Near Real-time SIM-box Fraud Detection in Telecommunication System Using Machine Learning Approach in the Case of Ethio Telecom.

A Thesis Presented

by

Misrak Birhanu

to

The Faculty of Informatics

of

St. Mary's University

In Partial Fulfillment of the Requirements
for the Degree of Master of Science

in

Computer Science

February 2024

ACCEPTANCE

Near Real-time SIM-box Fraud Detection in Telecommunication System Using Machine Learning Approach in the Case of Ethio Telecom.

By

Misrak Birhanu Nigatu

Accepted by the Faculty of Informatics, St. Mary's University, in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

Thesis Examination Committee:

____

Internal Examiner

____

External Examiner

____

Dean, Faculty of Informatics

February 2024

DECLARATION

I, the undersigned, declare that this thesis work is my original work, has not been presented for a degree in this or any other universities, and all sources of materials used for the thesis work have been duly acknowledged.

Misrak Birhanu Nigatu

Signature

Addis Ababa

Ethiopia

This thesis has been submitted for examination with my approval as advisor.

Jabesa Daba (Ass. Prof)

Signature

Addis Ababa

Ethiopia

February 2024

Acknowledgment

First and foremost, I want to express my gratitude to the Almighty, Jesus Christ the son of the Virgin Mary, for providing me with the chance and guidance to achieve my goals and be successful in various aspects of my life.

Second, I would like to express my deepest gratitude to Jabesa Daba (Ass. Prof) for his encouragement and constructive feedback throughout the development of this thesis work.

In third place, I would like to thank the Ethio Telecom Data Analytics and Cyber Security Fraud Management team for providing me with the required data for this thesis work.

A special thank you to Ato Samuel Fantaye of the St. Mary's University student support office for his assistance, encouragement, and helpful information sharing.

Finally, I would like to thank my family, my wonderful husband Zenagebrel Muluneh, and my beloved children Nathan Zenagebrel and Anna Zenagebrel, for their patience, support, and encouragement.

Contents

List of Figures

List of Tables

Acronyms

AAA	Authentication, Authorization, and Accounting
ANN	Artificial neural network
ASP	Active Server Page
AUC	Authentication center
BSC	Base Station Controller
BTS	Base transceiver station
CDMA	Code-Division Multiple Access
CDR	Call Detail Record
CEO	Chief Executive Officer
CRM	Customer Relation Management
EIR	Equipment Identity Register
FMS	Fraud Management System
GSM	Global System for Mobile Communication
HLR	Home Location Register
IRSF	International Revenue Share Fraud
LTE	Long Term Evolution
MSC	Mobile Switching Center
PBX	Private Branch Exchange
RF	Random Forest
SIM	Subscriber Identity Module
SMS	Short Message Service
SQL	Structural Query Language
SVM	Support Vector Machine
UMTS	Universal Mobile Telecommunications System
VLR	Visitor Location Register

Abstract

Telecommunication operators play a vital role in connecting individuals and businesses worldwide, facilitating seamless communication and adopting global connectivity. However, the telecommunications industry is vulnerable to various malicious activities and challenges by fraudsters seeking to exploit weaknesses in the system. One such form of fraud that has emerged as a significant challenge for telecom operators is SIM-box fraud.

This thesis work is targeted to develop a model that helps to detect SIM-box fraudulent subscribers in a near real-time manner. To achieve this, we have set up API integration with Ethio telecom CRM and CBS environment to retrieve call detail record flat files (textual DB) on an hourly basis. Then, we developed a function using ASP.net C# that enables us to preprocess the textual raw data and store it in a database that has been configured using an SQL server to store call detail records, voice, SMS, and Data tuples.

SQL view has been created that joins the CDR, Voice, SMS, and Data tables to combine all the required attributes in one place to facilitate further data analysis. Next, we aggregated different tuples using SQL query and created a C# function that can derive additional attributes that help to track the behaviors of available calls. Once, we analyzed and compiled data of call detail records that incorporate the Voice, SMS, and Data utilization of each subscriber, we split the dataset into 1_hour, 1_day, and 7_day datasets and fed them into selected machine learning algorithms.

Finally, we experimented by feeding the preprocessed, aggregated, and analyzed dataset to machine learning algorithms of Random Forest (RF), Support Vector Machine (SVM), and Neural Network (NN) algorithms using sci-kit-learn (sklearn) python library and 100% accuracy has recorded in RF and NN algorithms in all 1_Hour, 1_Day and 7_Day datasets. Hence, we have concluded that with a good CDR analysis engine or module, RF, and NN can effectively identify possible fraudulent subscribers.

Keyword: *Telecommunications fraud, SIM box, international call bypass, Machine learning , Call Detail Record, Classification, Voice call termination.*

Chapter One

Introduction

The current telecommunications industry is experiencing rapid technological developments in the process of transforming the way society interconnects and conducts business. The demand for seamless connectivity has increased from time to time and a robust cyber security mechanism, tools, and procedures are critically required to protect network infrastructures from various forms of telecom fraud. Among the commonly available frauds that target telecommunication service providers, SIM-box fraud is the major one and its detection and prevention has emerged as a significant concern [1].

SIM-box fraud involves an illegal practice of bypassing genuine telecommunication networks using SIM-box devices that are capable of routing international calls over the internet and bypassing legitimate infrastructure. This fraudulent activity not only leads to revenue losses for service providers but also brings substantial security risks and damages the integrity of the telecommunication ecosystem [2][3]. To address this increasing problem, it becomes necessary to develop effective methods for near real-time SIM-box fraud detection. Near real-time detection allows for immediate action, minimizing financial losses and preserving the network's integrity.

In recent years, the advancements in machine learning approaches teamed with the availability of large-scale datasets, have presented new opportunities for detecting and preventing fraud in telecommunication networks. It is now possible to identify unusual behaviors linked to SIM-box fraud and derive valuable insights from large amounts of network data by utilizing statistical analysis, machine learning algorithms, and pattern recognition techniques.

The primary objective of this thesis is to develop a more accurate and efficient SIM-box fraud detection model that can operate in near real-time, enabling prompt responses to potential fraud incidents by harnessing the power of data analytics and machine learning. To achieve this objective, various data sources commonly found in telecommunication networks (call detail records, network traffic data, and subscriber information) shall explored and collected from Ethio Telecom through a formal support letter from St. Mary's University and approval of Ethio telecom CEO, Frehiwot Tamru.

To deal with various telecom frauds, different FMS (Fraud Management System) solutions use advanced machine learning techniques to analyze the available CDR datasets and uncover patterns and anomalies associated with SIM-box fraud. In addition, the possibility of incorporating near real-time data streams and automated monitoring techniques shall be investigated to enhance the responsiveness of the fraud detection system [2].

By addressing the challenge of SIM box fraud through a machine-learning approach, this research contributes to the overall security and stability of telecommunication networks. The outcomes of this study have the potential to assist telecommunication service providers in proactively identifying and preventing SIM-box fraud, safeguarding their revenue streams, and protecting the interests of legitimate users [3].

In the subsequent chapters, the literature review, related work, proposed solution including methodology, data analysis techniques, experimental setup, and results obtained from the investigation will be discussed. The implications of the findings will be examined, and recommendations for further enhancements and future research in the field of SIM-box fraud detection will be proposed.

Overall, this thesis aims to provide valuable insights and practical solutions for the near real-time detection of SIM-box fraud in telecommunication networks, contributing to the establishment of a more secure and trustworthy communication environment for both service providers and subscribers.

1.1 Background

The telecommunications industry is a crucial facilitator of smooth communication and global connectivity through meeting the needs of individuals, businesses, and organizations. However, as the need for trustworthy and affordable telecom services grows, service providers have a lot of challenges, one of which is combating fraud. Among the several forms of fraud that pose a significant risk to communications networks, SIM-box fraud stands out [3][4].

SIM-box fraud, also known as interconnect bypass fraud, is the practice of routing international calls via the Internet using SIM-boxes rather than through genuine telecom network infrastructure. These SIM boxes are devices that have several SIM cards installed and may be operated from a distance to simulate authentic network activity. Fraudsters profit from this fraudulent practice by taking advantage of the differences in cost between local and international call rates. [2].

The effects of SIM-box fraud extend beyond financial losses for telecommunication service providers. Revenue diversion from legitimate network routes due to SIM-box fraud leads to large monetary losses, impacting profitability and impeding investments in network infrastructure and service enhancements. Furthermore, SIM-box fraud compromises the integrity of telecommunication networks, resulting in faded call quality, dropped calls, increased latency, and an overall degraded customer experience. In addition, the activities of SIM-box frauds can expose sensitive customer information to unauthorized access, compromising privacy and enabling further fraudulent activities [4].

To effectively prevent SIM-box fraud, the telecommunications sector needs a robust and efficient fraud detection technology that can operate in almost real-time. The conventional approach to fraud detection relies on manual analysis and rule-based programmed systems, which are not up to date with the ever-changing fraud schemes and the increasing volume and complexity of CDR data. Therefore, it is imperative to employ advanced analytics and machine learning approaches in order to promptly detect SIM-box fraud [3][4].

Data-driven methods have a great deal of promise for almost instantaneous SIM-box fraud detection through the inspection of large-scale datasets that include subscriber data, network traffic statistics, and call detail records, patterns, abnormalities, and behavioral traits linked to fraudulent SIM-box activities can be found. In addition, machine learning algorithms can learn from past data, spot unnoticed relationships, make precise forecasts, and help create efficient fraud detection models.

1.2 Motivation

After ten years in the telecom sector, six of those years were spent specializing in cybersecurity, where the author of this thesis saw widespread fraudulent activity that seriously harmed the telecom sector in various aspects of revenue loss, service quality degradation, trust issues, and regulatory in compliance. Hence, as a member of the Ethic telecom cyber security team that combats different fraudulent activities daily, the situation is highly motivated the author to do thesis research on the solution proposal of telecom fraud, particularly the SIM-box fraud detection mechanism.

