_COUNT_PER_IMSI	Calls per IMSI
UNIQUE_IMSIS_PER_IMEI	Unique IMSIs per IMEI
COUNT IMSI_PER_IMEI	IMSIs linked to one IMEI
COUNT IMEI_PER_IMSI	IMEIs linked to one IMSI

4.7 Dataset Formatting

Formatting is the process of re-engineering the input dataset so that it may be used with the specific deep learning method. Nominal or numeric data types are frequently used for features or attributes; achieving format consistency across all records in the entire file is crucial, as inconsistent record formats might cause issues when building models.Typically, a classification process entails dividing data into testing and training sets. As indicated in Table 9, the preprocessed

dataset used in this thesis was divided between training and testing parts. Before aggregation, the dataset contained a total of 104,600 records, of which 75% were normal and 25% were fraudulent. After applying the aggregation process, the resulting dataset is described in Table 8.

Table 8: Training and Testing Dataset Distribution Across Aggregation Levels

Aggregation	**Training Dataset**				**Testing Dataset**			
	Normal	Fraud	Total	Normal%	Normal	Fraud	Total	Normal%
4 hour	144949	48547	193496	74.91	16105	5391	21496	74.92
Daily	128142	42854	170996	74.93	14238	4758	18996	74.95
Monthly	15525	5171	20696	75.01	1725	571	2296	75.13

4.8 Model Training

In this subsection, we describe how we train deep learning models to detect SIM box fraud. We investigate the efficiency of several algorithms on diverse datasets, such as Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), and autoencoder. Our main goal is to find a compromise between maximizing true positives, which accurately detect fraudulent cases, and limiting false positives, which wrongly classify non-fraudulent cases as fraudulent.

We use the Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC) to determine the best model for SIMBox fraud detection to achieve this balance, which also contributes to the robustness and security of telecommunications networks. Models are trained using training strategies for the suggested machine learning algorithms. 10-fold cross-validation specifically.

The most popular training technique is K-fold cross-validation. [3]. In standard K-Fold Cross-Validation, the dataset is divided into k equal subsets (folds), and the model is trained and evaluated k times — each time using a different fold as the test set and the remaining k–1 folds

for training. However, this approach does not necessarily preserve the original class distribution in each fold, which can lead to biased evaluation, especially in imbalanced datasets.

To address this limitation, stratified K-fold cross-validation is employed in this study. This method ensures that each of the 10 folds maintains the same class distribution as the full dataset, thus providing a more reliable and representative performance assessment of the model. In each iteration, the model is trained on 9 folds and tested on the remaining one, and this process is repeated 10 times once for each fold. The final performance metrics are then averaged across all iterations.

4.9 Model Building

When the experimental setup is prepared, appropriate algorithms (LSTM, CNN, and autoencoder) are chosen, along with a training mode that is appropriate for this study (stratified 10-fold cross-validation technique) and an aggregated dataset mode (4H, daily, and monthly).

A total of 9 models were developed to identify SIM-box fraud using the potential combinations. The maximum number of building models using a combination of the three chosen modes (algorithms, training, and datasets) is displayed in Table 9. Each model building will be explained in detail in the coming subsections, providing detailed explanations of the models constructed using the LSTM, CNN, and autoencoder algorithms. Additionally, their classification performance was gathered and assessed. 9 models were developed in all to identify SIM-box fraud using the potential combinations.

4.9.1 LSTM Model Building

The LSTM (Long Short-Term Memory) model was developed to detect SIM-box fraud using sequential telecom CDR data. The model was trained using stratified 10-fold cross-validation for three different data aggregation levels

Table 9: List of Built Models

Aggregation Mode	Algorithm	Training Mode
4H	LSTM CNN Autoencodr	Stratified 10-Fold Cross-Validation
Daily	LSTM CNN Autoencodr	Stratified 10-Fold Cross-Validation
Monthly	LSTM CNN Autoencodr	Stratified 10-Fold Cross-Validation

Table 10: LSTM Build model

Algorithm	Aggregation	Training Mode	Build (s)	Evaluate (s)
LSTM	4H	stratified 10-fold cross-validation	6163.43	53.08
	Daily	stratified 10-fold cross-validation	5143.16	47.87
	Monthly	stratified 10-fold cross-validation	1075.39	18.59

4.9.2 CNN Model Building

Table 11: CNN Build model," presents the performance metrics for a Convolutional Neural Network (CNN) model under different data aggregation strategies: 4-hourly (4H), Daily, and Monthly. All models were trained using a "Stratified 10-cross-fold" method.

