

Fraud Detection on International Direct Dial Calls

**A.V.V.S Karunathilaka
2020**



Fraud Detection on International Direct Dial Calls

**A dissertation submitted for the Degree of Master
of Information Technology**

**A.V.V.S Karunathilaka
University of Colombo School of Computing
2020**



Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name : A. V. V. S Karunathilaka

Registration Number : 2016/MIT/029

Index Number : 16550299



Signature:

Date: 11/14/2020

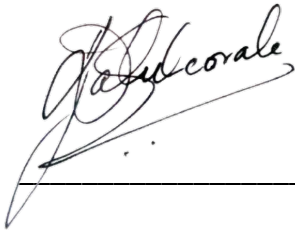
This is to certify that this thesis is based on the work of

Mr. A.V.V.S Karunathilaka

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: Dr. D. A. S. Atukorale



Signature:

Date: 19/11/2020

Abstract

Telecommunication operators' losses a considerable amount of their revenue due to fraudulent activities. These even causes poor service quality which leads to a bad customer experience. Technological advancements favor fraudsters with new tools and techniques. Subscriber Identity Module box (SIM box) bypass fraud is such a type which immersed with Voice Over IP (VoIP) technology. These are used to bypass the legitimate interconnect gateway route for International Direct Dials (IDD) calls considering the tariff is much higher than locally terminated, thus benefitting from the difference.

These activities can be detected by various approaches, such as, profiling, identifying calling patterns or test call generation, etc. But it must be in an expeditious way in order to minimize the loss.

In this research, Call Detail Records (CDR) are used in an hourly manner to differentiate fraudulent from legitimate subscribers with better performance. Artificial Neural Network (ANN) was used as the classification technique in building the prediction model. Training data set was prepared over one-month data. Results shows that the system performed with 99.59% accuracy.

Acknowledgements

I would like to express my gratitude towards,

- My Family members for their continuous support and encouragement.
- My supervisor Dr. D.A.S Athukorala for his constrictive expertise comments and advice.
- International Telecommunication Unit of Mobitel (Pvt.) Ltd for providing me with data and hardware.

Table of Content

Abstract.....	ii
Acknowledgements.....	iii
Table of Content	iv
List of Figures	vi
List of Tables	vii
List of Abbreviations	viii
Chapter 1. Introduction	1
1.1 Structure of the Thesis.....	2
Chapter 2. Background	3
2.1 Analysis	3
2.2 SIM-Box Fraud Scenario	5
2.3 Review of Similar Systems and Technologies	7
Chapter 3. Methodology.....	9
3.1 Machine Learning.....	9
3.1.1 Supervised Learning.....	9
3.1.2 Artificial Neural Network	10
3.1.3 Perceptron	11
3.1.4 Sigmoid.....	12
3.1.5 ReLU	13
3.2 Data Collection and Feature extraction	15
Chapter 4. Design.....	18
4.1 Data Pre-Processing	18
4.1.1 Data Loader Module	19
4.2 Training the Neural Network	20
4.2.1 TensorFlow.....	21
4.2.2 Keras.....	21
4.2.3 Model Building	21
4.2.4 Test Data Preparation	23
4.3 Prediction	24
4.4 UI Design	25
4.4.1 Dashboard.....	25
4.4.2 Number Search	26
4.4.3 Reporting.....	26

Chapter 5.	Evaluation.....	27
5.1	Under Sampled	28
5.2	Over Sampled.....	29
5.3	Comparison	30
Chapter 6.	Conclusion.....	31
6.1	Future Work.....	31
Chapter 7.	Bibliography	33
Chapter 8.	Appendices.....	35

List of Figures

Figure 1 - Global Bypass Fraud Losses	1
Figure 2 – CFCA Survey 2017	3
Figure 3 - Legitimate Route of International Call, adopt from [7]	6
Figure 4 - SIM-Box Fraud Rout of International Call, adopt from [8].....	6
Figure 5 - A Perceptron with Multiple Inputs and Single Output [11]	11
Figure 6 - Sigmoid Function	12
Figure 7 - ReLU Visualization.....	13
Figure 8 - A Multilayer Perceptron Neural Network [11]	15
Figure 9 - Loader module high-level diagram	19
Figure 10 - Visual representation of ANN	22
Figure 11 - Divide by MSISDN	24
Figure 12 - Overall System Architecture	25
Figure 13 - Dashboard Screen	25
Figure 14 - Number Search Screen	26
Figure 15 - Reporting Screen.....	26
Figure 16 - Selected Network Model	27
Figure 17 - Confusion Matrix (MO Unique > 3) – SET1	28
Figure 18 - Confusion Matrix (MO Unique > 5) – SET2	28
Figure 19 - ROC (MO Unique Calls > 3) – SET1.....	29
Figure 20 - ROC (MO Unique Calls > 5) – SET2.....	29
Figure 21 - Confusion Matrix (Over Sampled) – SET3.....	29
Figure 22 - ROC (Over Sampled) – SET3.....	30

List of Tables

Table 1 - CDR Format	17
Table 2 - Input Parameters	22
Table 3 - Test Data Preparation	23
Table 4 - Final Dataset.....	24
Table 5 – Training data sets and Accuracy.....	27
Table 6 - Under Sampled (MO Calls > 3) – SET1.....	28
Table 7 - Under Sampled (MO Calls > 5) – SET2.....	28
Table 8 - Over Sampled Results – SET3.....	29
Table 9 - Model Comparison.....	30

List of Abbreviations

ANN	Artificial Neural Network
AUC	Area Under Curve
CDR	Call Detail Record
CI	Cell ID
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
ISO	International Organization for Standardization
LAC	Location Area Code
ML	Machine Learning
MO	Mobile Originated
MSC	Master Switching Center
MT	Mobile Terminated
ROC	Receiver Operating Characteristics
SIM	Subscriber Identity Module
SIM Box	Subscriber Identity Module Box
SMS	Short Message Service
VoIP	Voice Over Internet Protocol

Chapter 1. Introduction

Fraud in telecommunication industry can be defined as, “any act of cheating and embezzlement in the use of telecommunications facilities that is intentionally committed by persons or organizations to avoid the cost of services or tracking recorded conversation”. In other words, it can also be defined as, abusive usage of a telco operator infrastructure and resources by a third party without the intention of paying them. It is a major concern since it is affecting the company financially.

Mobile operators are constantly and severely affected by security threats. Fraudulent activities in International Direct Dial (IDD) calls are one of the greatest causes of financial losses to the industry every year. In parallel with rapidly evolving services in the telecommunication industry, international fraudsters always get in line with developing more and more elaborate attack techniques. These activities impact directly on the company’s revenue and potentially will be reflects as increases in tariffs. Therefore, it is necessary for telco providers, governments, and users to establish and facilitate technical, political, economic, and social measures to prevent these kinds of fraud. Success in this cause will benefit all parties except the fraudsters.

Fraudulent activities in telecommunication industry are commenced in many ways. Here I am going to focus on IDD SIM Box bypass fraud, where the adversaries route calls illegally bypassing official and legitimate interconnection points.

In recent years Sim Box Bypass Fraud or Interconnect Bypass Fraud has been declared as one of the fastest growing types of frauds. As per 2017 Global Fraud Loss Survey by CFCA, Global Fraud Loss Estimate stands at \$29.2 Billion (USD) annually which is 1.27% of global telecom revenues. See Figure 1



Figure 1 - Global Bypass Fraud Losses

1.1 Structure of the Thesis

Thesis is divided in to six main chapters. In which the next coming chapter, a detailed analysis is presented followed by a discussion about past work on the problem domain, presenting relevant references and sources.

Next, in Methodology chapter, starting with an explanation on the machine learning methods and algorithms used in the project, how the data set is obtained and how to prepare data for the analysis is discussed.

Afterword, the Design, chapter definiens, which features are extracted from the data set and gives a detailed view of all the different modules used in the project. These are compiled into subsections, explaining technologies used, reasons for why they best suit the purpose and the architecture of modules. It also gives an understanding of how the test data sets are selected, prepared and what database systems are used, and how data is stored.

Evaluation phase is one of the most crucial part in any project. In Evaluation chapter, results of all the test phases will be laid out in detail with their respective statistical measures.

Finally, in Conclusion chapter, evaluated results are discussed and compared with the aim of selecting the best prediction model to be used in the live system. Report is concluded with a small section for Future Work which explores possibilities for improving and extending the system.

Chapter 2. Background

2.1 Analysis

In the telecommunications field, service providers store vast amount of data of their client's activity for various purposes. These records contain, both normal and fraudulent activity records. Fraudulent activity records should be expected to be substantially smaller than the normal activity, otherwise the business would be impractical.

Bypass fraudsters operates by offering to connect incoming international direct dialed voice calls to local numbers by redirecting them through Voice Over IP (VOIP) Gateways toward fixed or mobile subscribers. This VOIP gateway is commonly named as a SIM Box.