SIM-box fraud exposes telecom providers to a large financial risk and results in a considerable loss of revenue. Operators may reduce monetary losses, safeguard their profitability, and more efficiently deploy resources for network improvements and customer satisfaction by identifying fraud in almost real-time. Furthermore, it directly affects the standard of service offered to clients who are legitimate. Call drops, network congestion, and poor call quality are all caused by fraudulent activity. Through prompt detection and mitigation of fraudulent activities, operators can guarantee continuous and superior services, thereby augmenting client contentment and allegiance. [2].

From the perspective of trust, SIM-box fraud damages the reputation and trust of telecommunication operators. Customers rely on operators to deliver reliable and secure communication services. By actively detecting and preventing fraud, operators can safeguard their reputation, build trust with customers, and establish themselves as reliable service providers. Therefore, the amalgamation of all the mentioned issues will justify the motivation behind this thesis work.

1.3 Statement of the problem

In the current Ethiopian market, two telecom operators namely Ethio-telecom and Safaricom are providing telecom services to service consumers; though the magnitude of the SIM-box fraud affects both telecom providers, this thesis work is written based on the context of Ethio telecom network infrastructure and data. Based on the recent (2022/2023) Ethio telecom annual report the customer base is 72M with 9,078 mobile sites deployment, 24,552 km backbone fiber and 8800 km metro fiber installation including both urban and rural areas, with revenue of 75.8 billion Birr [5]. From the earned revenue local voice and internet data takes the lion's share and the revenue from the international call requires further attention due to the SIM-box fraudulent operators gaining an unfair advantage by offering lower call rates, and Ethio telecom is losing a considerable amount of foreign currency.

To deal with this problem, Ethio Telecom had deployed various fraud management solutions that consume various data, analyze, and report possible fraudulent numbers. Unfortunately, the problem is persisting, and an enormous number of calls are routed from the international link to SIM-box with different fraudulent software and this requires further research and development effort to identify updated fraud techniques, uncover call patterns, develop a detection engine, and generate fraudulent reports accordingly.

This thesis work will follow a data-driven approach, that involves collecting, analyzing, and interpreting data to inform and guide actions accordingly. Hence, data shall be collected from the Ethio-telecom CRM (Customer Relation Management) database and run the data-driven approach methodologies to bring feasible model artifacts that are tailored with Ethio-telecom infrastructure towards contributing to the revenue generation, service quality, reputation building, and regulatory compliance of Ethio telecom.

Hence, the research question for this thesis work is:

- What are the key features that can effectively distinguish between legitimate and fraudulent SIM-box traffic?

- Which machine learning method can effectively use provided datasets to predict the behavioral patterns of SIM-Box fraud?

1.4 Objectives

1.4.1 General Objective

The general objective of this study is to develop near real-time SIM-box fraud detection using a data-driven approach.

1.4.2 Specific Objectives

- To review and evaluate the advanced telecom fraud detection techniques.
- To gather CDR, and billing data from various telecommunication data sources.
- To preprocess the data to ensure quality and compatibility for subsequent analysis.
- To extract meaningful features from the collected data, combining network-level and subscriber-level attributes to create a comprehensive fraud detection model.
- To evaluate the performance of the proposed model and communicate the acquired knowledge.

1.5 Scope and Limitation

The scope of this thesis work is to develop a near real-time SIM-box fraud detection model using data-driven approach that leverages advanced analytics and machine learning algorithms, to analyze network data such as call detail records, network traffic data, and subscriber information with the aim of identify patterns, anomalies, and indicators of SIM-box fraud to enhance fraud detection capabilities.

The limitation of this work is that the sense of near real-time shall support starting from the last one-hour data because in the available telecom infrastructure CDR contains massive records and is not supported to streamline data in less than one hour. In addition, the performance of the work is considerably affected by the computational resource of processing power and storage capacity of the server, and query optimization will not be covered in this thesis targeting to resolve the computational overhead.

1.6 Significant of the study

The study's contributions to preventing financial losses, preserving network integrity, improving fraud detection skills, and fostering a safe and reliable communication environment will be highlighted in the relevance section. Additionally, it will serve as a foundation for our further work on the related topic.

1.7 Organization of the Thesis

The thesis will be organized into five chapters, including an introduction, literature review, related work, methodology, data analysis/results, recommendation, and conclusion. Each chapter will focus on specific aspects of the research, contributing to the overall understanding and evaluation of the proposed machine-learning approach for SIM-box fraud detection.

Chapter Two

Literature Review

This chapter presents the major concepts of mobile wireless communication network and its architectural structure, vulnerabilities to various fraud, and available fraud detection methodologies. Also, a review has been made on fundamentals of data-driven approach involving methods and techniques regarding data collection, processing, analysis, training, modeling, and visualizing, that is crucial element to succeed the objective of the thesis.

2.1 Mobile Wireless Communication Overview

Modern communication system research, infrastructure, and expenditure heavily revolve around mobile wireless networks. The constant consumer desire for enhanced system performance, encompassing better call quality, broader network coverage, and higher data rates, propels investments from manufacturers, operators, and users alike. At the core of system performance lies the meticulous design of mobile wireless networks that adhere to specific availability objectives [6].

Various mobile wireless network technologies and standards are in use globally, and they generally share similar network topologies [6]. The primary distinctions among these technologies are found in their air interfaces. Consistency in availability and reliability is a common thread across all mobile wireless networks. Upcoming advancements like long-term evolution (LTE) build upon existing technologies, making the analytical methods outlined here applicable with minor adjustments. The well-established foundation of current technologies facilitates a comprehensive assessment of availability, reliability, and operational implications within extensive network deployment [6][7].

While differences in design and implementation exist among various mobile wireless technologies (like GSM, CDMA, UMTS, LTE, and others), the foundational components and design principles persist consistently across all these technologies. Moreover, vendor interpretations of wireless standards introduce a degree of variations in deployments. Acknowledging these variations, the ultimate goal is to provide insights and instructions for analyzing and designing reliable, robust and secured mobile wireless networks [5].

2.2 Mobile Wireless Communication Components

Mobile wireless networks utilize a complex interconnected network of component that performs the functions necessary to provide wireless voice and data services to subscribers. The network elements manage call switching, facilitate the flow of data traffic, establish end-to-end call connections, and oversee overall network management.

Figure 2.1 shows a block diagram for a global system for mobile communications (GSM) wireless cellular network. Although the figure is specific to GSM networks, the basic network elements identified are common to GSM, CDMA, UMTS, and other network designs.

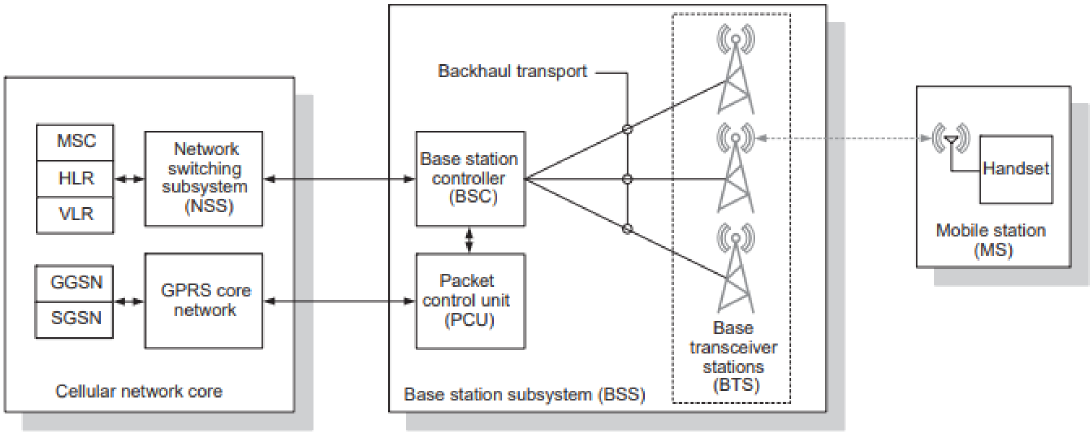

MSC
HLR
VLR
Network switching subsystem (NSS)
GGSN
SGSN
GPRS core network
Cellular network core
Backhaul transport
Base station controller (BSC)
Packet control unit (PCU)
Base station subsystem (BSS)
Base transceiver stations (BTS)
Handset
Mobile station (MS)

Figure 2. 1 GSM network block diagram [6]

The basic network switching, cell site backhaul, radio resource, and subscriber interface network elements are common to most mobile wireless network types. Vendor-specific implementations of mobile wireless networks contribute significantly to the redundancy deployment options available to network designers.

2.2.1 Network Switching Subsystem (NSS)

The NSS, or Network Subsystem, forms an integral part of the mobile wireless network, overseeing call switching and mobility management functions. The NSS consists of a number of discrete network elements. Some of the network elements are integrated into other elements while others are standalone devices [7][8]. The core network elements within the NSS are mobile switching center (MSC), home location register (HLR), visitor location register (VLR), authentication center (AuC), equipment identity register (EIR).

The MSC is a crucial component of the NSS and is necessary for the mobile wireless network to function. The MSC is typically set up as a region-wide, one-for-one redundant network component. To share the network load, large carriers frequently use several MSCs that are geographically different. This distributed architecture raises the availability of the network. In a topology with many MSCs, the failure of a single MSC does not result in a catastrophic network failure. Multiple geographically different MSC devices are only economically feasible for big subscriber counts due to the high cost of MSC network components [8].

2.2.2 Base Station Controller (BSC)

In cellular networks, the base station controller is in responsible for controlling the radio resources and handset requests provided by the base transceiver stations with which it is connected. Large-scale mobile wireless networks' BSCs are capable of controlling hundreds of separate base transceiver stations (BTSs). The BSC's criticality invites redundancy in its execution. The majority of mobile wireless networks use a one-for-one redundant hot standby BSC [8]. Figure 2.2 shows a base station subsystem (BSS) with a single BSC and a number of associated BTS units.