Table 11: CNN Build model

Algorithm	Aggregation	Training Mode	Build (s)	Evaluate (s)
CNN	4H	stratified 10-fold cross-validation	2824.91	26.00
	Daily	stratified 10-fold cross-validation	2451.92	22.43
	Monthly	stratified 10-fold cross-validation	525.41	7.44

4.9.3 Autoencoder Model Building

Autoencoder Build Model" presents the performance metrics for an autoencoder model under different data aggregation strategies: 4-hourly (4H), daily, and monthly. All models were trained using a "stratified 10-cross-fold" method shown in Table 12.

Table 12: Autoencoder Build model

Algorithm	Aggregation	Training Mode	Build (s)	Evaluate (s)
Autoencoder	4H	stratified 10-fold cross-validation	1848.10	22.66
	Daily	stratified 10-fold cross-validation	1625.63	21.25
	Monthly	stratified 10-fold cross-validation	347.93	6.91

4.10 Model Evaluation

We will examine the measures used to analyze the performance of our selected model in this subsection. Accuracy is a metric that indicates how well a deep learning model predicts outcomes. It is calculated by dividing the total number of accurate forecasts by the total number of predictions.

4.10.1 Confusion Matrix

Confusion matrices are one method of evaluating the effectiveness of deep learning algorithms, since the name suggests that the model becomes confused on the two classes. The true positive (TP), false positive (FP), false negative (FN), and true negative (TN) values of the classification class labels are contained in this 2x2 matrix.

Table 13: Conceptual Confusion Matrix

	Predicted Class 'N'	Predicted Class 'Y'
Actual Class 'N'	TP	FN
Actual Class 'Y'	FP	TN

- True Positive (TP): These are the cases where the actual class was positive, and the model correctly predicted it as positive.
- FP: These are the cases where the actual class was negative, but the model incorrectly predicted it as positive
- False Negative (FN): These are the cases where the actual class was positive, but the model incorrectly predicted it as negative
- TN: These are the cases where the actual class was negative, and the model correctly predicted it as negative

Instances that are correctly classified as either fraudulent or normal (legitimate) are denoted by TP and TN, respectively. On the other hand, FP and FN, respectively, detected cases of fraudulent and normal (legitimate) that were mistakenly classified.

Confusion matrix and classification accuracy are common classification measures used by a number of researchers [3, 10].

4.10.2 Performance Metrics

Accuracy: is a metric that indicates how well a deep learning model predicts outcomes. It is calculated by dividing the total number of accurate forecasts by the total number of predictions [32].It is defined as:

- TP =True Positives
- TN =True Negatives
- FP =False Positives
- FN=False Negatives

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

F-Measure: A harmonic mean of precision and recall is F-measure and is calculated as shown in Equation 4.2.

$$\text{F-Measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.2}$$

Recall:A metric called recall measures how well a deep learning model detects positive occurrences, or true positives, out of all the actual positive samples in the dataset.

It is computed as the sum of true positives and false negatives divided by the number of true positives [32]. It's described as

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.3}$$

Precision : Precision is a metric that measures how often a deep learning model correctly predicts the positive class. The number of true positive predictions divided by the total number of instances predicated as positive (including both false positives and true positives) is how it is calculated [32]. It is described as

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.4}$$

ROC curve: True Positive Rate (TPR) and False Positive Rate (FPR) are represented graphically by ROC. The X-axis shows the FPR, while the Y-axis shows the TPR. The technique is regarded as a perfect classifier when the ROC curves are too near to the upper-left corner of the area. Conversely, the technique is regarded as a low-level classifier if the ROC curves fall beneath the linear line (X=Y).