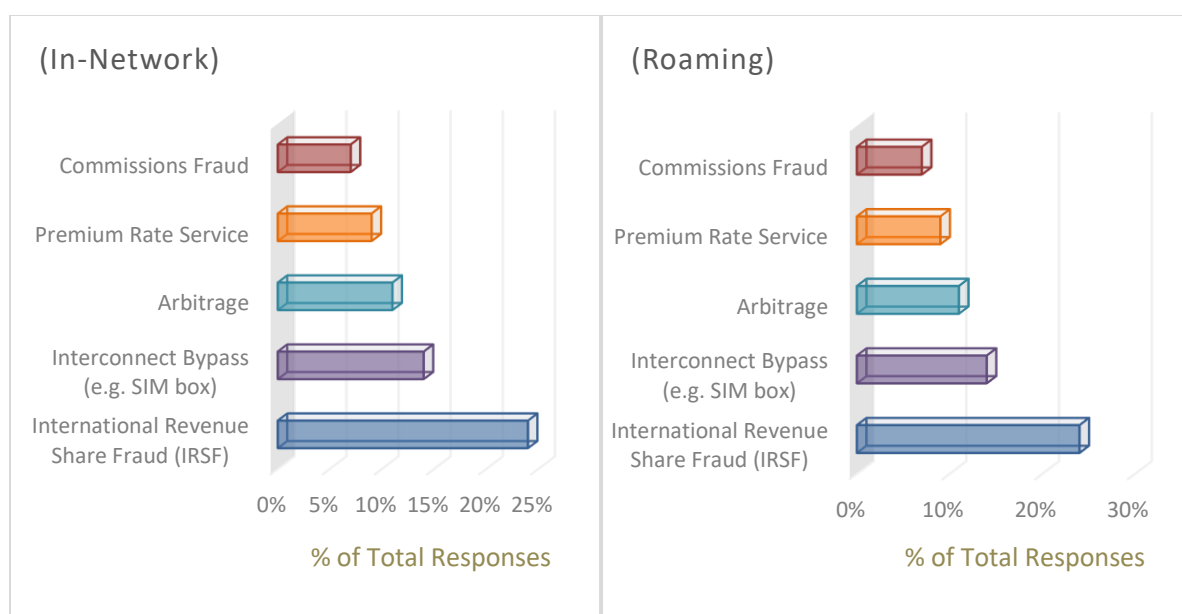


Figure 2 – CFA Survey 2017

Figure 2 is referred from the survey conducted by CFA (Communications Fraud Control Association) in 2017 clearly shows that SIM Box type frauds are a major concern in terms of finance.

SIM Boxes ranges from simple wireless GSM modules to very complex distributed nodes that virtualizes SIM cards that spans across the country. Ultimately, an international voice call which should terminate at a local number is connected to a SIM Box via VoIP and from there the call is connected to the local subscriber which is diverted through a SIM card in the SIM Box. This way, fraudster will be tricking the operator into charging the call as a locally

originated call thus the cost is reduced as, typically the local call charge is way lower than internationally terminated calls.

In detecting International bypass fraud, Test Call Generation (TCG) was one of the oldest and original methods developed. This is a very simple method. Telecommunication provider first provisions some test numbers and then generates international calls to them from large variety of providers. Upon receiving the calls, the incoming calling line identity (CLI) is tested against the original incoming CLI. Properly terminated call with legitimate routing involved should have the A-Party CLI intact. This way, a difference in incoming CLIs can identify the SIM Box numbers therefore blacklist those numbers. For many years, TCG was used in detecting International Bypass Fraud

Unfortunately, fraudsters benefit so much from international voice bypass for them to simply give up. As the years pass and technology advancing, fraudsters developed many anti-detection schemes to avoid detection by TCG. These include decoys, forking, white listing and plain old internal fraud. As the technology advanced, so did the of SIM Box capabilities armed with anti-detection schemes. This significantly decreased the effectiveness of TCG.

Mobile Switching Center (MSC) creates a Call Detail Record (CDR) when a transaction passed through it. In this project I am using MSC CDR's as activity records since it contains information related to a single instance of telephone call or other related transactions.

Traditional detection systems either randomly blocks some percentage of calls which are detected and declared as fraudulent calls from their algorithm or make detections from querying static data and aggregating values over a large time window or generating a set of features. These processes are either less accurate or time consuming and during this time window a fraudster can make number of successful bypass attempts before being identified and blocked. This is why we need to consider smaller time slots and make predictions in real time or near real time. In addition, system should have the ability to adapt and change since fraudsters always changes their behavior by the nature.

The solutions available in research literature lacks in time sensitivity or uses traditional databases which requires large computing power thus again consumes more time to provide

an output. Some of them has less awareness about problem domain or context and is not able to detect complex patterns in CDRs.

2.2 SIM-Box Fraud Scenario

In SIM Boxes, local SIM cards are used for rerouting/bypassing international calls from mobile network operators then transfer them over the Internet and deliver them back by means of VoIP gateway device called SIM-Box, as local calls to the operator's cellular network

A SIM-Box is a VoIP gateway, that connects international calls from VoIP to a local mobile operator using domestic SIM cards. SIM-Box fraud is a type of fraud that has developed with the growth of VoIP technologies, and its successes is depend on the obtainability of SIM card and SIM-Box devices. In countries that have less control on the distribution of SIM cards and availability of SIM-Box devices as observed in some countries of Asia and Africa, this fraud type is a main challenge [8]. SIM-Box equipment includes SIM slots, antennas, and Ethernet ports that can be used to get the SIM-Box equipment connected to the Internet [2]. Fraudsters equip SIM Box with multiple local SIM cards, by means of this setup the fraudsters can forward international calls through local numbers of the operator to make it act as local call which bypasses interconnect charges.

- Subscriber A and subscriber B Reside in different countries, country A and B respectively. In legitimate route of an international call:
- Subscriber A places a call to subscriber B over the mobile operator and pays the service provider for the call.
- The call generated by subscriber A forwarded to international gateway in country A. The home international gateway of country A routes the received call to a transient operator and pays for it.
- The transient operator then routes this call to a destination (country B) international gateway and pay a toll to the destination international operator.
- Finally, the international gateway of country B terminates the call through his network to subscriber B



Figure 3 - Legitimate Route of International Call, adopt from [7]

Whereas in SIM-Box fraud route of an international call [1]:

- Subscriber A places a call to subscriber B in the domestic mobile operator network and pays it for the call.
- The call generated by subscriber A forwarded to home international gateway in country A.
- The home international gateway of country A routes the received call to a transient operator and pays for it.
- The transient operator then routes this call to a SIM-Box placed in country B using VoIP and pay a toll to the SIM-Boxer.
- The SIM-Box then places a separate call on the network of country B to subscriber B using its local SIM card, that is why it looks like a local call, and pay only for the local call by avoiding interconnect cost.
- Finally, subscriber B in country B receives an international call from abroad but with a local number.

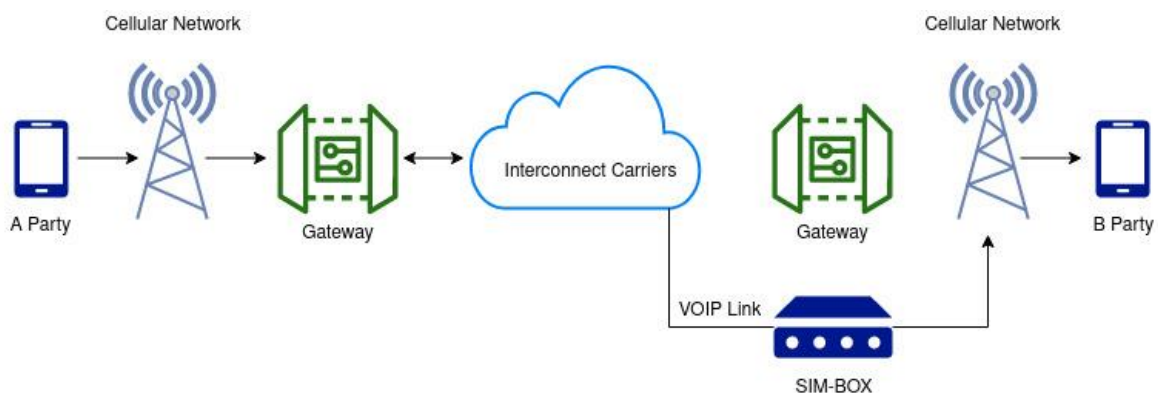


Figure 4 - SIM-Box Fraud Route of International Call, adopt from [8]

2.3 Review of Similar Systems and Technologies

A number of researches has been conducted using different tools and techniques or methods. In this section, we review some work related to SIM-Box detection using machine-learning techniques.

A research by Sallehuddin et al., [2] with the title “Classification of sim box fraud detection using support vector machine and artificial neural network.” They design and compare two classifiers SVM and ANN. A CDR data of 234,324 calls made by 6,415 subscribers from only one Cell, so it contains only one cell-ID, for a period of two months were collected. The dataset consisted of daily aggregated 2,126 fraudulent subscribers and 4,289 legitimate subscribers, which are of two thirds normal and one third SIM-Box fraud. The researchers extracted nine features, such as Total Unique Numbers Called, Total Minutes, Total Calls and Average Call time (min), etc. Then extracted features were used to train the two proposed classifiers. They trained the algorithms applying the prepared dataset and variety of parameter settings. Finally, they found that accuracy of SVM is higher compared to ANN. SVM was having 99.06% accuracy with lesser training time, while ANN was having 98.69% accuracy

The paper [3] recommends assessing the changes on patterns of usage behavior of subscribers over a period for fraud detection. CDR data requires analysis to make them applicable for pattern generation and input for data mining techniques, as they are unstructured and unorganized. In article [4] a rule-based approach is used to detect fraudulent call activity. Method described, had used CDR data sampled over two iteration periods (study and test). The study period contains CDR of customers with non-anomalous behavior. Customers were grouped according to their similar usage behavior such as average number of local calls per week and so on. They come with a probabilistic model to describe their usage for customers in each group. Then the thresholds were determined by calculating change within a group.

A classifier comparison research made by AlBougha, [5] which compares the detection performance of four data mining classification algorithms in detecting SIM-Box fraudulent subscribers from a real CDR data. They trained the classification algorithms applying daily aggregated and labeled dataset. They found that among the Classifiers, Logistic, Boosted Trees, Support Vector Machine (SVM), and Artificial Neural Network (ANN), Boosted Trees and Logistic performed the best.

Another research was conducted by Murynets et al., [6] with the title “Analysis and detection of SIM box fraud in mobility networks.” They applied supervised classification techniques. The classifiers are linear combination of three-decision trees (functional tree, random forest and alternating decision tree). They applied it on real CDR data form an operator in USA, a larger dataset was used with accounts that are nationwide distributed. The data is collected having 500 of fraudulent accounts and 93000 legitimate accounts. To train the classifier, they divided the dataset to one third for testing and two-thirds for training. Using IMEI as the device

identifier, which is unique to every device, other than the subscriber identifier which is unique to every subscriber. They identified 48 distinctive patterns as features of fraudulent and legitimate devices. They observed that fraudulent SIM-Boxes has following common patterns. Static physical location, High number of International Mobile Subscriber Identity (IMSI)s registered per single device, large number of international phone calls and large number of outgoing calls originated compared to incoming calls, and using Random Forest (RF) the classifier attained an accuracy of 99.95% with lowest false positive.