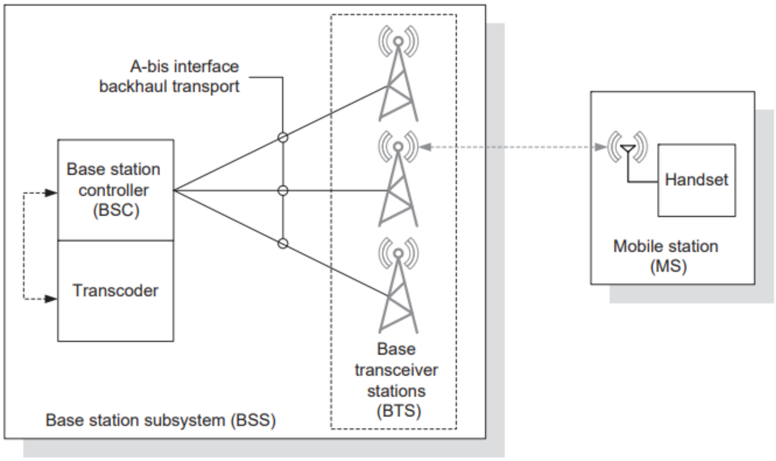


Figure 2. 2 Base station subsystem block diagram [7]

Base station controllers can be configured as stand-alone devices or as devices that also incorporate other functions and the base station controller function. Some BSC devices incorporate the “transcoding” function directly into the BSC while others utilize a separate device to perform the transcoding function. Transcoding refers to the operation of changing the voice signal from one coding scheme to another [6].

2.2.3 Standard Mobile Communication Call Flow

A mobile phone call is made possible by a complicated yet effective technique that seamlessly connects callers in different locations. When a call is originated, the sender's handset converts the dialed number into electronic signals, which are then transmitted via radio waves to the nearest base transceiver station (BTS). These signals are directed to the Mobile Switching Center (MSC), a crucial element of the mobile network, which determines the recipient's location by retrieving from visitor location register (VLR) and home location register (HLR) database components. The call request is routed to the recipient's network, and the recipient's base transceiver station (BTS) relays the incoming call signal to the intended receiver. Upon answering, the recipient's voice is transformed into electronic signals and sent back through the BTS, ultimately reaching the MSC. This center merges the two streams of voice data, establishing a real-time communication channel.

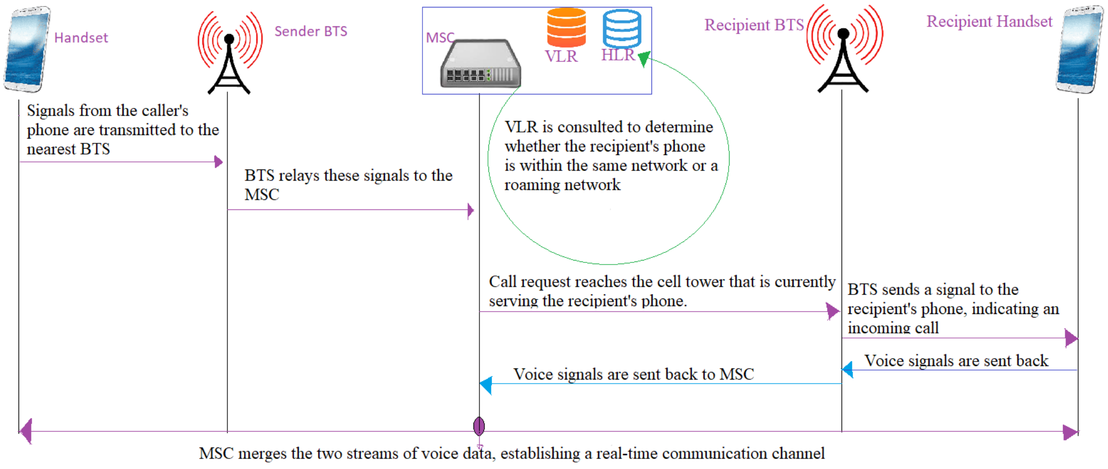

Handset
Sender BTS
MSC
VLR
HLR
Recipient BTS
Recipient Handset
Signals from the caller's phone are transmitted to the nearest BTS
BTS relays these signals to the MSC
VLR is consulted to determine whether the recipient's phone is within the same network or a roaming network
Call request reaches the cell tower that is currently serving the recipient's phone.
BTS sends a signal to the recipient's phone, indicating an incoming call
Voice signals are sent back
Voice signals are sent back to MSC
MSC merges the two streams of voice data, establishing a real-time communication channel

Figure 2. 3 Mobile Call Flow

Throughout the call, data packets containing voice information travel between devices, ensuring efficient data transmission. The call concludes when either party ends the conversation, prompting their respective phones to signal the BTS, which then informs the MSC to terminate the connection [7][8].

Figure 2.3 process showcases the remarkable orchestration of technology that supports every mobile phone call, allowing individuals to connect seamlessly regardless of their physical locations.

2.3 Mobile Wireless Network Interconnection and Roaming Service

Interconnect and roaming are essential components of modern telecommunication systems, enabling seamless connectivity for mobile users across different networks and geographic regions. Interconnect refers to the physical and logical connections between different telecommunications networks, while roaming allows mobile users to maintain connectivity while moving outside their home network's coverage area. Interconnect agreements are established between network operators to exchange traffic and allow users to communicate across different networks. These agreements are crucial for the smooth operation of mobile networks, ensuring that calls, text messages, and data can be routed seamlessly between different providers [9].

To achieve seamless interconnect between operators, agreement must be in place with the key aspects of traffic exchange, signaling, and interworking. Traffic exchange defines the terms and conditions for exchanging traffic between networks, including pricing and quality of service (QoS) parameters. Signaling refers to the establishment of signaling protocols and procedures for routing calls, text messages, and data between networks. Interworking ensures compatibility between different network technologies and protocols to enable seamless interoperation. Hence, interconnect agreements play a critical role in enabling the global reach of mobile telecommunication systems, allowing users to communicate with each other regardless of their network provider or location [9].

When a mobile user is outside the service region of their home network, roaming allows them to continue using their phones. To establish connectivity, a mobile user's device communicates with the foreign network's infrastructure when they join it. AAA (authentication, authorization, and accounting) procedures are used in this process to make sure that users are allowed to roam on the foreign network and that their usage is appropriately recorded [9].

The key aspects of roaming service include location management, mobility management and billing. Location management ensures tracking the location of mobile users as they move between different networks to ensure seamless handover of calls and data sessions. Mobility management maintains user sessions and provides continuity of service as mobile users move between networks; and billing system establishes the mechanisms for billing roaming charges between network operators [9].

2.4 Telecom Billing System

A software program that controls the billing procedure for telecom services is called a telecom billing system. It oversees creating bills, setting tariffs, gathering usage data, and maintaining client accounts. Due to their complexity and need to process large numbers of transactions, telecom billing systems are essential to telecom operators' profitability.

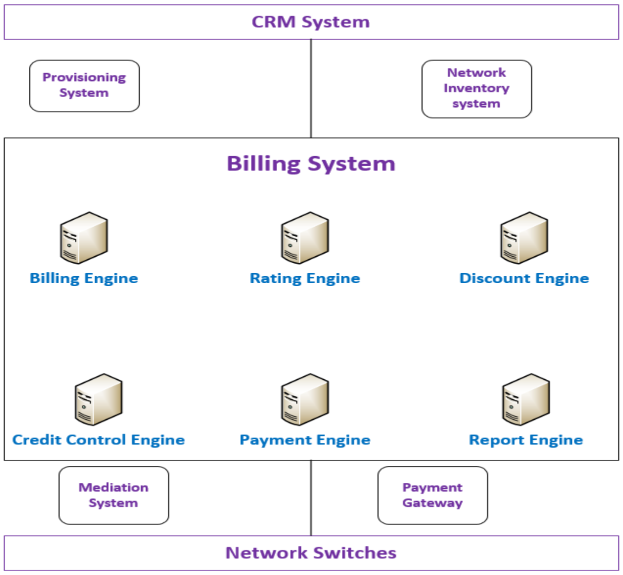


Figure 2. 4 Telecom Billing Architecture [10]

Customer Relationship Management (CRM), provisioning system and Mediation system are a major components of telecom billing system. CRM systems provide a centralized platform for storing, managing, and analyzing customer data, enabling telecom operators to gain valuable insights into customer behavior, preferences, and needs. A provisioning system is a crucial component that manages the process of preparing and equipping a network to provide new services to users. It encompasses a range of activities, from assigning phone numbers and installing equipment to configuring network elements and activating services. A mediation system plays a critical role in bridging the gap between different network elements and applications. It acts as a translator, transforming and harmonizing data formats and protocols to enable seamless communication and data exchange between disparate systems [10].

2.5 Mobile Wireless Communication Vulnerabilities

Mobile wireless communication infrastructure is a complex system with multiple components, and it is susceptible to various vulnerabilities that can be exploited by fraudsters at the infrastructure level, that result in a significant security challenge and revenue loss [4].

Table 2. 1 Mobile Communication Vulnerabilities

Vulnerabilities	Target Component	Description
Rogue base stations	• Base Station	Fraudsters can deploy unauthorized base stations that mimic legitimate ones, intercepting voice calls, SMS, and data packets from connected devices.
Denial of Service	• Base Station	Fraudsters might flood base stations with requests or signals, causing them to become overloaded and disrupting network services.
Firmware and Software	• Base Station • MSC • HLR • VLR	Outdated firmware and software in network components can contain known vulnerabilities that fraudsters can exploit.

Packet Interception	• MSC • Backhaul • Core Network	Fraudster might intercept and manipulate data packets traveling through the network, compromising data integrity, confidentiality and availability.
Network Traffic Analysis	• MSC • Backhaul • Core Network	Fraudster might analyze network traffic patterns to gain insights to identify potential targets.
Roaming	• MSC • Backhaul • Core Network	While roaming, inconsistent security standards between various networks can give attackers opportunity to take advantage of relaxed security controls.