5 Results and Discussion

This chapter describes the experiment results of the research with stratified ten-cross-fold validation and applied while using selected Long Short-Term Memory(LSTM), Convolutional Neural Network(CNN), and autoencoder.

5.1 Performance Evaluation

Detecting SIM-box fraud with DL algorithms and comparing the effectiveness of each method is the primary goal of this study. SIM-box fraud detection studies have been covered in the previous chapters, and the aggregation technique is used to get the intended study outcomes in order to combat SIM-box fraud activities.

In order to detect SIM-box fraud, experiments are carried out using the selected algorithms. For each experiment, the final dataset instances were obtained using the three aggregations (4H, daily, and monthly). stratified 10-fold cross-validationvalidation procedures are used to carry out the DL algorithm experiment. The final experiment results of the model are recorded, evaluated, and compared to each other. When comparing the models, the LSTM model outperforms the CNN and autoencoder models in terms of accuracy. We generated three different models using these DL methods. On the training techniques (stratified 10-fold cross-validationvalidation), all three of the LSTM models outperform the other models built with the CNN and autoencoder algorithms. Next, the LSTM model is compared according to its aggregation mode; using stratified ten-cross-fold validation, the monthly aggregation mode gets the greatest accuracy of 99.81%. CNN models also achieve the highest accuracy when compared with autoencoder classifier models. While comparing CNN's model with each other each model achieves almost similar performance values which is about 99.78% accuracy. Each models' performance result stated in Table 16. The last but not The last algorithm, the autoencoder model, performed much worse than the other two classifier algorithm models. DL Algorithm's classification performance can be presented using graphical representation. The ROC curve is a graphical representation tool that makes it simple to show the models' performances. It primarily illustrates the relationship

between True Positive Rate (TPR) and False Positive Rate (FPR). Performance evaluation of the model based on stratified 10-fold cross-validationvalidation for four-hour, daily, and monthly aggregation modes is shown in Figures 10, 11, and 12, respectively.