In contrast, to the use of CDR for detecting SIM-Box, Reaves et al. [7], they used real time call audio analysis. They designed a system that analyses raw voice signal data received for incoming calls to distinguish if it has a legitimate mobile voice footprint or has passed through VoIP. They were using fast signal processing techniques for the process in identifying fraudulent calls originated from SIM Boxes. System was able to detect over 87% real SIM Box calls with no false positives in analyzing only 30 seconds of audio.

In above mention works, they have considered wider data aggregation period, i.e. daily, weekly, monthly, etc. Even though this may increase generalization accuracy, it also allows a wider window for fraudsters to operate and make money. Subsequently narrow data granularity level is recommended in order to adopt near-to-real-time detection scheme. However, lesser data has lesser distinguishing patterns for detection.

In this project, I have considered hourly sets of data for training and also processed on a larger daily set for comparison. Moreover, I pre-processed raw CDR's to derive some features such as number of unique and total LAC's or number of Unique A and B party SMS's. At operational stage, the plan is to stream near real-time MSC CDR's to the system for stripping down irrelevant fields, pre-process, and record them in hourly tables. Then the records are sent through the trained network and results will be stored with accuracy levels.

Chapter 3. Methodology

3.1 Machine Learning

Machine learning is described as the complex computational process of automatic pattern recognition and intelligent decision making based on training sample data [8]. Machine learning is a sub field of AI (Artificial Intelligence) and can be deliberated as a method or technique that enables computers solve problems by learning in the problem domain. Using machine learning, we can detect patterns that are meaningful in a given data set [8] [9]. These algorithms has potential and are used in many application domain such as, data mining, which has patterns that are hidden and it can dynamically adapt to changing conditions [10]. Machine learning methods or algorithms can be evaluated by running a prediction on the same test data set that were used to train the network or by using a separate pre-sampled dataset with known outcomes and comparing final results.

There are four general machine-learning methods

- Supervised
- Un-supervised
- Semi-Supervised
- Reinforced

3.1.1 Supervised Learning

Supervised models are illustrated as learning a function, which is $f(x) = y$, where y is the label (also called class) of the data and x denotes the attributes of these examples (also called features). Supervised learning models trained with data that have been pre-classified [11]. The examples of input/output functionality referred to as the training data. Caution is required in order to ensure that the training data is correctly classified. The supervised learning methods are categorized based on the objective functions and the structures of learning algorithms. Popular categorizations include SVM, ANN, and decision trees [12]. In the case of fraud detection, since legitimate calls occur more often than fraudulent calls, the training data will mostly contain legitimate calls, leading to a misclassification of the model. This needs attention in the supervised learning models. In the upcoming sections discussed some popular supervised machine Learning algorithms.

3.1.2 Artificial Neural Network

Artificial Neural Network (ANN) is a system based that is on the biological neural network, such as the brain. According to [13], neural networks represent a brain symbol for information processing. ANN has the ability to learn for the data that we feed to it through iterating them over the network whilst adjusting its bias levels and synaptic weights. They are also able to improve their performance through learning. There are many varieties of learning algorithms that we can use in designing an ANN. These algorithms differ from one another since the adjustments they make on synaptic weights and biases in formulating a node or neuron are unique to each. Learning algorithms can be described as a set of well-defined rules that are prescribed for the solution of a problem. Error-correction, Boltzmann, memory-based and competitive learning are among the learning algorithms for ANN. ANN learning paradigm is either supervised (associative learning) or unsupervised (self-organizing). In the case of supervised, there is a need to train or teach the input and output pattern. But for the case of unsupervised neural network, it only requires input patterns. Then it develops its own representation of the input event.

ANN can be classified in to [13]

- Feed forward Neural Network
- Recurrent Neural Network
- Self- Organizing Map

In Feed Forward Neural Network, activation is piped through the network from input units to output units. Sometimes they are also known as static networks. No explicit feedback connections are defined. Conventional Feed Forward Neural Network are able to approximate any finite function given satisfactory hidden nodes that are needed in accomplishing this. It is the first and simplest type of ANN. Recurrent Neural Network on the other hand, are dynamical networks with cyclic path of synaptic connections, which is used for handling time-dependent problems and paths are acting as memory elements. Self-Organizing Map mainly used for cluster analysis. The big developments in ANN during the past few decades have motivated human ambitions to create intelligent machines with human-like brain. Nowadays, ANNs is considered as, one of the most efficient pattern recognition and classification tools [13] [14].

Where ANN is being used to solve a problem in a supervised machine learning approach, set of weights has to be determined to minimize the classification error. Least mean-square convergence is a well-known method in many learning paradigms. The objective of ANN is to minimize the errors between the ground truth Y and the expected output $f(X; W)$ of the network as, $E(x) = (f(X; W) - Y)^2$. The behavior of ANN depends on both the weights and the transfer function, which are specified for the connections between neurons. Implicitly, ANN models will be defining the relationships between input values and output values, and used to solve for complex pattern recognition problems, especially when the relationship between variables cannot be clearly determined by human mind.

3.1.3 Perceptron

Perceptron is the simplest kind of ANN, which contains a single neuron which has multiple inputs and only one output. Perceptron's are used to classify linearly separable classes. A perceptron is fed with real valued vector as inputs, then it will calculate the combination of inputs in a linear manner and outputs 1 or -1 based on a threshold or a selected function. The goal is to determine a weight which will cause perceptron to produce the correct or accurate output based on training data. This is called as the learning problem. Typically, in order to get an accurate prediction vector, training data is iterated over and over the prediction algorithm in batches. The resultant prediction rule is then used to predict and compare on a test data set.

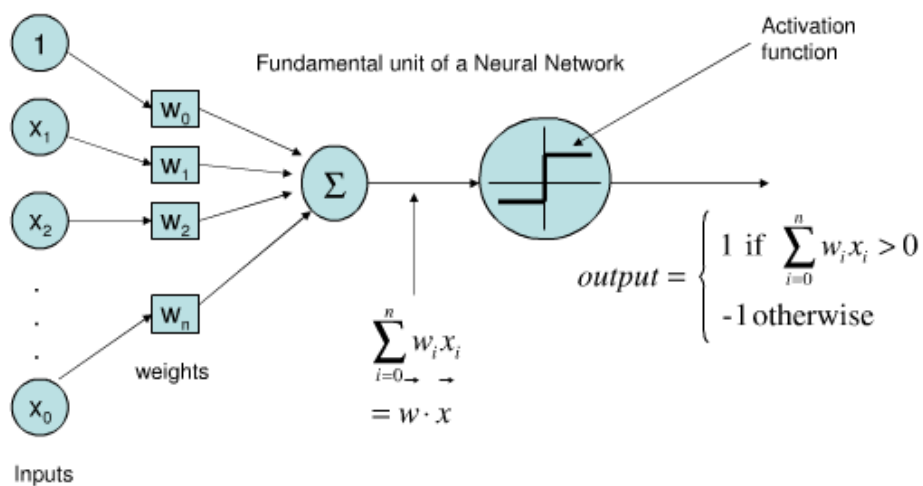


Figure 5 - A Perceptron with Multiple Inputs and Single Output [11]

In perceptron model, the weighted sum is calculated using,

$$\sum_{j=1}^m w_j \cdot x_j = w_1 \cdot x_1 + \dots + w_m \cdot x_m$$

Then evaluated and passed to an activation function, which compares it to a predetermined threshold Ω . If the weighted sum is greater than the threshold Ω , then the perceptron fires and outputs 1, otherwise outputs 0 (-1).

There are many of activation functions that can be used. ReLU, sign, linear, step and sigmoid functions are the most used activation functions.

3.1.4 Sigmoid

The Sigmoid Function has an S-shape curve (Figure 6). The sigmoid function exists between 1 and 0. This is why it especially used on models that are used to solve probability problems. Since probability of everything only ranges between 1 and 0, and easy to compute, sigmoid is the best option. [15]

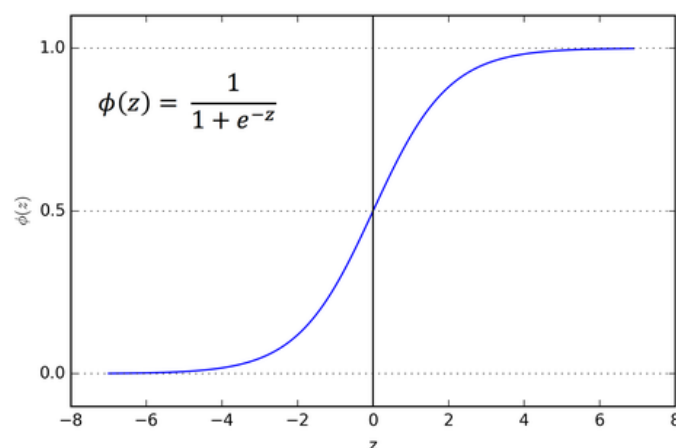


Figure 6 - Sigmoid Function

The function is differentiable. Which means, slope of the curve can be calculated at any two points. A sigmoid function is a bounded, differentiable, real function that is defined for all real input values and has a non-negative derivative at each point. A sigmoid "function" and a sigmoid "curve" refer to the same object. [16]

3.1.5 ReLU

Rectified Linear Unit (ReLU) is also a commonly used as activation function lately. Mathematically, it is defined as, [17]

$$y = \max(0, x)$$

ReLU is visually represented in below graph,

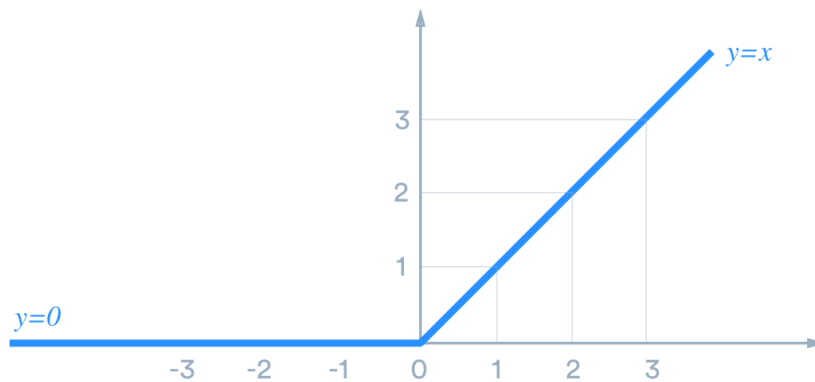


Figure 7 - ReLU Visualization

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

- No complicated math is involved. Therefore, it is easy to compute. This improves performance of training or prediction.
- It converges faster. Linearity means that the slope does not plateau, or “saturate,” when x gets large. It does not have the vanishing gradient problem suffered by other activation functions like sigmoid or tanh.
- It is sparsely activated. Since ReLU is zero for all negative inputs, it is likely for any given unit to not activate at all.