Table 2.1 describes summarized view of mobile communication infrastructure level vulnerabilities and mobile network operators shall put in place strong security measures, such as frequent security assessments, reliable encryption protocols, intrusion detection systems, access controls, and continuous monitoring, to mitigate these risks. Regulatory agencies, manufacturers, and operators must work together to make security a high priority throughout the whole mobile wireless communication infrastructure.

2.6 Telecommunication Frauds

The term "*telecommunication fraud*" refers to a broad range of fraudulent practices that take advantage of system vulnerabilities and technical holes in telecommunication networks to gain unauthorized access to confidential data, financial benefit, or other advantages. This kind of fraud takes advantage of how communication networks are interconnected, technology and social engineering. Fraudsters use a variety of tricks to deceive users into disclosing private information like passwords, financial information, or personal identification, including phishing emails, vishing phone calls, and SMS scams [2][7].

Table 2. 2 Common Types of Telecom Fraud

Telecom Fraud	Description
Interconnect bypass fraud (SIM box Fraud)	This involves routing foreign calls through a local network element using a device called a SIM box. The fraudster target is to avoid paying international call tariff.
International revenue sharing fraud (IRSF)	Fraudster will exploit complexities of international mobile networks and revenue-sharing agreements between telecommunication operators. The fraudsters intentionally generate a large volume of international calls to premium-rate or high-cost destinations, with the intention of generating revenue that is shared between the fraudsters and the operators hosting the premium services.
Wangiri Fraud	Fraudsters make missed calls to victim's phones, then the victim calls back the number, thinking that it is a missed call from a legitimate source. Often resulting in premium-rate charges.
Caller ID Spoofing	In order to appear as though a call is coming from a reliable source, attackers change caller ID information, tempting victims to answer and reveal information.
Cloning	Fraudsters can clone copies of a subscriber's SIM card to gain unauthorized utilization of their services.
PBX Hacking	Fraudsters gain unauthorized access to private branch exchange (PBX) systems used by organizations to make long-distance calls, leading to substantial phone bill charges.
Smishing	Fraudsters use SMS messages to deceive users into clicking on malicious links or downloading malware-infected attachments.
Phishing	Fraudsters employ fraudulent emails, texts, or phone calls to deceive users into disclosing sensitive information such as passwords, credit card numbers, or personal identification details.

2.6.1 SIM-box Fraud (Interconnect Bypass Fraud)

A SIM-box, often referred to as a SIM server or SIM bank, is a hardware equipped with SIM card slots, GSM module, antennas, connectivity ports, CPU, memory, colling system, power supply and enclosure that used in the telecommunications industry to manage numerous SIM cards at once. SIM-boxes are used frequently for testing, automation, and effectively managing a huge number of SIM cards with the capability of simulating mobile devices. However, SIM-box also be used fraudulently by many fraudsters [6].

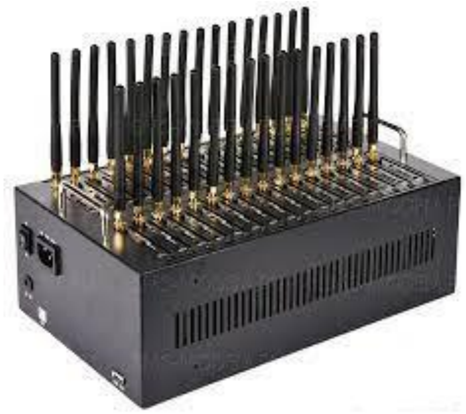


Figure 2. 5 SIM-Box Device [Internet]

SIM-box fraud, often referred to as SIM card fraud or interconnect bypass fraud, is a sophisticated and continuously evolving type of cyber-crime that poses serious challenges for operators, networks, and regulatory bodies. It targets to achieve international call by local call price with the help of sim-box device and several local SIM cards.

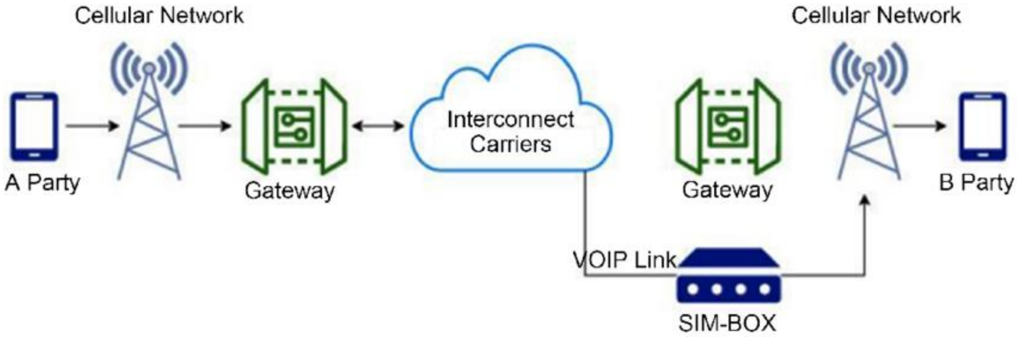


Figure 2. 6 . SIM-Box fraud rout of international call [6]

In SIM-box fraud, the fraudsters insert several local SIM cards into the SIM-box slots that are configured with software to simulate regular mobile devices. During an international call is made to a targeted destination, the SIM-box intercepts the call and routes it through a local SIM card corresponding to the destination. Then, the fraudsters avoid international call charges by making the international call appear as local-to-local communication [6][7].

SIM-box fraud exploits vulnerabilities of telecommunications infrastructure and leads to financial losses for legitimate telecom operators. Hence, monitoring call patterns attentively, detecting suspect SIM cards used in connection with SIM-boxes, and working together with regulatory agencies and telecom companies are all necessary to identify and prevent SIM-box fraud [2].

2.6.2 SIM-box Fraud Detection Mechanisms

SIM-box fraud is a sophisticated type of telecom fraud that must be detected with a multifaceted strategy that combines different detection techniques to identify anomalies in the communications network. Call detail record (CDR) analysis is a crucial technique that uses algorithms to track variables including call frequency, duration, and the ratio of incoming to outgoing calls. Unusual trends, such as a high number of incoming calls with brief durations, can indicate possible SIM-box fraud [6].

Another crucial method is to examine the International Mobile Subscriber Identity (IMSI) and International Mobile Equipment Identity (IMEI) numbers connected to SIM cards and devices. Deviations from predicted usage patterns may be an indication of fraud. Geospatial research examines the relationship between call volume and locations, exposing anomalies that could point to the existence of SIM-box fraud [4][6].

Detection accuracy is improved by keeping monitoring on call traffic abnormalities, spotting suspicious behavior patterns, using voice biometrics, audio analysis, machine learning and artificial intelligence algorithms. Regulatory agencies, telecom providers working together, and information exchange about suspect numbers and trends all help to speed up the detection process. Together, these techniques can support in the early detection of SIM-box fraud, the protection of legal operators, and the overall integrity of the telecommunications industry.

2.7 Data-Driven Approach

The data-driven approach in data science establishes a methodology where decisions and insights are drawn from a comprehensive analysis of data. This approach brings huge significance in modern business and research landscapes, as it supports informed decision-making through experimental evidence rather than assumptions or perception. A crucial phase of this approach involves the collection and preprocessing of data. Data can be sourced from various enterprise systems, is precisely cleansed, transformed, and integrated to eliminate errors, inconsistencies, and redundancies, ensuring the data's accuracy and reliability as the foundation for subsequent analysis.

The fundamental components of the data-driven strategy are modeling and machine learning. This process uses advanced algorithms to extract useful patterns, forecasts, and classifications from the data. Depending on the nature of the current challenge, different strategies, such as supervised, unsupervised, and semi-supervised learning, are used. Engineering features, which extract useful properties from raw data, is a crucial stage as well. To improve the performance of data-driven models, this approach uses modifications such one-hot encoding, dimensionality reduction, and scaling [11].

2.8 Machine Learning Algorithms

Machine learning is a set of computer algorithms that can be applied to machines associated with artificial intelligence. These machines are often responsible for tasks such as recognition, prediction, diagnosis, and other similar tasks. These machines learn from the data that is fed to them. The data is used to train the machine to perform its functions, it is called training data. Machines “learn” from the training data and continue to learn when new data is fed to them. Hence, Machine learning can also be defined as the process of solving a practical problem by gathering a dataset, and algorithmically building a statistical model based on that dataset. That statistical model is assumed to be used somehow to solve the practical problem [11]. Different learning mechanisms are used to analyze the training data based on the problems that need to be solved. These mechanisms can be classified into three – supervised learning, unsupervised learning, and reinforcement learning [11][12].

Machine learning is an intersection of different statistical and mathematical subjects, each subject brings a new methodology that can be incorporated in both machine learning and artificial intelligence [11]. The fundamental goal is to uncover patterns, relationships, and insights within complex datasets, enabling systems to make predictions, decisions, and recommendations without explicit programming. The approach is established in the recognition that data itself contains valuable information, and algorithms can be designed to autonomously extract and utilize this information for a wide array of applications.

In the telecommunications industry, machine learning has a significant impact on helping to estimate organizational profits or losses, detect telecom fraud, and predict service consumer turnover to ensure customer satisfaction. Network optimization is one of its main uses due to telecommunication networks are complex, with many interrelated parts. Network administrators may optimize routing, lower downtime, and improve overall performance by using machine learning algorithms to analyze massive volumes of network data and detect trends and abnormalities [6].

In addition, machine learning is crucial for predictive maintenance. Machine learning models can forecast equipment failures and suggest preventative maintenance by evaluating data from various network components, such as routers and switches.

This reduces downtime and operational expenses. Particularly in the customer service domain, chatbots and virtual assistants can also be deployed with machine learning engines to manage various customer requests and improve customer satisfaction.

2.9 Supervised Learning

Supervised learning is a type of machine learning where the model is trained on labeled data. In supervised learning, training is a crucial step where data about instances in the past are given to the machine to assist it predict any future events. The labeled data fed essentially consists of training examples, these examples consist of inputs and the desired outputs. These desired outputs are also known as supervisory signals. The machine then uses a supervised machine algorithm that creates an inferred function that is used to forecast any events. If the outputs are discrete, the function is called a classifier, and if the outputs are continuous, the function is known as a regression function. The function is used to predict the outputs for future inputs too. This algorithm is used to generate a generalized method to reach the output from the data that was fed in as input [11].