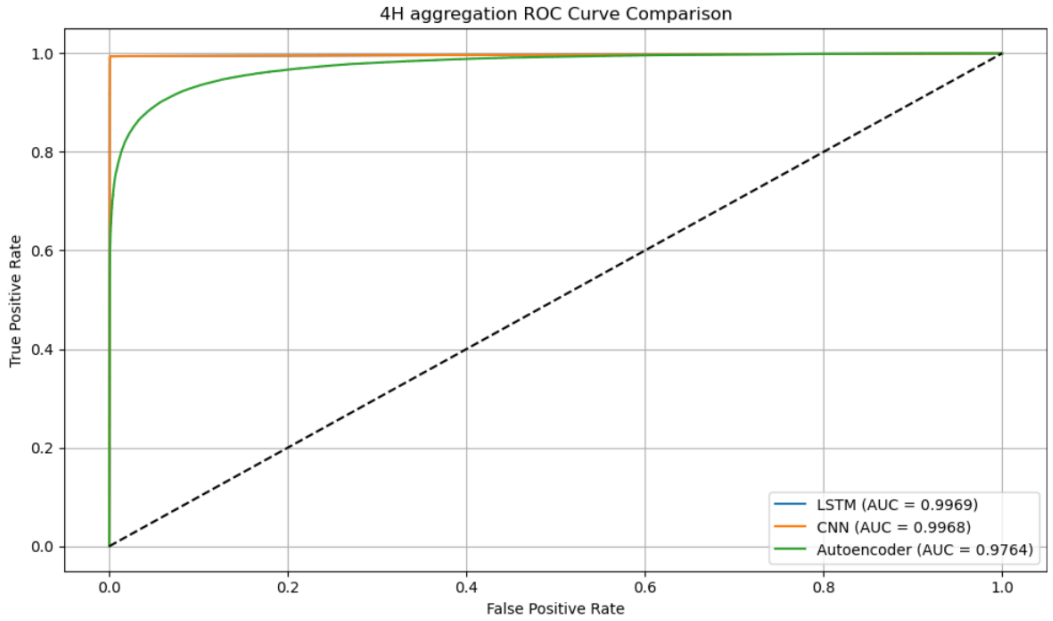

Figure 9: ROC Curve for 4H Aggregation

Figure 9 shows that the Receiver Operating Characteristic (ROC) curves for the LSTM, CNN, and autoencoder models when data is aggregated every 4 hours. The ROC curve plots the true positive rate against the false positive rate at various threshold settings for binary classification. The diagonal dashed line represents a random classifier, and curves closer to the top-left corner indicate better performance. The Area Under the Curve (AUC) value provides a single metric summarizing the overall performance.

Result/Finding: For 4-hour aggregation, the LSTM model shows an AUC of 0.9969, the CNN model an AUC of 0.9968, and the autoencoder an AUC of 0.9764. While LSTM and CNN maintain very high AUCs, the autoencoder's AUC has improved noticeably compared to the daily aggregation, indicating better discrimination ability at this aggregation level.

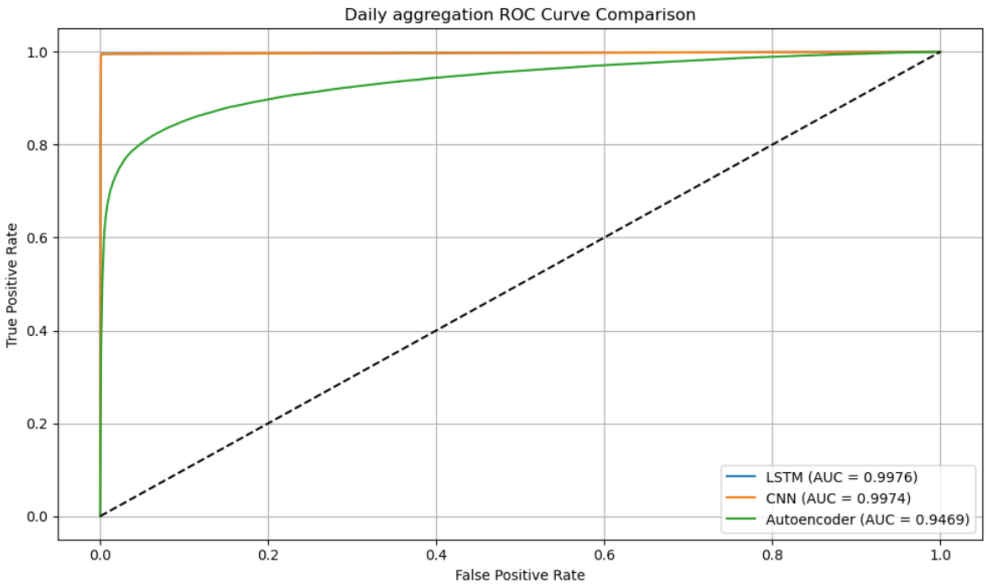

Figure 10: ROC Curve for Daily Aggregation

This figure is another ROC curve, identical in format to Figure 9, but it displays the performance of the models when data is aggregated daily.

Result/Finding: For daily aggregation, the LSTM model achieves an AUC of 0.9976, the CNN model an AUC of 0.9974, and the autoencoder an AUC of 0.9469. Both LSTM and CNN show exceptionally high AUC values, indicating their strong ability to distinguish between classes, performing significantly better than the autoencoder.

Figure 11 also displays the ROC curves for the models when data is aggregated monthly, following the same format as Figures 9 and 10..

Result/Finding: In the monthly aggregated scenario, all three models show extremely high performance. LSTM achieves an AUC of 0.9983, CNN an AUC of 0.9976, and the autoencoder an impressive AUC of 0.9972. This indicates that at a monthly aggregation level, all models are highly effective in their classification task, with the autoencoder's performance approaching that of LSTM and CNN.