For a classification problem that has linearly separable classes, only a single perceptron is sufficient.

A single perceptron is enough to solve any classification problem that has linearly separable classes. If two or more nonlinearly separable classes exist, a network with single perceptron will not be enough to solve the problem. Such a nonlinearly separable problem is solved by using most popular types of ANN Multilayer Perceptron (MLP) [18]. In a MLP neural network,

inputs of each perceptron are connected to other perceptron's from previous layer. The perceptron will fire based on the threshold defined and weighted sum of inputs. Ideally ANN/MLP is composed of three layers. Input layer, hidden layer, and the output layer. The input layer comprises of nodes that are equal to number of system variables. Information flows through the hidden layers to output layers. Weight factors that are associated with every connector controls the flow. Nodes defined in output layer, denotes the classification decision of the system. The values of the output nodes are compared with limits to determine the prediction output and classify each input.

The training process is the iteration of input values which has predefined classification, over the network, which adjusts the weights. As this process continue to iterate, the weight values will get optimized which ultimately gets minimized to an error function. To verify the performance of the network, it is tested against testing samples which were prepared at the initial stage. As [14] stated, training is defined as the process of iterating through the training data set to obtain optimal weights which are adjusted at each iteration. To learn a neural network, random weights and biases are generated at first. Then, a training instance is passed to the neural network, where the output of each layer is passed to the next layer until computing the predicted output at the output layer, according to the initial weights. The difference between predicted values and actual values can be defined as output error. According to the computed error at output layer, the weights between the output layer and the hidden layers are adjusted/corrected, and then the weights between the input layer and the hidden layer are adjusted in a backward fashion (The best known example of a neural network training algorithm back propagation). Another training instance is passed to the neural network and to the process of evaluating the error at the output layer, thereby correcting the weights between the different layers from the output layer to the input layer. Repeating this process for as many epochs will help in learning the neural network.

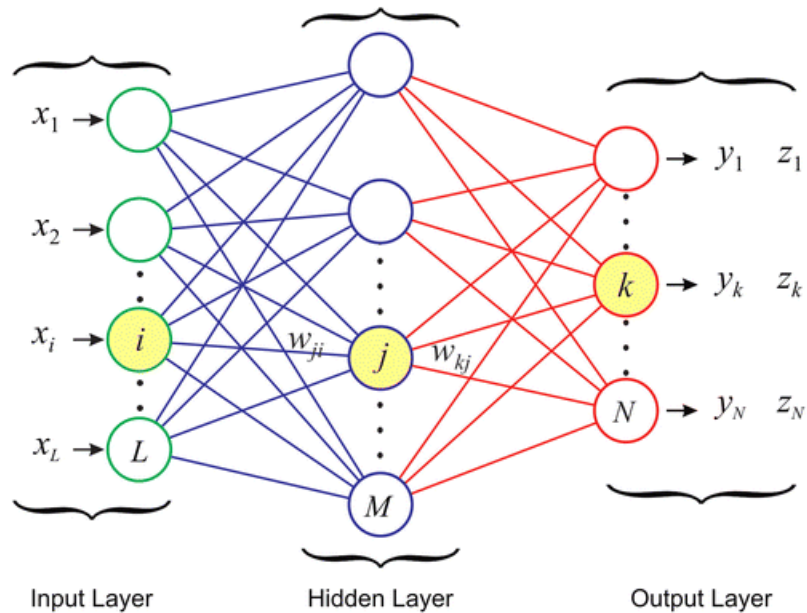


Figure 8 - A Multilayer Perceptron Neural Network [11]

ANN's are popular because of their qualities such as, ability to learn and generalize adaptability, robustness, parallel data processing and fault tolerance. These qualities enable them to solve complex multi input-output relationship and non-linear problems. And also, are quite useful in practical applications, since the capability of parallel processing, ability of non-linear mapping, ability to learn from the problem domain and their subsequent adaptability to the environment. ANN has shown to be effective and superior as a technique of modeling for non-linear data sets compared to other models. This superiority enables it to be applied for prediction and data fitting. Forecasting, pattern recognition, classification and prediction are some of the business applications and use cases where neural networks are being used [13].

3.2 Data Collection and Feature extraction

First and far most crucial problem in machine learning is finding and deciding on set of features in a given data set to be used in constructing a classification model. Recent research has shown that, constructed ML algorithms significantly underperforms, if extracted feature set contains, redundant or irrelevant information [14]. If selected data set is noisy, unreliable or if it contains redundant or irrelevant information, it will be challenging to determine optimum weights at the training stage, since the data is too unstable. Data preparation involves processing the raw data so that machine-learning algorithms can produce a

structural description of the information that is implicit in the data. It defines process and makes suitable to a data mining technique. It is the first and most important step in machine learning and plays a crucial role and has a huge impact in entire process.

The objective to detect SIM-Box fraud analyzing CDR data to identify anomalies behavior of fraudulent subscribers. The collected large amounts of CDR need to be organized to form patterns and scenarios of normal usage and fraud situations. Real-world data are typically noisy, massive in volume, and may originate from a bulk of heterogeneous sources; hence, knowing the data deeply is a vital prerequisite for data preparation. Prior to modeling and evaluation, raw data should be prepared. Data preparation involves having a closer look at fields and data values, knowing what the data implies, identifying the fields that make the data and type of values they contain and their behavior, to get a better sense of it. This helps in fixing inconsistencies incurred during data integration, identifying existence of outliers and extremes, extract similarity and deference's of data objects with respect to others.

For this project, MSC CDR's of Mobitel (Pvt.) Ltd. Is used. Whenever subscriber receive or make a call or SMS a CDR is generated, which contains complete information of the call such as IMSI, IMEI, A-Party Number, B-Party Number, etc.

A row CDR is a one pipe (|) separated line in a text file. Refer *Appendix A.2* for a row CDR. Field description of the CDR is shown in Table 1.

Data structure of a raw CDR

Field	Name	Description
1	Call-type	{moCallRecord(0), mtCallRecord(1), moSMSRecord(6), mtSMSRecord(7), moeCallRecord(12), mcfCallRecord(13), forwardCallRecord(100)}
2	MSISDN	{servedMSISDN}
3	A-Number	{Calling Number}
4	B-Number	{Called Number}
5	C-Number	{CallForwarded Number}
6	Event date	{yyyymmdd}
7	Event time	{hhmmss}
8	Duration	{in seconds}
9	Service center	{SMS service center}
10	MSCid	{switch id}
11	cell-id [LAC]	{For MO it is source cell id and for MT it is destination cell id.}
12	cell-id [CI]	{For MO it is source cell id and for MT it is destination cell id.}
13	IMEI	{}
14	IMSI	{}
15	Call-drop flag	{Map the available value in the field "causeForTerm"}
16	Partial flag	{Requires only the aggregated final CDR}
17	MSRN	{Map the available value in the field "roamingNumber"}
18	In-Trunk Group	{}
19	Out-Trunk Group	{}
20	Call Reference	{Call merging is required}
21	diagnostics	{Ex:- diagnostics = 00 90 00 00 00; Mediation need to output the value of "diagnostics" field}
22	CGA-id	{Map the available value in the "changeOfglobalAreaID"}
23	Event type	{Print Post-paid = "POS" / Pre-paid = "PRE"}
24	Vendor TAG	{Print Huawei = "HW"}

Table 1 - CDR Format

Chapter 4. Design

4.1 Data Pre-Processing

CDR mediation feed is received to our server location (/CDR) in a near-real-time manner. Since Mobitel (Pvt.) Ltd. has MSC's of two vendors' (Huawei and ZTE), mediation is pushing the files to two location from each vendor. Regardless of vendor separation, mediated CDR format is similar. So, two identical loading modules are used.

After analyzing and careful consideration, selected below parameters as the input of our neural network.

- Mobile Number
- Number of Unique MO SMS
- Number of Unique Location Area Codes for MT SMS
- Number of Unique MT SMS
- Number of Unique Location Area Codes for MT SMS
- Number of unique MO calls
- Number of unique LAC's of MO calls
- Number of unique IMEI's of MO calls
- Total MO call minutes
- Number of unique MT calls
- Number of unique LAC's of MT calls
- Number of unique IMEI's of MT calls
- Total MT call minutes

Instead of loading data in tabular manner and deriving above data at retrieval, using a schema less document database (MongoDB) and write pre-structured data in bel is faster and require less computational power at retrieval. Data is written in below structure with unique a key Mobile Station International Subscriber Directory Number (MSISDN). For every processed raw detail record, value appended to the relevant field array.

ObjectID is auto generated by MongoDB and timestamp formatted to ISODate standard.

See Appendix A1 for a real document entry constructed over one hour.

4.1.1 Data Loader Module

Data Loader module is written in C++, due to language's imperative and object-oriented features, and because it is a general purpose language with a rich standard library and contains the important parts including the core language providing all the required building blocks including data types, variable, literals, etc.

Loader module has three significant classes

- Reader Class
- Data Queue Class
- Writer Class

Figure 9 shows a high-level class diagram of the loader module.