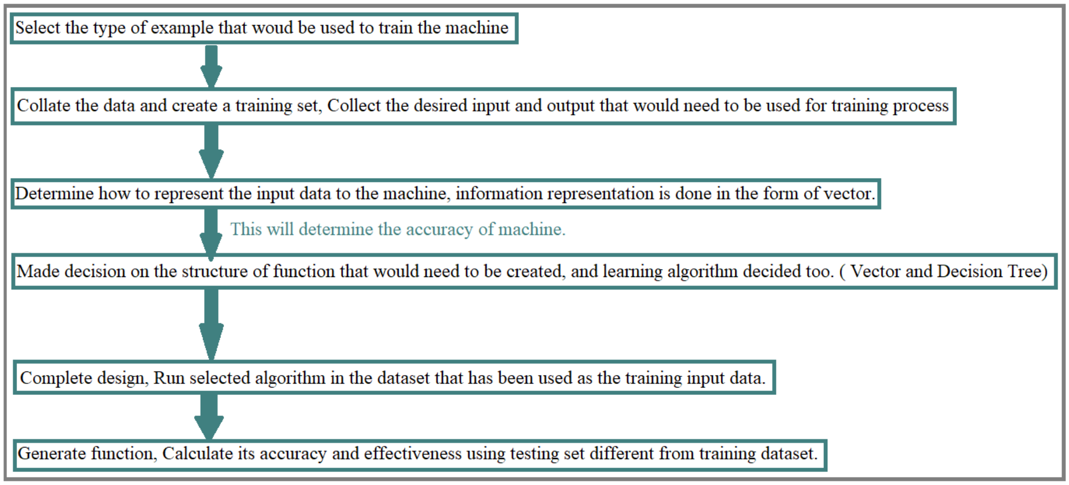

Select the type of example that woud be used to train the machine
Collate the data and create a training set, Collect the desired input and output that would need to be used for training process
Determine how to represent the input data to the machine, information representation is done in the form of vector.
This will determine the accuracy of machine.
Made decision on the structure of function that would need to be created, and learning algorithm decided too. (Vector and Decision Tree)
Complete design, Run selected algorithm in the dataset that has been used as the training input data.
Generate function, Calculate its accuracy and effectiveness using testing set different from training dataset.

Figure 2. 7 High Level Flow of Supervised Learning

In supervised learning, the dataset is the collection of labeled examples $\{(x_i, y_i)\}_{i=1}^{N}$. Each element xi among N is called a feature vector. A feature vector is a vector in which each dimension j = 1..., D contains a value that describes the example somehow. That value is called a feature and is denoted as $x^{(j)}$ [11].

For all examples in the dataset, the feature at position j in the feature vector always contains the same kind of information. The goal of a supervised learning algorithm is to use the dataset to produce a model that takes a feature vector x as input and outputs information that allows deducing the label for this feature vector. For instance, the model created using the dataset of phone number could take as input a feature vector describing a call record and output a probability that the phone number has executing fraud.

During the usage of supervised learning algorithms, some issues might be arise associated with bias - variance tradeoff, function complexity and dimensionality of input space []. While working with machine learning the ***bias - variance tradeoff*** is the most common constraint and it occurs when a machine is trained using some training data sets, it gives predictions that are systematically incorrect for certain output. It can be said that the algorithm is biased towards the input data set. A learning algorithm can also be considered to have a high variance for input. This occurs when the algorithm causes the machine to predict different outputs for that input in each training set [12].

Another concern is deciding the amount of training data used based on the complexity of the classifier or regression function to be generated. Suppose the function to be generated is simple, a learning algorithm that is relatively inflexible with low variance and high bias will be able to learn from a small amount of training data. However, on many occasions, the function will be complex. This can be the case due to a large number of input features being involved or due to the machine being expected to behave differently for different parts of the input vector. In such cases, the function can only be learned from a large amount of training data. These cases also require the algorithms used to be flexible with low bias and high variance [12].

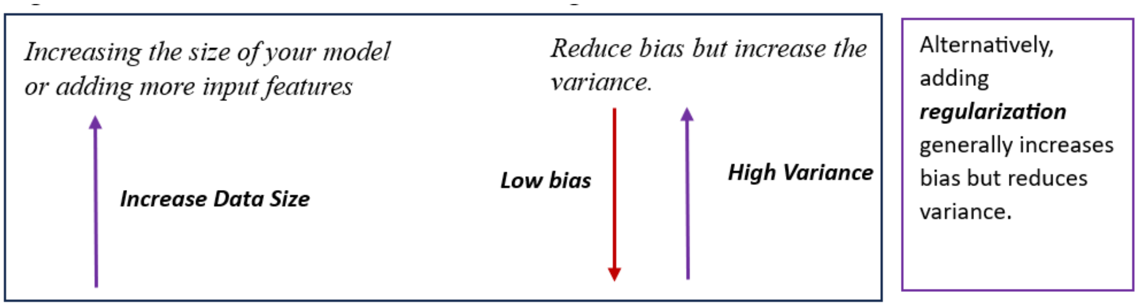

Increasing the size of your model or adding more input features
Increase Data Size
Reduce bias but increase the variance.
Low bias
High Variance
Alternatively, adding regularization generally increases bias but reduces variance.

Figure 2. 8 Bias vs. Variance tradeoff

2.9.1 Decision Tree Learning

Decision tree learning is a supervised machine learning algorithm that can be used for both classification and regression problems. It works by constructing a tree-like model of the data that is comprised of a root node, branches, internal nodes, and leaf nodes where each node in the tree represents a decision and each leaf node represents a prediction. To construct a decision tree, the algorithm uses a recursive partitioning process, where each node is divided into child nodes, and this process continues until a stopping criterion is met. This ensures that data can be effectively subdivided into smaller, more manageable subsets [12][13].

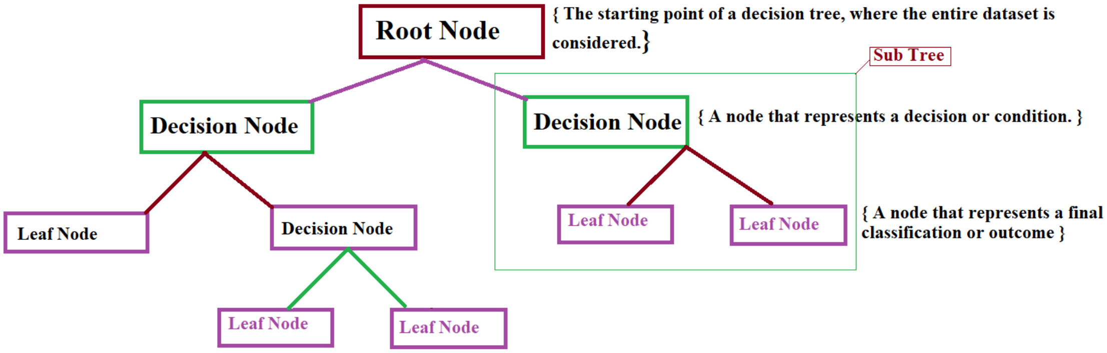

Root Node
{ The starting point of a decision tree, where the entire dataset is considered.}
Sub Tree
Decision Node
Decision Node
{ A node that represents a decision or condition. }
Leaf Node
Decision Node
Leaf Node
Leaf Node
{ A node that represents a final classification or outcome }
Leaf Node
Leaf Node

Figure 2. 9 Decision Tree Structure

Typically, binary splits are performed through decision trees, which means that each node splits the data into two subsets depending on a particular feature or condition. This assumes that every choice can be expressed as a binary option. During the decision tree building process top-down, greedy approaches are used, and each split is selected to maximize information gain or reduce impurity at the current node. To measure impurity at the current node, entropy and Gini impurity measurements can be applied to measure how successfully a split distinguishes classes. The impurity measurement selection might have impact on achieving optimal decision tree construction.

Entropy is a measure of the uncertainty or randomness in a dataset. A branch with an entropy of zero is a leaf node and A branch with entropy more than zero needs further splitting [13].

$$E(s) = \sum_{i=1}^{c} -p_i \, log_2 \, P_i \qquad (2.1)$$

Where, S Current data set,

i set of classification in S | event in the current data set S

Pi Probability of an event i.

In the context of decision tree learning, entropy is used to measure the impurity of a node in the tree. A node with a high entropy is impure, meaning that it contains a mix of different classes. A node with a low entropy is pure, meaning that it contains mostly data points from the same class. Constructing a decision tree is all about finding an attribute that returns the highest information gain and the smallest entropy. Information gain is a decrease in entropy. It computes the difference between entropy before split and average entropy after split of the dataset based on given attribute values [13].

Information gain is the other fundamental idea to choose a root node and decision node. The method continuously computes for the other attributes to build the hierarchy of the tree starting with the attribute with the largest information gain value, which is then assigned as the root node.

$$Information\ Gain = Entropy(before) - \sum_{j=1}^{k} Entropy(j, after) \qquad (2.2)$$

2.9.2 Random Forest Algorithm

A random forest algorithm is a supervised machine learning algorithm that can be used for both classification and regression tasks. It works by constructing a multitude of decision trees and then averaging their predictions to produce a final prediction [14].