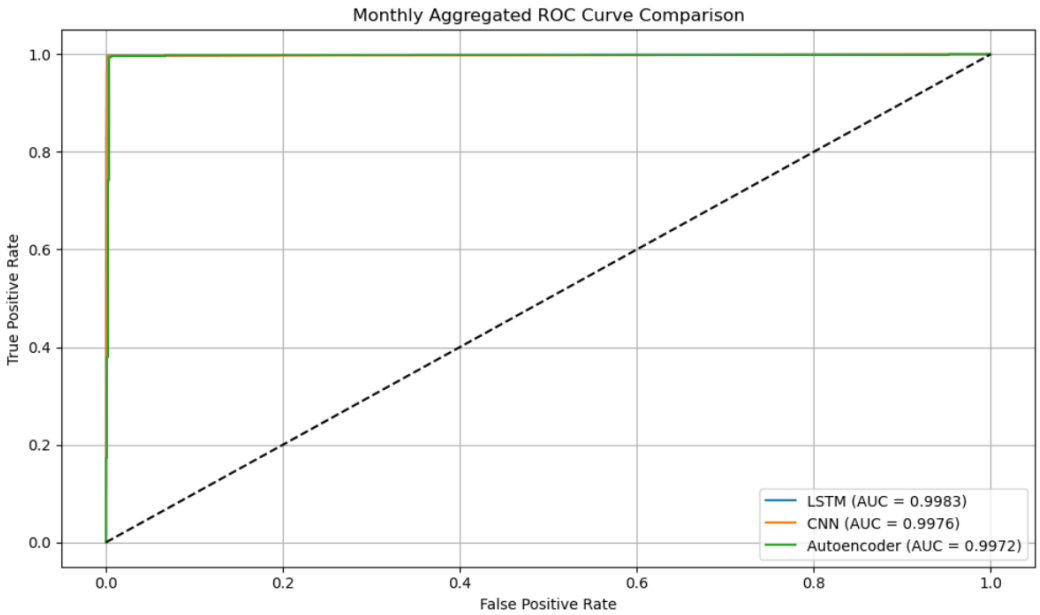

Figure 11: ROC Curve for Monthly Aggregation

The below tables consistently show that LSTM and CNN models perform exceptionally well and are highly comparable across 4-hour, daily, and monthly data aggregations. Both models consistently achieve very high recall, precision, F-measure, ROC, and accuracy scores (typically above 0.99 for most metrics and 99% for accuracy).In contrast, the autoencoder model consistently underperforms LSTM and CNN across all aggregation levels, with its performance (especially precision and accuracy) generally decreasing as the aggregation period increases (from 4 hours to daily), although it shows some recovery in recall with monthly aggregation.

Table 14: Selected Models Using 4 Hour Aggregation and comparison

Model	Recall	Precision	F-Measure	ROC	Accuracy
LSTM	0.9937	0.9968	0.9953	0.9969	99.76%
CNN	0.9939	0.9973	0.9956	0.9968	99.78%
Autoencoder	0.9303	0,7778	0.8472	0.9764	91.59%

In Table 14, LSTM and CNN models show excellent and very similar performance, both achieving over 99% accuracy. The autoencoder performs significantly worse, with 91.59% accuracy.

Table 15: Selected Models Using Daily Aggregatin and comparison

Model	Recall	Precision	F-Measure	ROC	Accuracy
LSTM	0.9950	0.9969	0.9959	0.9976	99.80%
CNN	0.9949	0.9973	0.9953	0.9974	99.76%
Autoencoder	0.8777	0,7327	0.7987	0.9469	88.72%

LSTM and CNN maintain their high and comparable performance (over 99% accuracy). The autoencoder's performance further deteriorates to 88.72% accuracy, making it the least effective shown in table 16.