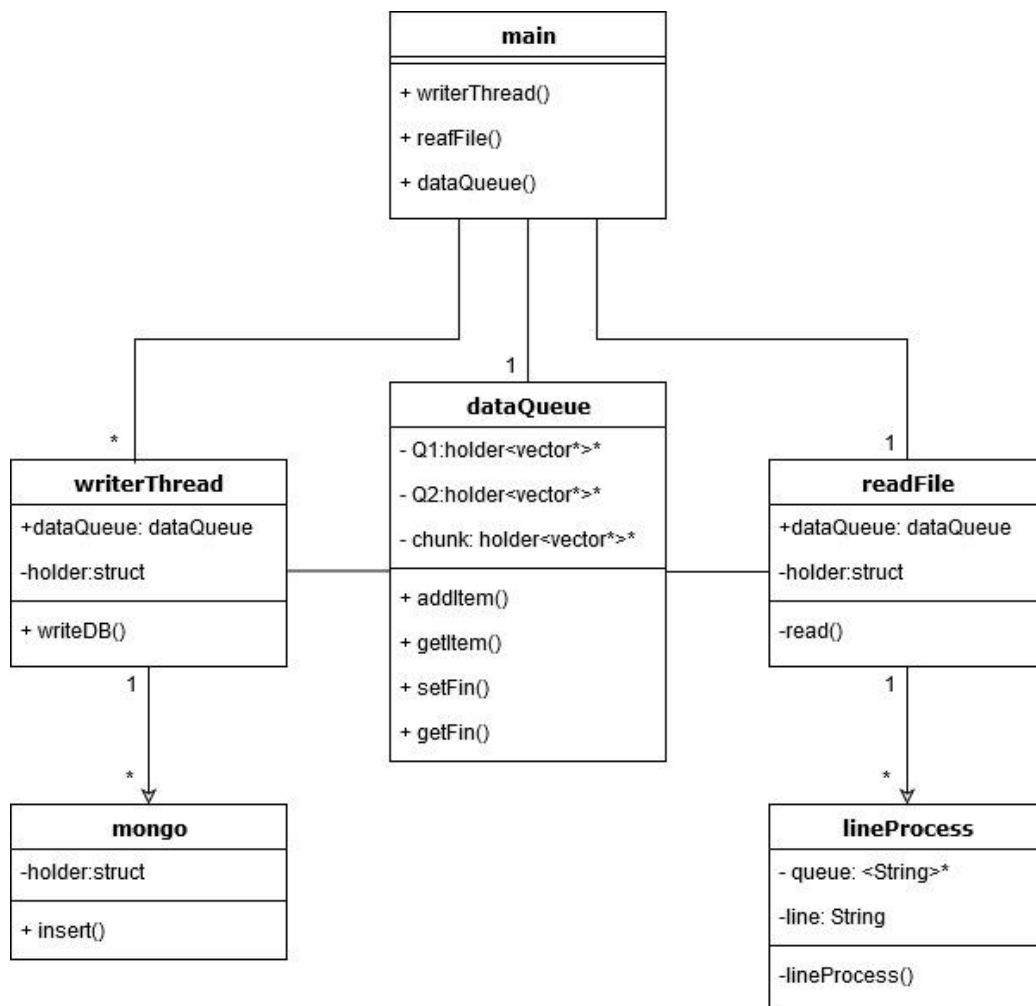


Figure 9 - Loader module high-level diagram

A data queue thread is shared between reader and writer thread. Reader thread watches the directory location where CDR's are collected and when a file is found, data is added to the

queue as chunks. Then writer thread pickups those chunks and insert/update the mongo instance. Object dataQueue holds two C++ queues which automatically switches from read and write modes depending on the read queue data population and read state to synchronously load data.

Since the data is considered in an hourly manner, loader module automatically creates databases with the naming convention *HOURLY_{MONTH}{DAY}*. Inside each collection contains hourly data named *M{HOUR}*. This will be easier for training and predictions as well for historical analysis.

4.2 Training the Neural Network

Considering all the collected records in the dataset is challenging. Mining a database of big data is a laborious task, and requires either advanced parallelized hardware and ML algorithms, or the use of sampling to reduce size of the data to be considered. In addition, since the provided fraudulent subscriber numbers are much fewer than the target subscribers, which results unbalanced dataset proportionality of normal and fraud classes. Typically, classification algorithms will underperform if working with an unbalanced data set. Naturally, results will be biased towards the majority class.

It is agreed that proper sampling is important in ML [34]. Specially in fraud detection problems, fraudulent transaction will always have a very low percentage. Since this makes the data highly unbalanced, an appropriate classification approach needs to be selected.

There're three methods to overcome this problem having its pros and cons [21]

- **Under sampling** – If minority class has sufficient data, majority class data will be randomly removed. There is a possibility that some of the important information will be missed.
- **Oversampling** – Randomly copy samples over and over from minority class to obtain a balanced dataset. One downside is that it may cause data overfitting.
- **Synthetic sampling (SMOTE)** – Data will be synthetically generated for minority class, similar to nearest neighbor.

Decisions about how large of a sample to use must be made rationally. Ideally 15% to 25% is preferred for ANN depending on the problem domain.

To prepare the training set, since network will be blocking the confirmed fraudulent numbers, first obtained the recent confirmed list of fraudulent numbers from ITU division. For normal subscribers, obtained a list of subscribers that are registered in various VAS services which fraudsters dose not usually do.

A separate module is used to create training sample which retrieves hourly summarized data stored in MongoDB instance. Then pre-processes the data and stores in a MySQL table for easy retrieval when training the network. Structure of the table is shown in *Appendix B.2*.

4.2.1 TensorFlow

TensorFlow is a free and open-source software library which is written in C++, Python and CUDA for differentiable and dataflow programming that can be used in variety of applications. In machine learning problems such as classification or prediction, this symbolic math library is commonly used. It is a production grade library, which is used at the technology giant Google for example.

TensorFlow can span across multiple CPU's and GPUs (with optional SYCL and CUDA extensions for general-purpose computing on graphics processing units) leveraging their computing capacity. Its architecture is in such a way, making it easy to deploy across multiple platforms from GPUs, CPUs, TPUs, general purpose hardware, desktop computers, mobile to large computing clusters.

Multidimensional data arrays which neural networks perform operations are called as tensors. Since library expresses its computational results as stateful dataflow graphs, hence the name TensorFlow.

4.2.2 Keras

Keras is a high-level API which runs on top of TensorFlow CNTK and Theano which simplifies the usage of these libraries with predefined functions. It is a fast, user friendly, extensible and modular API. It supports both GPU and CPU deployments.

Keras was developed and is maintained by Francois Chollet and is part of the Tensorflow core, which makes it Tensorflows preferred high-level API. [19]

4.2.3 Model Building

In this stage several models were trained applying varieties of training modes and datasets. Experiments were conducted trying different number of layers, densities, and activation functions.

Python programming language is used for construction, training, and prediction of the neural network. Keras high-level API is used to create the network using TensorFlow as a backend. Numpy arrays and matrixes are used to hold, prepare and format the data which is retrieved from loaded mongo instance. Neural network is initialized with a sequential input layer twelve input nodes experimental hidden layers. Finally, since this is a binary classification, output layer is defined with one perceptron with sigmoid activation function.

Description of the input parameters, and the graphical representation on a sample neural network is shown in Table 2 and Figure 10.

Input Parameters	
1	MO SMS Unique B Numbers
2	MO SMS Unique LACs
3	MT SMS Unique A Numbers
4	MT SMS Unique LACs
5	MO Call Unique B Numbers
6	MO Call Unique LACs
7	MO Call Unique IMEIs
8	Total MO Calls in minutes
9	MT Call Unique A Numbers
10	MT Call Unique LACs
11	MT Call Unique IMEIs
12	Total MT Calls in minutes

Table 2 - Input Parameters

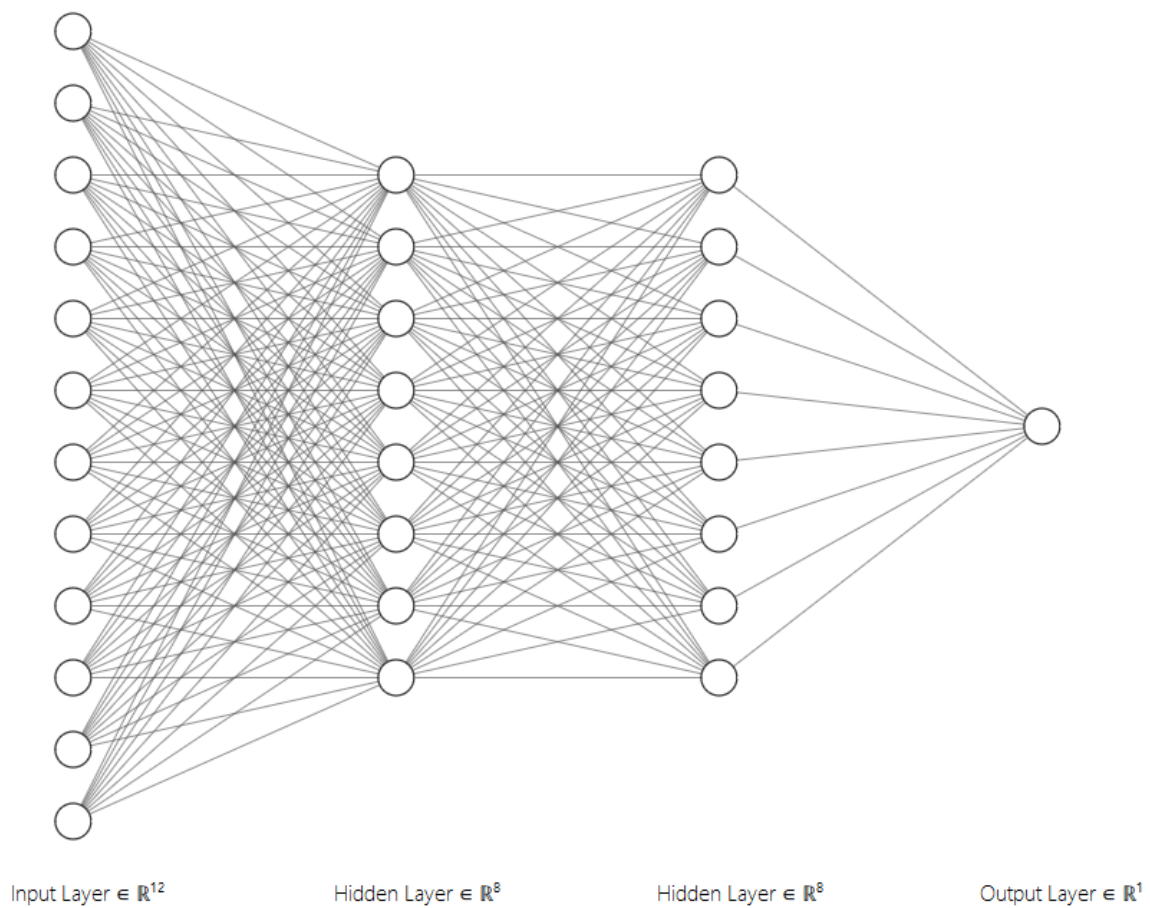


Figure 10 - Visual representation of ANN

Trained model is saved as a file in ".hd5" format which saves the model with weights to be used in the prediction module.