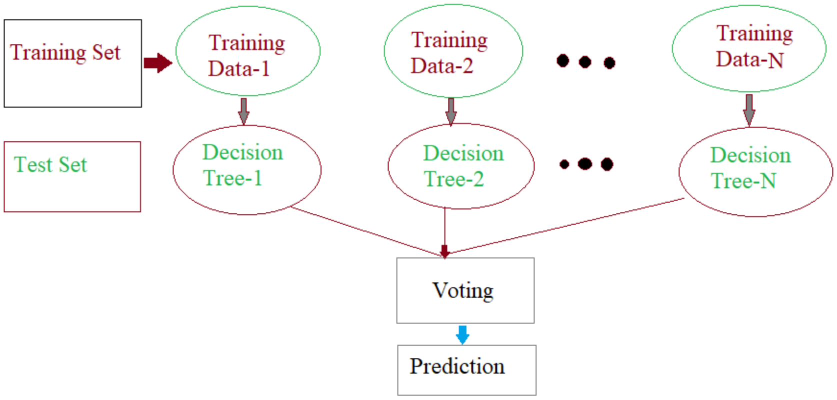


Figure 2. 10 Random Forest Algorithm

A random forest algorithm is more accurate and robust than individual decision trees because they average out the predictions of many different trees. This helps to reduce the effects of overfitting and noise. A random forest machine learning result depends on the strength of the individual tree of the forest and correlation of any tree in the forest, each individual tree of the forest gave their own decision and the most voted is taking as the result of the algorithm [12][14].

2.9.3 Support Vector Machine (SVM)

Support vector machines (SVMs) are a powerful machine learning algorithm that can be used for both classification and regression problems. SVMs are based on the idea of finding a hyperplane in the feature space that separates the data into two classes. The hyperplane is chosen such that the distance between the data points and the hyperplane is maximized. This maximizes the margin of separation between the two classes, which helps to improve the generalization performance of the model [11][15].

Support vector machine sees every feature vector as a point in a high-dimensional space, then it puts all feature vectors on an imaginary dimensional plot and draws an imaginary dimensional line (a hyperplane) that separates training set with positive labels from training set with negative labels. In machine learning, the boundary separating the training set of different classes is called the decision boundary.

The equation of the hyperplane is given by two parameters, a real-valued vector w of the same dimensionality as our input feature vector x, and a real number b like this [11]:

$$wx - b = 0 \tag{2.3}$$

where the expression **wx** means $w^{(1)}x^{(1)} + w^{(2)}x^{(2)} + \cdots + w^{(D)}x^{(D)}$ and D is the number of dimensions of the feature vector **x**.

The goal of the SVM learning algorithm is to leverage the dataset and find the optimal values w^* and b^* for parameters w and b. Once the learning algorithm identifies these optimal values, the model f(x) is then defined as:

$$f(x) = sign(w^*x - b^*) \tag{2.4}$$

Where, **f(x)** Linearly discriminant function **x** Feature vector (input vector), **w** Adjustable weight vector to control direction of the hyper plane, **b** Bias which control the hyperplane position.

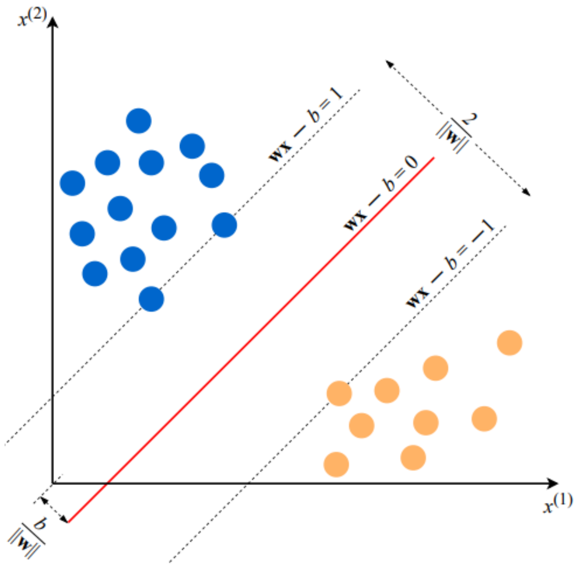


Figure 2. 11 An example of an SVM model for two-dimensional feature vectors [7]

2.10 Unsupervised learning

Unsupervised learning is a type of machine learning where the model is not given any labeled data. Instead, the model is allowed to learn patterns and relationships in the data on its own. Unsupervised learning algorithms are used for a variety of tasks, including anomaly detection, which locates unusual or unexpected data points, dimensionality reduction, which allows reducing the number of features in a dataset without losing too much information, and clustering, which groups similar data points together [11].

Unsupervised learning algorithms are typically more computationally expensive to train than supervised learning algorithms. However, they can be used to solve a wider range of problems, and they are not dependent on labeled data.

The amount of computing power to train unsupervised learning algorithms is often higher than that of supervised learning algorithms. They don't rely on labeled data, though, and can be applied to a larger range of issues. K-means clustering, Hierarchical clustering, Principal component analysis (PCA), t-distributed stochastic neighbor embedding (t-SNE) and Anomaly detection is the most popular algorithms of unsupervised learning [11].

2.11 Related Work

This section examines the contributions of different authors to the detection and prevention of SIM-box fraud with frameworks, models, and architectural artifacts. The artifact for this thesis will be a functional model for near real-time SIM-box detection that is going to be applied for telecom voice infrastructure. The proposed solution will assist telecom operators in fraudulent activities detection based on huge CDR data analysis with the consideration of near real-time effect.

Frehiwot Mola [16], utilized call detail record (CDR) from Ethio-telecom data source to develop a model that can categorize a particular number as normal or fraudulent using data mining techniques. For classification, the author indicates decision trees, rule-based, neural network and hybrid algorithms are applied. In the paper it discussed that nine features are selected and extracted after data analysis performed from the given data set. The author describes two month of data is used for the work and using expert judgement and various data filtration methods of call duration, total call per day and, calling time 23,904 call record has utilized and finally 12,686 records used for experimentation. The limitation of this work is that the author has not clearly defined the step-by-step methodology regarding classification algorithm and how the mentioned 99.4795% accuracy is achieved. In addition to this, the amount of data that is used is very small that tend to higher variance, that the model's predictions fluctuate more significantly with different training data samples and the model qualify on memorizing the specific instances rather than generalizing to new, unseen data.

Kahsu Hagos [17], developed a model that can classify Call Detail Records (CDRs) as legitimate subscriber and fraudulent using three classification techniques of Random Forest (RF), Artificial Neural Network (ANN) and Support Vector Machine (SVM). The author categorizes and aggregated the data set as 4-hour, daily and monthly, then fed into the three selected algorithms to build the model. Hence based on the author finding the Random Forest (RF) algorithm with 4 hour aggregated dataset scored 95.99% accuracy and a lesser false-positive compared with the remaining two (ANN and SVM) algorithm with 4 hours, daily and Monthly aggregated dataset. The identified gap of this thesis work is the generated artifact seems something like comparative analysis of different supervised machine learning algorithms and from the perspective of the thesis initial objective, the artifact should be a model, framework or architectural solution that contextualized with the problem definition. In addition, from the perspective of problem complexity and required accuracy, the consumed sample size is small.

Fitsum Tesfaye [18], proposed near real-time SIM-box detection through collect call detail record (CDR) data, then relevant attributes were selected and preprocessing such as data cleaning, integrating and aggregating tasks were performed. Using sliding window (SW) aggregation mode that provides a relevant dataset instance and reduces detection delay by using supervised Machine Learning (ML) algorithm. Three supervised ML classifier algorithms were used, namely Random Forest (RF), Artificial Neural Network (ANN), and Support Vector Machine (SVM) with the two validation techniques 10-fold cross-validation and supplied test. The identified gap of this work is the proposed solution doesn't show the architectural mechanism of how near real-time detection is there while continuous CDR flow is required to achieve near real-time detection. In addition to this, from the perspective of problem complexity and required accuracy, the consumed sample size is very small.

Roselina Sallehuddin [19], applied two classification techniques of Artificial Neural Network (ANN) and Support Vector Machine (SVM) in which the author was used to develop SIM-box fraud detection model. The classification uses nine selected features of data extracted from Customer Database Record. The performance of ANN is compared with SVM to find which model gives the best performance. From the experiments, it is found that SVM model gives higher accuracy compared to ANN by giving the classification accuracy of 99.06% compared with ANN model, 98.71% accuracy. Besides, better accuracy performance, SVM also requires less computational time compared to ANN since it takes lesser amount of time in model building and training. This paper is like a comparative analysis between SVM and ANN for the case of SIM-box fraud detection but doesn't justify scientifically why decision tree and Naïve Bayes classification techniques are not well performing for the specified problem definition.

The referenced related works are applied the methodologies of data mining, supervised machine learning algorithm of decision tree, random forest, and support vector machine to uncover the unseen relation between service number and classification (normal and fraudulent) using call details record. In this thesis, we applied a SQL Analysis and data driven machine learning approach to track call patterns and draw findings that can detect SIM-box fraud numbers effectively.

Table 2. 3 Related Work Summary

No	Author	Title	Publication Year	Applied Technique	Limitation
1	Frehiwot Mola	Analysis and Detection Mechanisms of SIM Box Fraud in The Case of Ethio Telecom	2017	Decision Tree, rule-based induction, and hybrid of J48 and PART	Small Dataset (12,686 records)
2	Kahsu Hagos	SIM-Box Fraud Detection Using Data Mining Techniques: The Case of ethio telecom	2018	RF NN SVM	Small Dataset Weka tool
3	Fitsum Tesfaye	Near-Real Time SIM-box Fraud Detection Using Machine Learning in the case of ethio telecom	2020	RF ANN SVM	Small Dataset Weka tool
4	Roselina Sallehuddin	Detecting SIM Box Fraud by Using Support Vector Machine and Artificial Neural Network	2015	SVM ANN	2,126 fraud subscribers and 4,289 normal using small datasets

Chapter Three

Research Design and Methodology

In order to address the specified research question, this chapter will discuss the research design and applied methodology that shall be used to achieve the research objective. This study follows Experimental research methodology and both quantitative and qualitative data collection methods have been used to gain comprehensive understanding of the current telecom solution deployment architecture, SIM-box fraud techniques, available fraud management system solutions, and the infrastructure perspective of how telecom call detail record (CDR) data is organized, due to this critically determine the efficiency of SIM-box detection solution to manage issues in near real-time manner.