Table 16: Selected Models Using Monthly Aggregation and comparison

Model	Recall	Precision	F-Measure	ROC	Accuracy
LSTM	0.9972	0.9951	0.9961	0.9983	99.81%
CNN	0.9972	0.9939	0.9955	0.9976	99.78%
Autoencoder	0.9977	0,8269	0.9043	0.9972	94.75%

Table 16 presents the performance of three deep learning models (LSTM, CNN, and autoencoder) using monthly aggregation. Similar to the previous aggregations (4-hour and daily), the LSTM and CNN models continue to exhibit very high and comparable performance across all metrics. Both models achieve recall, precision, F-measure, and ROC scores above 0.99 and accuracy rates above 99.7%.The autoencoder model's performance improves with monthly aggregation compared to daily aggregation, particularly in recall (0.9977) and ROC (0.9972), where it even surpasses LSTM and CNN in recall. However, its precision (0.8269), F-measure (0.9043), and accuracy (94.75%) still lag behind the LSTM and CNN models.

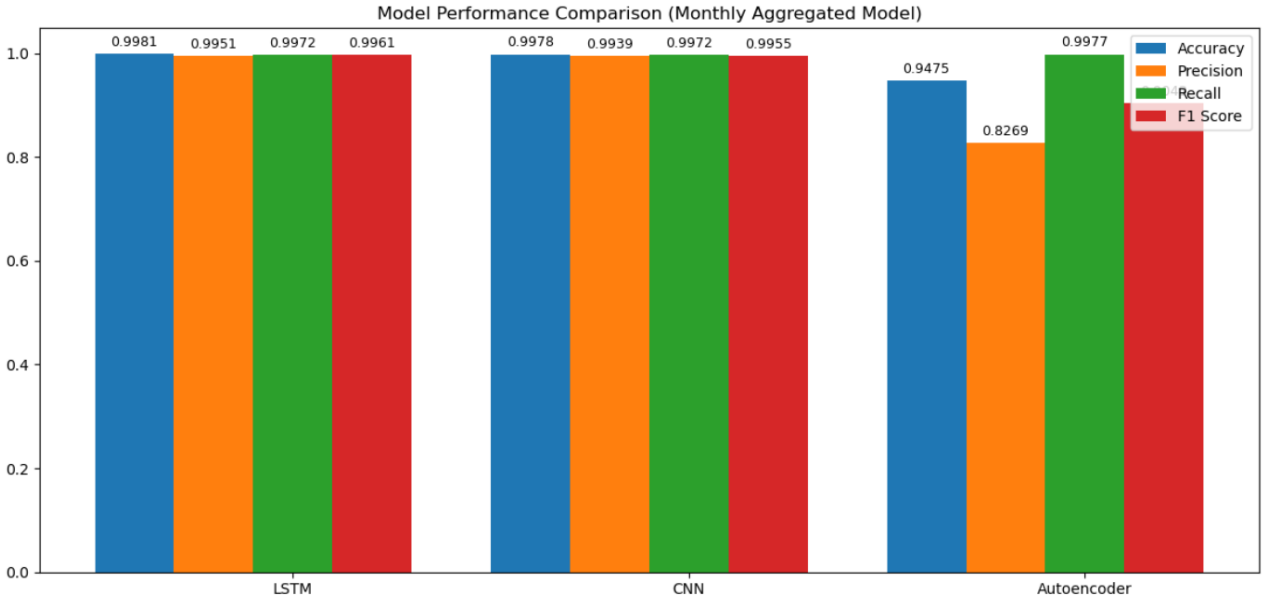

Figure 12: bar chart for Monthly Aggregation

This figure is a bar chart comparing four key performance metrics—Accuracy, Precision, Recall, and F1 Score—for the LSTM, CNN, and Autoencoder models when using monthly aggregation. Each set of bars represents a model, and the height of the bar indicates the value of the respective metric. Higher values indicate better performance.

Result/Finding: LSTM continues to show strong performance with an accuracy of 0.9981, precision of 0.9951, recall of 0.9972, and an F1 score of 0.9961. CNN is also highly effective, with an accuracy of 0.9978, precision of 0.9939, recall of 0.9972, and an F1 score of 0.9955. The autoencoder significantly improves its performance at this aggregation level, achieving an accuracy of 0.9475, precision of 0.8269, recall of 0.9977, and an F1 score of 0.9048. While the autoencoder's recall is particularly high, its precision and F1 score still lag behind LSTM and CNN, which maintain overall superior and more balanced performance.