There are many discussions and arguments about determining the number of hidden layers, density and number of nodes that should be used and followed in building a neural network. The universal approximation theorem [21] even states that a feed-forward network, with a single hidden layer which contains a finite number of neurons, can approximate continuous functions with mild assumptions on the activation function. Below three rules of thumbs were used for determining the number of hidden layers and number of neurons in each layer [22].

- The number of hidden neurons should be between the size of the input layer and the size of the output layer.
- The number of hidden neurons should be 2/3 the size of the input layer, plus the size of the output layer.
- The number of hidden neurons should be less than twice the size of the input layer.

4.2.4 Test Data Preparation

To train the network, Mobile ITU division has provided 495 number of confirmed MSISDN's over a period of one month. As for confirmed normal MSISDN's a list of 10,943 MSISDN's were obtained.

These selected MSISDN's were used to filter out and prepare the test dataset. Furthermore, since hourly aggregation is considered, outliers have to be removed from fraudulent transactions as well. According to ITU, the main parameter to be considered is mobile originated number of unique calls. Hence fraudulent transactions are filtered even more to obtain two data sets depending on the number of MO unique calls as the recommendation. Table 3 describes the test data preparation statistics.

		Fraud		Normal	
		Count	%	Count	%
Confirmed Number of MSISDN's		495		10943	
Filtered Hourly Transaction Over One Month		10647	0.86%	1230287	99.14%
Outliers Removed	MO Unique Numbers > 3	490	0.04%	1230287	99.96%
	MO Unique Numbers > 5	236	0.02%	1230287	99.98%

Table 3 - Test Data Preparation

With outliers removed, dataset became more unbalanced (Fraudulent = 0.04% and 0.02%).

Considering the problem domain and data set, three data sets were prepared. One over sampled with slightly altered MO call duration and two under sampled sets. Given the volume of fraudulent transactions, randomly selected 10,000 and 5,000 normal transactions for under sampling sets. Table 4 describes the final data set.

	Fraud		Normal	
	Count	%	Count	%
Under Sampled	490	4.67%	10000	95.33%
	230	4.40%	5000	95.60%
Over Sampled	307571	20.00%	1230287	80.00%

Table 4 - Final Dataset

4.3 Prediction

A separate module using the same API and backend is created which loads the saved trained model and it is executed on hourly basis. This way, the network can be optimized and re-trained over time without affecting normal operation since it's common for fraudsters to change their patterns constantly.

Same as the trainer module, this module also follows the same process to retrieve data from MogoDB instance and passes it through loaded trained model and store only the detected fraudulent activities to a MySQL database. Since the data needs to be kept for a long time for future reference, it is distributed in hundred tables depending on the last two numbers in MSISDN as shown in Figure 11.

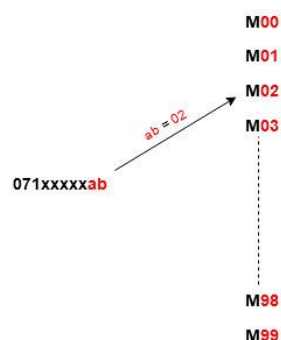


Figure 11 - Divide by MSISDN

At the end of each prediction, a summary of the iteration is inserted into a MySQL "SUMMARY" table. Table definition can be found at *Appendix B.3*.

Figure 12 shows the high-level overall system architecture of the loading process.

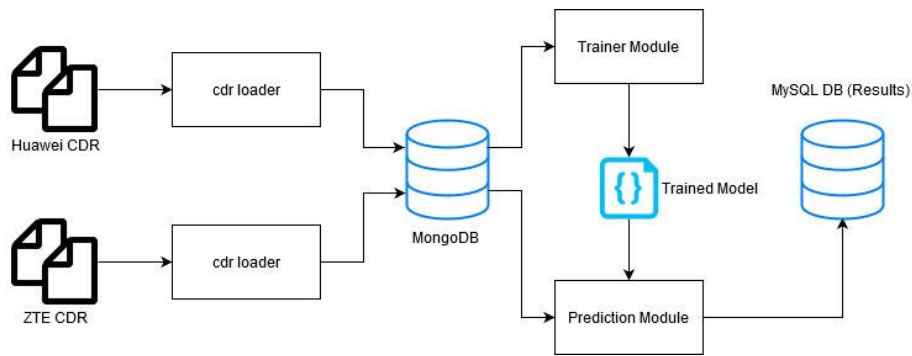


Figure 12 - Overall System Architecture

4.4 UI Design

Even though given the nature of the system, a GUI is not mandatory. Ultimate goal is to detect potential fraudulent numbers and make the necessary decision automatically or generate automated daily or hourly reports depending on ITU division requirement. But for ease of access and system visibility, a simple web interface was developed with below three modules. Front end is developed with React Framework and webservice is written in node express js middleware framework. UI components are re-used [20] components which are under MIT license.

1. Dashboard
2. Number Search
3. Reporting

4.4.1 Dashboard



Figure 13 - Dashboard Screen

Dashboard screen is comprised with two, line charts. Charts are rendered with the data retrieved from "SUMMARY" table and initially it will display data for past twenty-four hours.

4.4.2 Number Search

This page provides an interface to search for a mobile number and will return hourly results if fraudulent was detected.

Date	Hour	Mobile Originated						Mobile Terminated
		SMS_B_NUMBER_UNIQUE	SMS_IC_UNIQUE	CALL_B_NUMBER_UNIQUE	CALL_IC_UNIQUE	CALL_IMEI_UNIQUE	CALL_DURATION	SMS_A_NUMBER_U
2020-04-01	21	0	0	24	3	1	1273	0
2020-04-02	19	0	0	25	3	1	1013	0

Figure 14 - Number Search Screen

4.4.3 Reporting

Two types of reports can be generated against fraud data. Reports are downloaded as text files.

1. Unique Numbers specifying a date range
2. Unique Numbers at specific date and hour

Generate Report - Date Range

Date Range

From: May 7, 2020 To: May 7, 2020 Go

Generate Report - Hourly

Choose a day

YYYY-MM-DD Select hour Go

Figure 15 - Reporting Screen

Chapter 5. Evaluation

Aim of this project is to build a system to predict SIM Box frauds on an hourly basis. To achieve this, we need to ensure that we are using the network model that best detects SIM Box frauds. Models were built using the rules of thumb described in section 4.2.3. This section presents evaluation of results collected and recorded for evaluation.

Although several models were tested, for evaluation below model were selected. Figure 16 the number of layers and number of nodes in each layer.

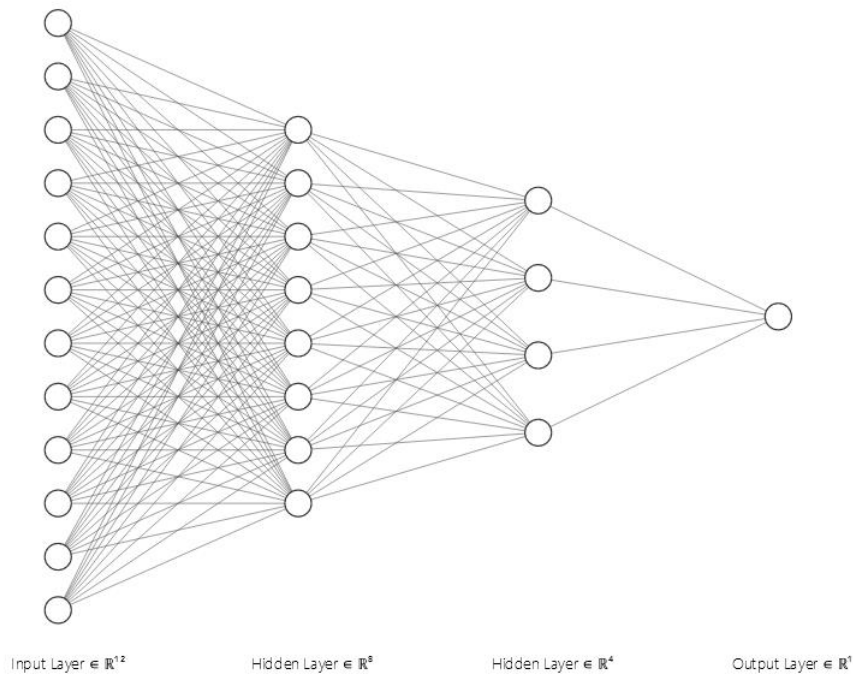


Figure 16 - Selected Network Model

As described in section 4.2.4, three datasets were considered in training the network and the accuracy they have shown is in Table 5.