The goal of this thesis is to consume and analyze telecom CDR data and produce an efficient SIM-box detection model that can classify subscribers according to their call behaviors. Hence, the primary input data for this thesis shall be a CDR and its flow of processing is depicted as below.

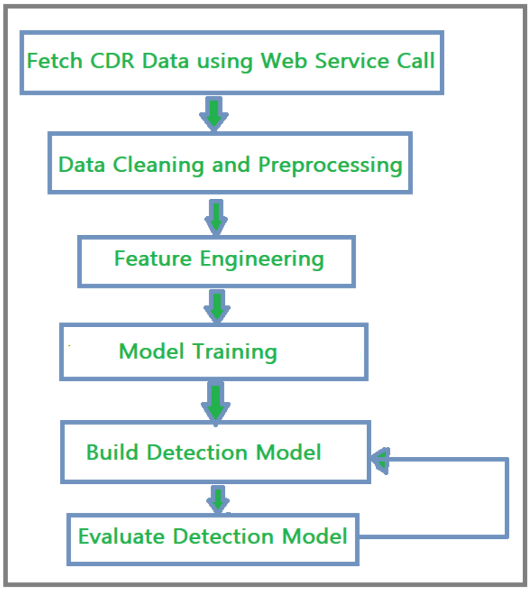


Figure 3. 1 Steps of SIM-box Detection Model Development

3.1 Retrieving CDR Data

In telecommunication systems, signaling data, call detail record and customer profile data is the most prominent type of data that needs to perform various computations and decisions. Call Detail Records (CDR) are the data records that are automatically produced by a mobile network system, telecoms exchange or other telecommunications systems. CDR records are produced for every phone call that passes through the dedicated infrastructure and contains all the data pertaining to that call, ranging from time, date, duration, source number and destination number to more detailed records that includes further information and possibly used for billing, law enforcement, fraud management, etc.

In order to support near real-time SIM-box fraud detection, continuous CDR data flow must be in place between the telecom operator infrastructure and the SIM-box fraud detection system using the below architectural specification.

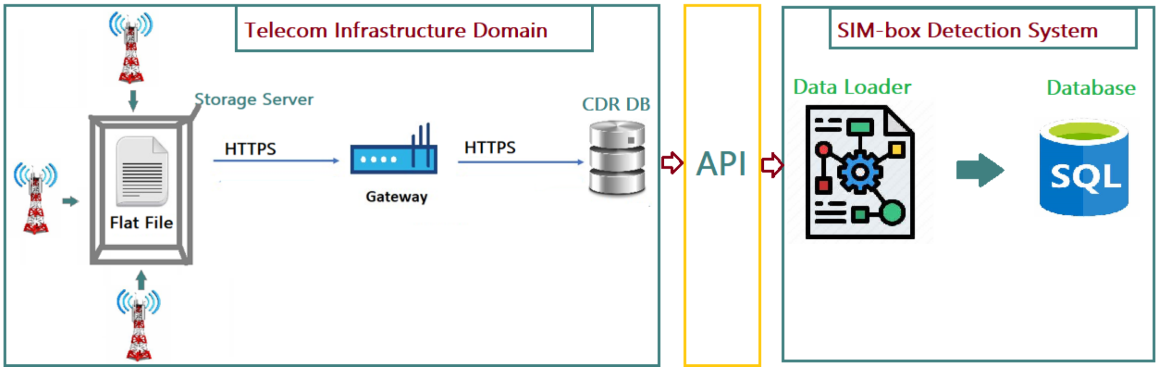
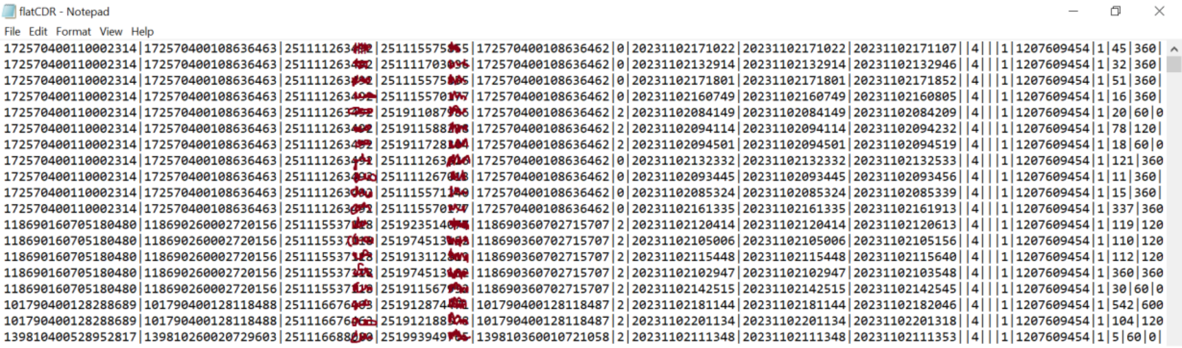


Figure 3. 2 CDR Data Flow Diagram

The raw CDR data shall be collected from different mobile operator solutions like CBS, CRM, HLR, VLR DBs, and base transceiver station (BTS) into a central storage server equipped with flat file database. A flat file database is a type of database that stores all its data in a single file. The data in a flat file database is typically organized in a table format, with each row representing a record and each column representing a field. Flat file databases are simple and easy to use, but they can be difficult to manage as they grow large. Hence, within a configured timeframe it will be imported into CDR database iteratively.

In order to achieve near real-time SIM-box detection, there must be seamless data integration between telecom infrastructure and SIM-box detection solution.

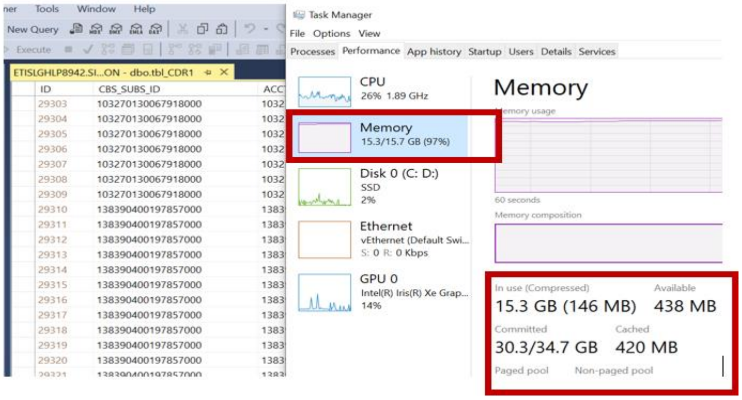
Therefore, API (Application Programming Interface) or Web service should be available to expose near real-time data for the SIM-box data fetcher component; then the data will continuously import into SQL Server database through data loader component that can query data from the web service iteratively.



```
172570400110002314|172570400108636463|251111263[redacted]|25111557[redacted]|172570400108636462|0|20231102171022|20231102171022|20231102171107||4|||1|1207609454|1|45|360|
172570400110002314|172570400108636463|251111263[redacted]|251111703[redacted]|172570400108636462|0|20231102132914|20231102132914|20231102132946||4|||1|1207609454|1|32|360|
172570400110002314|172570400108636463|251111263[redacted]|251115575[redacted]|172570400108636462|0|20231102171801|20231102171801|20231102171852||4|||1|1207609454|1|51|360|
172570400110002314|172570400108636463|251111263[redacted]|251115570[redacted]|172570400108636462|0|20231102160749|20231102160749|20231102160805||4|||1|1207609454|1|16|360|
172570400110002314|172570400108636463|25111126[redacted]|251911087[redacted]|172570400108636462|2|20231102084149|20231102084149|20231102084209||4|||1|1207609454|1|20|60|0
172570400110002314|172570400108636463|251111263[redacted]|251911588[redacted]|172570400108636462|2|20231102094114|20231102094114|20231102094232||4|||1|1207609454|1|78|120|
172570400110002314|172570400108636463|251111263[redacted]|25191172[redacted]|172570400108636462|2|20231102094501|20231102094501|20231102094519||4|||1|1207609454|1|18|60|0
172570400110002314|172570400108636463|251111263[redacted]|25111126[redacted]|172570400108636462|0|20231102132332|20231102132332|20231102132533||4|||1|1207609454|1|121|360
172570400110002314|172570400108636463|251111263[redacted]|25111126[redacted]|172570400108636462|0|20231102093445|20231102093445|20231102093456||4|||1|1207609454|1|11|360|
172570400110002314|172570400108636463|25111126[redacted]|25111557[redacted]|172570400108636462|0|20231102085324|20231102085324|20231102085339||4|||1|1207609454|1|15|360|
172570400110002314|172570400108636463|25111126[redacted]|251115570[redacted]|172570400108636462|0|20231102161335|20231102161335|20231102161913||4|||1|1207609454|1|337|360
118690160705180480|118690260002720156|251115537[redacted]|25192351[redacted]|118690360702715707|2|20231102120414|20231102120414|20231102120613||4|||1|1207609454|1|119|120
118690160705180480|118690260002720156|25111553[redacted]|251974513[redacted]|118690360702715707|2|20231102105006|20231102105006|20231102105156||4|||1|1207609454|1|110|120
118690160705180480|118690260002720156|251115537[redacted]|25191311[redacted]|118690360702715707|2|20231102115448|20231102115448|20231102115640||4|||1|1207609454|1|112|120
118690160705180480|118690260002720156|251115537[redacted]|251974513[redacted]|118690360702715707|2|20231102102947|20231102102947|20231102103548||4|||1|1207609454|1|360|360
118690160705180480|118690260002720156|251115537[redacted]|25191156[redacted]|118690360702715707|2|20231102142515|20231102142515|20231102142545||4|||1|1207609454|1|30|60|0
101790400128288689|101790400128118488|25111667[redacted]|25191287[redacted]|101790400128118487|2|20231102181144|20231102181144|20231102182046||4|||1|1207609454|1|542|600
101790400128288689|101790400128118488|25111667[redacted]|25191218[redacted]|101790400128118487|2|20231102201134|20231102201134|20231102201318||4|||1|1207609454|1|104|120
139810400528952817|139810260020729603|25111668[redacted]|25199394[redacted]|139810360010721058|2|20231102111348|20231102111348|20231102111353||4|||1|1207609454|1|5|60|0|
```

Figure 3. 3 Flat CDR DB

The data loader component is responsible for retrieving data from telecom CDR database infrastructure and store into SIM-Box Detection solution database continuously to enable the sense of near real-time detection. In the current Ethio-telecom infrastructure, the flat raw data is stored in a dedicated storage server that hosts additional database management system to move the flat raw data into relational database format for easy data manipulation and control. This process will take more than 21 hours to complete one day CDR for 70.3 million voice subscribers.