In summary, while all models show strong performance with monthly aggregation, LSTM and CNN maintain their superior overall performance, especially in precision and accuracy, compared to the autoencoder. The autoencoder's performance in recall with monthly aggregation is notable.

6 Conclusion and Recommendation

6.1 Conclusion

Telecommunications companies in developing nations use the money they receive from international calls to offset their expansion costs. Scammers, however, take advantage of this situation by offering callers a lower charge and stealing money from operators. VoIP technology, in conjunction with SIM-Box and local SIM cards, enables them to redirect international calls and return them as local calls. Businesses can reduce their revenue loss by implementing a system that helps identify SIM-box fraudsters early and prevents the scammers from conducting business.The objective of this research work was to develop a model and propose the deployment scenarios to detect SIM box fraud in the telecommunications industry, specifically in Ethio Telecom, Ethiopia.

In this study, we examined CDR data and selected eleven relevant fields that help differentiate between SIM-box fraudulent and legitimate subscribers. Higher focus should be paid to the CDR aspects that show the beneficial user profile. The foundation for the study's success is a thorough understanding of the fraud type's behavior. More data is being produced by mobile devices, and this data contains important information regarding scams. For analysis, it is difficult to collect this large amount of data over an extended period of time. In order to overcome this difficulty, a solution must be devised.Developing a near-to-real-time analysis approach is one of the ways, and by deriving the identified attributes, we presented three dataset profiles: hourly, daily, and monthly aggregated. We have also selected three Algorisms—LSTM, CNN, and autoencoder—based on the characteristics of the fraud. Many models were created by combining these components. The models from each profile type that performed better in terms of detection were selected from the ones that were obtained.

On the monthly dataset, the LSTM models perform somewhat better than the other two. The model's accuracy on the 4-hour and daily datasets is 99.76% and 99.80%, respectively, while it is 99.81% on the monthly datasets. This indicates that as the level of data granularity grows, so does the models' performance. This is because historical user data collected over a longer

period of time generates more identifiable user patterns.

6.2 Recommendation and future work

As future work or suggestions for improving near-real-time SIM-box fraud detection, include more CDR attributes and doing ongoing research using CDR data analysis such as location and Mobile Termination ID (Receivers cell ID). Furthermore, utilizing cutting-edge techniques to shorten detection times and get closer to real-time, as well as quality reduction research of a voice conversation in order to identify SIM-box fraud.

References

[1] N. B. Ethiopia. (2019, feb) Ethiopia arrests 32 engaged in telecom fraud. https://newbusinessethiopia.com/crime/ethiopia-arrests-32-engaged-in-telecom-fraud/.

[2] M. Kashir and S. Bashir, "Machine learning techniques for sim box fraud detection," in *2019 International Conference on Communication Technologies (ComTech)*. IEEE, 2019, pp. 4–8.

[3] H. Kahsu, "Sim-box fraud detection using data mining techniques: The case of ethio telecom," Ph.D. dissertation, PhD thesis. School of Electrical and Computer Engineering Addis Ababa . . . , 2018.

[4] N. Abuhamoud, I. Alsadi, and S. Ali, "Detecting simbox fraud using cdr files and neo4j technology," in *2021 IEEE 1st International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering MI-STA*. IEEE, 2021, pp. 259–263.

[5] R. Sharifi, M. M. Majdabadi, and V. T. Vakili, "Mobile user-activity prediction utilizing lstm recurrent neural network," in *2019 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*. IEEE, 2019, pp. 1–7.

[6] O. I. Provotar, Y. M. Linder, and M. M. Veres, "Unsupervised anomaly detection in time series using lstm-based autoencoders," in *2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT)*. IEEE, 2019, pp. 513–517.

[7] L. Taylor. (2019, feb) Cfca 2019 global fraud loss survey. https://cfca.org/document/cfca-2019-fraud-loss-survey-pdf/.