	Accuracy
SET1 - MO Unique > 5 (Under Sampled)	0.9928
SET2 - MO Unique > 3 (Under Sampled)	0.9847
SET3 - Over Sampled	0.9959

Table 5 – Training data sets and Accuracy

5.1 Under Sampled

Detailed Results of two under sampled Tests

	precision	recall	f1-score	support
0	0.99	0.99	0.99	10000
1	0.81	0.87	0.84	490
accuracy		0.98	10490	
macro avg	0.9	0.93	0.92	10490
weighted avg	0.99	0.98	0.98	10490

Table 6 - Under Sampled (MO Calls > 3) – SET1

	precision	recall	f1-score	support
0	1	1	1	5000
1	0.9	0.95	0.93	236
accuracy	0.99	5236		
macro avg	0.95	0.97	0.96	5236
weighted avg	0.99	0.99	0.99	5236

Table 7 - Under Sampled (MO Calls > 5) – SET2

Confusion Matrix

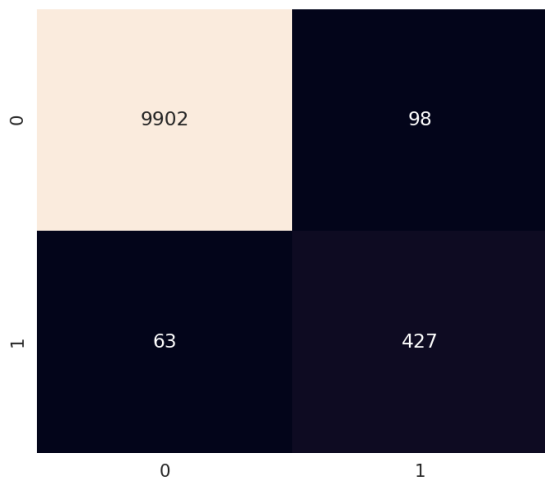


Figure 17 - Confusion Matrix (MO Unique > 3) – SET1

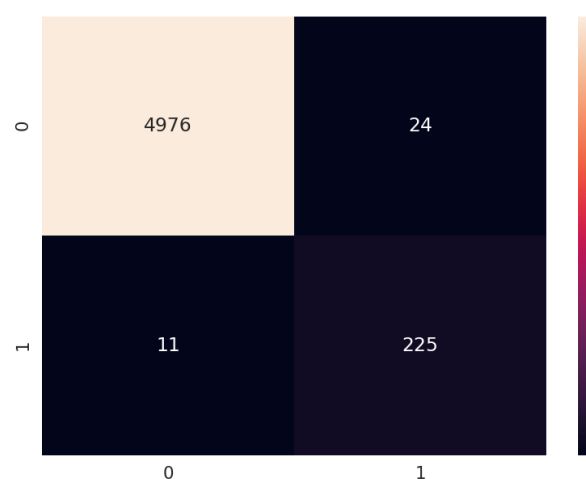


Figure 18 - Confusion Matrix (MO Unique > 5) – SET2

Receiver Operating Characteristics

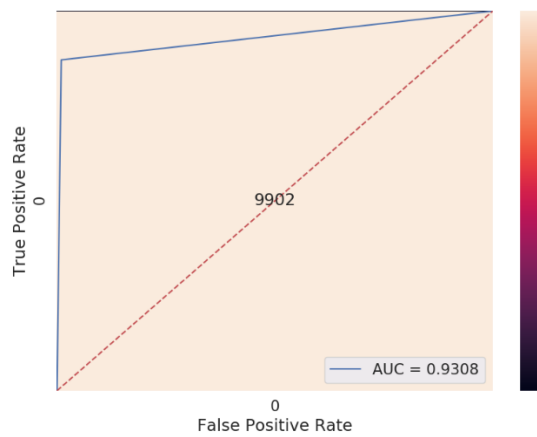


Figure 19 - ROC (MO Unique Calls > 3) – SET1

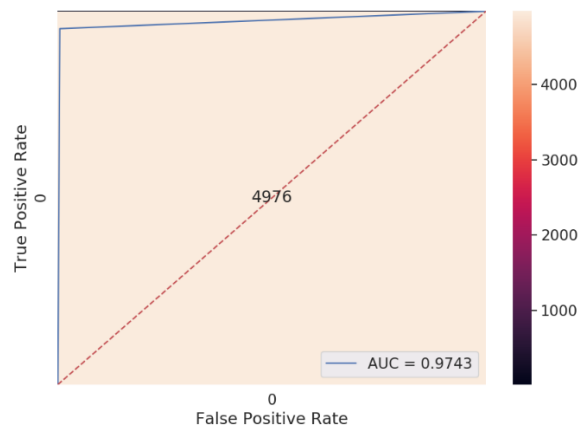


Figure 20 - ROC (MO Unique Calls > 5) – SET2

5.2 Over Sampled

Detailed Results

	precision	recall	f1-score	support
0	1	1	1	1230287
1	0.99	1	0.99	307571
accuracy	1	1537858		
macro avg	0.99	1	1	1537858
weighted avg	1	1	1	1537858

Table 8 - Over Sampled Results – SET3

Confusion Matrix

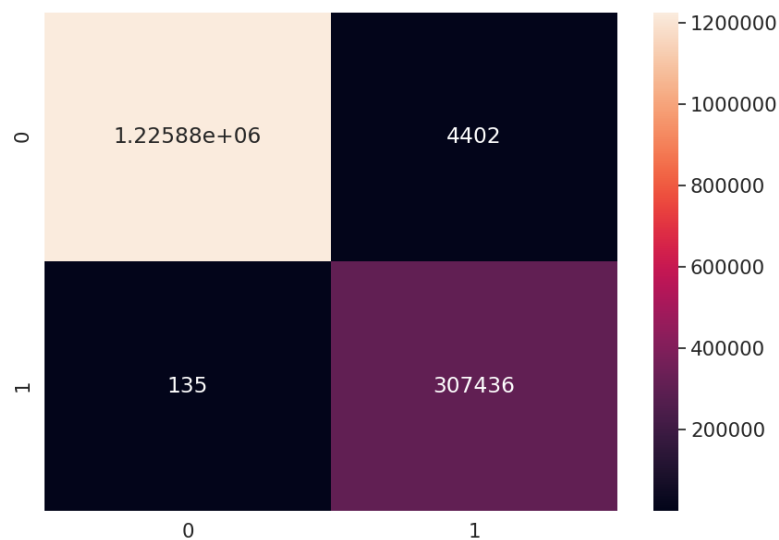


Figure 21 - Confusion Matrix (Over Sampled) – SET3

Receiver Operating Characteristics

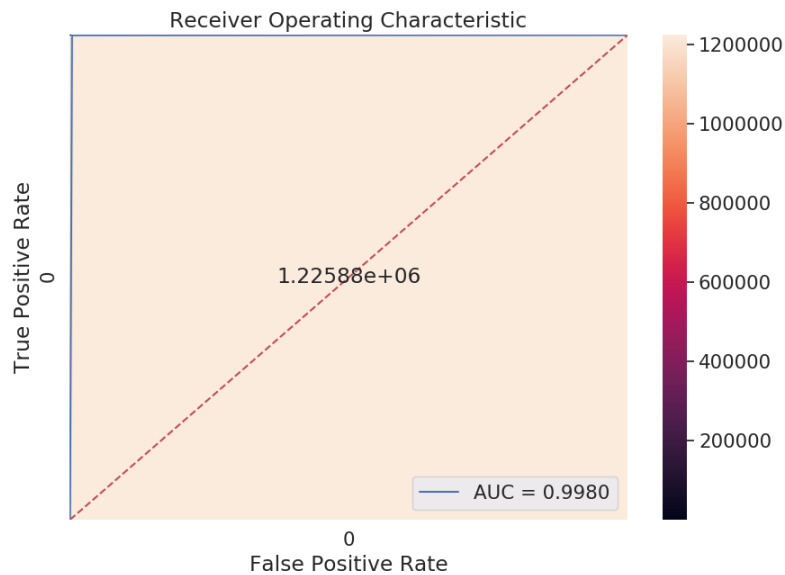


Figure 22 - ROC (Over Sampled) – SET3

5.3 Comparison

To compare ML models and how they performed, is a good method to compare Sensitivity, Specificity and AUC. Below table compares all calculated matrixes of all three attempts.

	Sensitivity	Specificity	AUC
SET1	0.993677873	0.813333333	0.9308
SET2	0.997794265	0.903614458	0.9743
SET3	0.999889887	0.985883696	0.998

Table 9 - Model Comparison

The results in Table 9 - Model Comparison depicts that all the models have comparable performance measures. However, the model trained with SET3 has better results than the other two in all the areas.

With respect to the domain of our problem, we are more interested in “sensitivity”. In other words, it tells us, what percentage subscribers were correctly identified as frauds.

Chapter 6. Conclusion

Telecommunication operators tend to impose higher tariffs for international calls in developing countries. However, this scenario is abused by SIM-Box fraudsters delivering less expensive price to callers and divert the revenue from operators. The VoIP technology with SIM-Box and local SIM cards supports them in diverting the international call and deliver them back as a local call. A mechanism which support to detect SIM-Box frauds early, and hinder the fraudsters making business helps the operators to minimize their loss of revenue.

In this project, we have analyzed MSC CDR's and identified 12 features that can be extracted and used in identifying fraudulent transactions. Data is streamed into our server in near real time manner. However, accumulating this big data for long period of time for analysis is challenging. Moreover, it is not feasible to do analytics on such a data set in standard hardware. Hence in this project, hourly aggregated data sets were prepared as pre-processed data and set for analytics module. We also prepared three training data sets with under sampling and over sampling methods due to a highly imbalanced data set.

Model trained with over sampling showed better performance with accuracy of 99.59%, whilst other two under sampled models showed slightly less accuracy, which are 98.47% and 99.28%. On the other hand, we have also calculated other sensitivity and AUC for evaluating the models. In all three sets sensitivity (0.99988 > 0.997794, 0.993677) and AUC (0.998 > 0.9743, 0.9308) showed higher values for SET3. So, from all the observations, we can determine that SET3, which is the over sampled set performed better than other two under sampled sets.

This result concludes that, model built with over sampled dataset is more appropriate to use in classification model for SIM-Box fraud detection. This will allow the service provider to detect SIM Box frauds on hourly slots and block them accordingly.

6.1 Future Work

Even though we use a machine learning approach to solve the classifier problem, as we discussed in early chapters, fraudsters do evolve rapidly. Hence the model must be trained with confirmed data sets frequently. To achieve that, we can add a new module in web interface to upload confirmed numbers and spawn a learning job allowing user to set parameters such as date range etc. Service provider should define a procedure to obtain confirmed fraudulent numbers from other providers and a responsible person/ team to train the network with new data set and test its outcome.