[8] C. F. C. Association. (2021, December) Telecom fraud losses increasing, according to cfca report. https://transnexus.com/blog/2021/cfca-fraud-loss-survey-report/.

[9] I. Ighneiwa and H. Mohamed, "Bypass fraud detection: Artificial intelligence approach," *arXiv preprint arXiv:1711.04627*, 2017.

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

[11] M. Frehiwot. (2017, December) Analysis and detection mechanisms of sim box fraud in the case of ethio telecom.

[12] A. H. Elmi, S. Ibrahim, and R. Sallehuddin, "Detecting sim box fraud using neural network," in *IT Convergence and Security 2012*. Springer, 2013, pp. 575–582.

[13] G. Y. Koi-Akrofi, J. Koi-Akrofi, D. A. Odai, and E. O. Twum, "Global telecommunications fraud trend analysis," *International Journal of Innovation and Applied Studies*, vol. 25, no. 3, pp. 940–947, 2019.

[14] L. G. Kabari, D. N. Nanwin, and E. U. Nquoh, "Telecmmunications subscription fraud detection using artificial nueral networks," *Transactions on Machine Learning and Artificial Intelligence*, vol. 3, no. 6, p. 19, 2016.

[15] A. J. Hussain and E. Chew, "Data mining and telecommunication fraud detection using artificial neural networks," 2015.

[16] M. I. Akhter and M. G. Ahamad, "Detecting telecommunication fraud using neural networks through data mining," pp. 601–6, 2012.

[17] H. M. Marah, O. M. Elrajubi, and A. A. Abouda, "Fraud detection in international calls using fuzzy logic," IEEE, pp. 1–6, 2015.

[18] A. Abdallah, M. A. Maarof, and A. Zainal, "Fraud detection system: A survey," pp. 90–113, 2016.

[19] C.F.C.Associationetal.,œGlobalfraudlosssurvey,PressRelease,NewJersey,NJ(CFCA),vol.10, p.2013,2017., [Accessed 08-06-2025].

[20] A. Gebremeskel, "Subscription Fraud Prevention in Telecommunication using Deep Learning Approach: the case of ethio telecom," http://etd.aau.edu.et/handle/123456789/30102, 2021, [Accessed 08-06-2025].

[21] I. Murynets, M. Zabarankin, R. P. Jover, and A. Panagia, "Analysis and detection of sim-box fraud in mobility networks," in *IEEE INFOCOM 2014-IEEE conference on computer communications*. IEEE, 2014, pp. 1519–1526.

[22] Y. Bengio, I. J. Goodfellow, and A. Courville, "Deep learning'an mit press book in preparation," *Draft chapters available at*, 2015.

[23] A. Sharma, "Differences between machine learning & deep learning," *ARTIFICIAL INTELLIGENCE*, 2018.

[24] R. M. Prakash, N. Thenmoezhi, and M. Gayathri, "Face recognition with convolutional neural network and transfer learning," IEEE, pp. 861–864, 2019.

[25] I. Namatēvs, "Deep convolutional neural networks: Structure, feature extraction and training," pp. 40–47, 2017.

[26] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: concepts, cnn architectures, challenges, applications, future directions," pp. 1–74, 2021.

[27] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE access*, vol. 7, pp. 53 040–53 065, 2019.

[28] "TensorFlow - Wikipedia — en.wikipedia.org," https://en.wikipedia.org/wiki/TensorFlow, [Accessed 18-06-2025].

[29] "Keras - Wikipedia — en.wikipedia.org," https://en.wikipedia.org/wiki/Keras, [Accessed 18-06-2025].

[30] B. Kuster, "Face detection and face recognition in python programming language," in *The Seventh International Conference on Informatics and Applications (ICIA2018)*, 2018.

[31] M. Sruthi, S. Sarath, R. Sathish, and S. Shanthosh, "A fast and accurate face recognition security system," in *Journal of Physics: Conference Series*, vol. 1916, no. 1. IOP Publishing, 2021, p. 012185.

[32] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," 2020.