Although current system considers only CDR based parameters, it can be developed to take customer profiling information into consideration for decision making process. System should maintain a separate profile for each in-net customer so this information can be weighted in

case of a previous fraud attempts from the same customer and use as an input for learning and predicting.

Since system performance heavily depend on compute for CDR loading, prediction and MongoDB operations. A clustered architecture is well suited for this project. Database tier can be clustered since MongoDB natively supports clustering. Compute operations can be decentralized with multiple nodes since project already has independent modular architecture.

Chapter 7. Bibliography

- [1] M. Z. R. P. J. a. A. P. I. Murynets, "Analysis and detection of simbox fraud in mobility networks," *INFOCOM 2014 Proceedings IEEE*, p. 1519–1526, 2014.
- [2] S. I. A. H. E. a. R. Sallehuddin, "Classification of sim box fraud detection using support vector machine and artificial neural network," *International Journal of Innovative Computing vol. 4, no. 2*, 2014.
- [3] O. A. Abidogun, "Data mining, fraud detection and mobile telecommunications: Call pattern analysis with unsupervised neural networks," 2005.
- [4] R. K. G. a. S. K. Meher, "A rule-based approach for anomaly detection in subscriber usage pattern," *Proceedings of World Academy of Science, Engineering and Technology*, pp. 396-399, 2007.
- [5] M. R. AlBougha, "Comparing data mining classification algorithms in detection of simbox fraud," the Repository at St. Cloud State, 2016.
- [6] M. Z. R. P. J. a. A. P. I. Murynets, "Analysis and detection of simbox fraud in mobility networks," *INFOCOM Proceedings IEEE, IEEE*, pp. 1519-1526, 2014.
- [7] E. S. A. B. H. C. a. P. T. B. Reaves, "Boxed out: Blocking cellular interconnect bypass fraud at the network edge.," in *USENIX Security Symposium*, 2015.
- [8] E. F. M. H. a. C. P. I. Witten, *Data Mining: Practical machine learning tools and techniques.*, Morgan Kaufmann, 2016.
- [9] M. B. K. a. E. B. M. B. M. Mohammed, *Machine learning: algorithms and applications*, CRC Press, 2016.
- [10] N. C. a. J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*, Cambridge university press, 2000.
- [11] G. Williams, "Data mining with Rattle and R: The art of excavating data for knowledge discovery," *Springer Science & Business Media*, vol. 2011.
- [12] S. D. a. X. Du, *Data mining and machine learning in cybersecurity*, 2016: CRC press.
- [13] P. Gaur, "Neural networks in data mining," *International Journal of Electronics and Computer Science Engineering*, vol. 1, no. 3, 2013.
- [14] J. H. a. J. P. M. Kamber, *Data mining: Concepts and techniques*, 2012.

- [15] S. SHARMA, "Networks," [Online]. Available: <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>.
- [16] J. Han and C. Morag, "The influence of the sigmoid function parameters on the speed of backpropagation learning," *Natural to Artificial Neural Computation. Lecture Notes in Computer Science*, pp. 195-201, 1995.
- [17] D. Liu, "A Practical Guide to ReLU," [Online]. Available: <https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7>.
- [18] C. Z. a. Q. Y. S. Zhang, "Data preparation for data mining," *Applied artificial intelligence*, vol. 17, no. 5-6, p. 375–381, 2003.
- [19] S. Chatterjee, "Deep learning unbalanced training data? Solve it like this.," 27 5 2018. [Online]. Available: <https://towardsdatascience.com/deep-learning-unbalanced-training-data-solve-it-like-this-6c528e9efea6>.
- [20] G. Tanner. [Online]. Available: <https://towardsdatascience.com/introduction-to-deep-learning-with-keras-17c09e4f0eb2>.
- [21] W. Chen, *The Electrical Engineering Handbook*, Academic Press , 2004.
- [22] P. Jeff Heaton, "The Number of Hidden Layers," 01 06 2017. [Online]. Available: <https://www.heatonresearch.com/2017/06/01/hidden-layers.html>.
- [23] @CreativeTim, "black-dashboard," [Online]. Available: <https://www.creative-tim.com/product/black-dashboard>.
- [24] I. I. a. H. Mohamed, arXiv preprint arXiv:1711.04627, 2017.
- [25] K. V. a. P. K. Singh, "An insight to soft computing based defect prediction techniques," *International Journal of Modern Education and Computer Science*, vol. 7, no. 9, p. 52, 2015.

Chapter 8. Appendices

Appendix A Data Examples

Appendix A.1 Real document entry constructed over one hour

```
{
  "_id" : ObjectId("5e5967708a40920d050d0556"),
  "MSISDN" : "0712899106",
  "DATETIME" : ISODate("2020-02-28T13:55:35.000Z"),
  "IMSI" : "413010535925076",
  "MT_CALL_A_NUMBER" : [
    "0756371677",
    "0776186947",
    "0754746818",
    "0775127211"
  ],
  "MT_CALL_A_NUMBER_UNIQUE" : [
    "0756371677",
    "0776186947",
    "0754746818",
    "0775127211"
  ],
  "MT_CALL_DURATION" : [
    "12",
    "9",
    "35",
    "25"
  ],
  "MT_CALL_LC_UNIQUE" : [
    "6110017198",
    "6110043763",
    "6110023078",
    "6110023166"
  ],
  "MT_SMS_A_NUMBER" : [
    "0712755777",
    "0712755777",
    "0712755777"
  ],
  "MT_SMS_A_NUMBER_UNIQUE" : [
    "0712755777"
  ],
  "MT_SMS_LC_UNIQUE" : [
    "6110023162",
    "6110023167"
  ],
  "MO_CALL_B_NUMBER" : [
    "0779736269",
```

```

        "0775127211",
        "0775722456",
        "0775127211",
        "0754746818",
        "0715229865",
        "0771655693",
        "0776186947"
    ],
    "MO_CALL_B_NUMBER_UNIQUE" : [
        "0779736269",
        "0775127211",
        "0775722456",
        "0754746818",
        "0715229865",
        "0771655693",
        "0776186947"
    ],
    "MO_CALL_DURATION" : [
        "57",
        "25",
        "33",
        "8",
        "123",
        "13",
        "17",
        "45"
    ],
    "MO_CALL_IMEI_UNIQUE" : [
        "352880108299730"
    ],
    "MO_CALL_LC_UNIQUE" : [
        "6110017977",
        "6110023162",
        "6110023077",
        "6110023073",
        "6110023167",
        "6110023072"
    ]
}

```

Appendix A.2 ROW CDR

```

7|0711812720|0850380860390840200690|0711812720||20181111|000047||947700
0003|947100180|63430|171|869649021128160|413012688050358|||||1246629422
67|||POS|HW|||||41301|SMS|CS

```

Appendix B Data Structures

Appendix B.1 MongoDB Document Structure

```
{
  "_id" : ObjectId(),
  "MSISDN" : "",
  "DATETIME" : ISODate(),
  "IMSI" : "",
  "MT_CALL_A_NUMBER" : [ ],
  "MT_CALL_A_NUMBER_UNIQUE" : [ ],
  "MT_CALL_DURATION" : [ ],
  "MT_CALL_LC_UNIQUE" : [ ],
  "MT_SMS_A_NUMBER" : [ ],
  "MT_SMS_A_NUMBER_UNIQUE" : [ ],
  "MT_SMS_LC_UNIQUE" : [ ],
  "MO_CALL_B_NUMBER" : [ ],
  "MO_CALL_B_NUMBER_UNIQUE" : [ ],
  "MO_CALL_DURATION" : [ ],
  "MO_CALL_IMEI_UNIQUE" : [ ],
  "MO_CALL_LC_UNIQUE" : [ ],
  "MO_SMS_B_NUMBER" : [ ],
  "MO_SMS_B_NUMBER_UNIQUE" : [ ],
  "MO_SMS_IMEI_UNIQUE" : [ ],
  "MO_SMS_LC_UNIQUE" : [ ],
  "MT_SMS_A_NUMBER" : [ ],
  "MT_SMS_A_NUMBER_UNIQUE" : [ ],
  "MT_SMS_LC_UNIQUE" : [ ]
}
```

Appendix B.2 Learning Set Table Structure

```
CREATE TABLE `DATA_04` (  
  `MSISDN` int(11) NOT NULL,  
  `DAY` date NOT NULL,  
  `HOUR` int(11) NOT NULL,  
  `MO_SMS_B_NUMBER_UNIQUE` int(11) DEFAULT NULL,  
  `MO_SMS_LC_UNIQUE` int(11) DEFAULT NULL,  
  `MT_SMS_A_NUMBER_UNIQUE` int(11) DEFAULT NULL,  
  `MT_SMS_LC_UNIQUE` int(11) DEFAULT NULL,  
  `MO_CALL_B_NUMBER_UNIQUE` int(11) DEFAULT NULL,  
  `MO_CALL_LC_UNIQUE` int(11) DEFAULT NULL,  
  `MO_CALL_IMEI_UNIQUE` int(11) DEFAULT NULL,  
  `MO_CALL_DURATION` int(11) DEFAULT NULL,  
  `MT_CALL_A_NUMBER_UNIQUE` int(11) DEFAULT NULL,  
  `MT_CALL_LC_UNIQUE` int(11) DEFAULT NULL,  
  `MT_CALL_IMEI_UNIQUE` int(11) DEFAULT NULL,  
  `MT_CALL_DURATION` int(11) DEFAULT NULL,  
  `FRAUD` int(11) DEFAULT NULL,  
  PRIMARY KEY (`MSISDN`, `DAY`, `HOUR`) USING BTREE  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Appendix B.3 Hourly Summary Table

```
CREATE TABLE `SUMMARY` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `DATE` date NOT NULL,  
  `HOUR` int(11) NOT NULL,  
  `FRAUD` int(11) DEFAULT NULL,  
  `TOTAL` int(11) DEFAULT NULL,  
  PRIMARY KEY (`id`) USING BTREE  
) ENGINE=InnoDB AUTO_INCREMENT=48 DEFAULT CHARSET=latin1;
```