ID	CBS_SUBS_ID	ACC
29303	103270130067918000	1032
29304	103270130067918000	1032
29305	103270130067918000	1032
29306	103270130067918000	1032
29307	103270130067918000	1032
29308	103270130067918000	1032
29309	103270130067918000	1032
29310	138390400197857000	1383
29311	138390400197857000	1383
29312	138390400197857000	1383
29313	138390400197857000	1383
29314	138390400197857000	1383
29315	138390400197857000	1383
29316	138390400197857000	1383
29317	138390400197857000	1383
29318	138390400197857000	1383
29319	138390400197857000	1383
29320	138390400197857000	1383
29321	138390400197857000	1383

CPU
26% 1.89 GHz

Memory
15.3/15.7 GB (97%)

Disk 0 (C: D:)
SSD
2%

Ethernet
vEthernet (Default Swi...
S: 0 R: 0 Kbps

GPU 0
Intel(R) Iris(R) Xe Grap...
14%

Memory

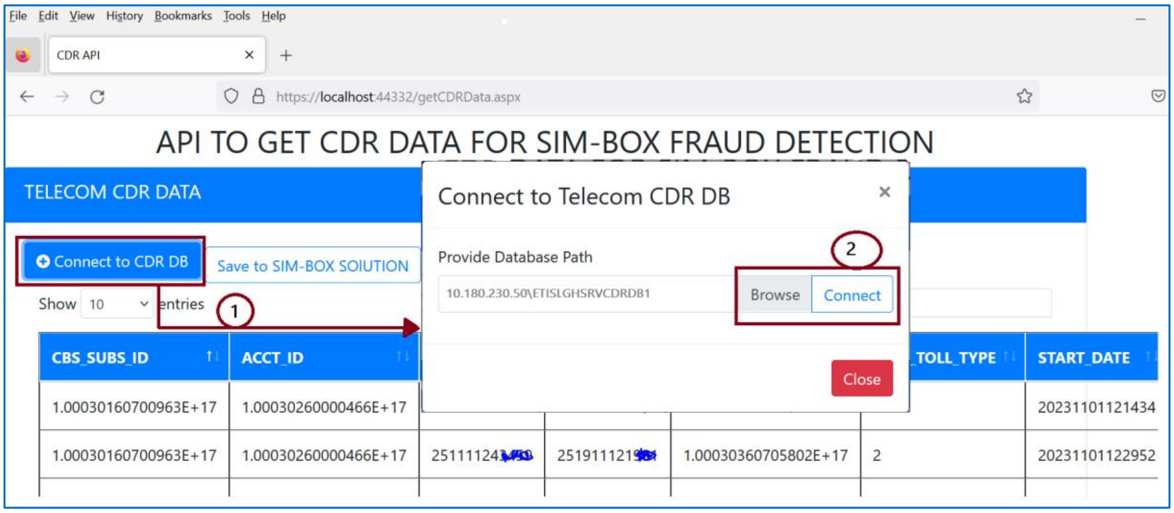
In use (Compressed): 15.3 GB (146 MB)
Available: 438 MB
Committed: 30.3/34.7 GB
Cached: 420 MB
Paged pool
Non-paged pool

Figure 3. 4 CDR Data Fetching Memory Utilization

Figure 4.4 describes, the computing power perspective and, based on the experimentation that we had to fetch one day CDR data into SIM-Box Solution it just consumed 97 % of 16 GB server memory resource.

3.2 CDR Data Description

As we can see from the section 4.1 and figure 4.5, Integration API has developed between telecom CRM infrastructure and SIM-Box Detection solution server using web service to enable near real-time CDR data flow, preprocessing, aggregation, detection, and visualization. Table 4.1 describes 20 main attributes of CDR, its supported data type and high-level description.



API TO GET CDR DATA FOR SIM-BOX FRAUD DETECTION

Connect to Telecom CDR DB

Provide Database Path

10.180.230.50\ETISLGHSRVCDRDB1

CBS_SUBS_ID	ACCT_ID				_TOLL_TYPE	START_DATE
1.00030160700963E+17	1.00030260000466E+17					20231101121434
1.00030160700963E+17	1.00030260000466E+17	25111124[redacted]	25191112[redacted]	1.00030360705802E+17	2	20231101122952

Figure 3. 5 CDR Data Integration Solution

Due to the fraudulent uses well-known and similar connectivity techniques for SIM-box fraud, we have preferred to use one month (November 2023) CDR data and for efficient data filtration different core network experts' judgment has applied to identify suspicious SIM's by checking other value-added service usage experience (SMS, Data), call initiating time, call duration, total call made, voice call usage continuity and balance charging history. Hence, we have just selected 1Million CDR data to proceed with further manipulation.

Table 3. 1: CDR Columns, Supported Data Type and Descriptions

No	Attribute	Data Type	Description
1	CBS_SUBS_ID	Big Integer	Customer subscription ID from Convergent billing system side.
2	ACCT_ID	Big Integer	Transaction ID for Charging System
3	SERV_NO	Integer	Mobile number that initiates the call.
4	OTHER_NO	Integer	Mobile number that received the initiated call.
5	CUST_ID	Big Integer	Customer ID from Customer Relation Management system side.
6	CALL_TOLL_TYPE	Integer	Charging Type (Peek hour, off peek hour)
7	START_DATE	Date Time	Established call connection date.
8	START_TIME	Date Time	Established call connection start time.
9	END_TIME	Date Time	Established call connection end time.
10	NET_TECH_TYPE	Integer	The network type that the call is using.
11	IMSI	Integer	International Mobile Subscriber Identity Number.
12	CELL_ID	Integer	A unique identifier assigned to each cell in a cellular network.
13	SGMT_TYPE	Integer	The mobile number payment type (Prepaid and postpaid).
14	PROFILE_ID	Integer	Database generated ID for customer profile.
15	SERV_TYPE	Integer	The utilized service type of voice, SMS, or Data.
16	CALL_DUR	Integer	Duration between call establishment start and end time.
17	CHRG_DUR	Integer	The call duration that considered by charging system.
18	FREE_DUR	Integer	Free call duration type if any discount or package is added to the service number.
19	ACTUAL_USG_DUR	Integer	Service utilization duration between call establishment start and end time.
20	Data_day	Date	The date that call is recorded.
21	ROAM_CNTRY_CODE	Integer	Country code for roaming service
22	MSC_ADDR	Integer	Message Switching Center Address Number
23	VOC_DATA_CALL_DUR	Integer	Established Call using data
24	VOC_DATA_CHRG_DUR	Integer	Charging Duration for call using data
25	DATA_TOTAL_VOL		
26	DATA_DNLD_VOL	Integer	The amount of data that downloaded
27	DATA_UPLD_VOL	Integer	The amount of data that uploaded
28	DATA_ONLINE_DUR	Integer	Duration in second for online stay
29	DATA_DNLD_FEE	Integer	Payment for Data download
30	DATA_UPLD_FEE	Integer	Payment for Data upload

31	DATA_CHRG_FEE	Integer	The amount of money expected to pay for data usage.
32	SMS_CALL_CNT	Integer	SMS counter
33	SMS_INTER_FEE	Integer	Payment for international text message
34	SMS_TOTAL_FEE	Integer	The amount of money expected to pay for SMS usage.
35	SIM_CreatedDate	Date	The date when the SIM is provisioned.

3.3 Data Selection

Working with large amounts of data in a data science project has drawbacks in terms of computational complexity, interpretation, and data quality and interpretation. Errors, inconsistencies, and biases are common in big data analysis, and these factors might produce false or misleading results. Furthermore, incomplete, and missing data points are frequent in large data, requiring the use of imputation techniques (which replace missing data with substituted values) or the exclusion of some data points, which may have an impact on analysis. From the point of interpretation, complex models trained on large volumes of data can be prone to overfitting, where they memorize the training data but fail to generalize to new situations. Thus, to prevent drawing incorrect conclusions, it is essential to understand and analyze results, validate on huge data sets, and clean data.

Hence, selecting the right data from the vast volume of data is a crucial step and it determines the success of the finding through shaping the insight. Selecting relevant features that represent the important characteristics and variability of the data is a process that requires to involve domain expert to enable the target algorithm to properly identify patterns in the data.

3.3.1 Selecting Relevant Attributes

As clearly described in table 4.1, from the telecom CDR DB 20 columns are collected that contain different values including null and redundant values. In order to achieve the objective of this thesis work, the following relevant attributes are selected as the following for further processing.

Table 3. 2 Selected CDR Attributes

No	Attribute	Description
1	Service_Number	Mobile number that initiates the call.
2	Other_Number	Mobile number that received the initiated call.
3	START_TIME	Established call connection start time.
4	END_TIME	Established call connection end time.
5	CELL_ID	A unique identifier assigned to each cell in a cellular network.
6	CALL_DUR	Duration between call establishment start and end time.
7	DATA_DNLD_VOL	The amount of downloaded data volume
8	DATA_UPLD_VOL	The amount of uploaded data volume
9	DATA_ONLINE_DUR	Time period for how long the SIM stayed connected.
10	SMS_CALL_CNT	Text message Count for a particular service number
11	IMSI	International Mobile Subscriber Identity Number.
12	Total SMS	Total number of SMS sent by the subscriber.
13	Upstream_Data_Traffic	Amount of upload data to the internet.
14	DownStream_Data Traffic	Amount of download data from the internet.
15	SIM_CreatedDate	The date where the SIM is created.

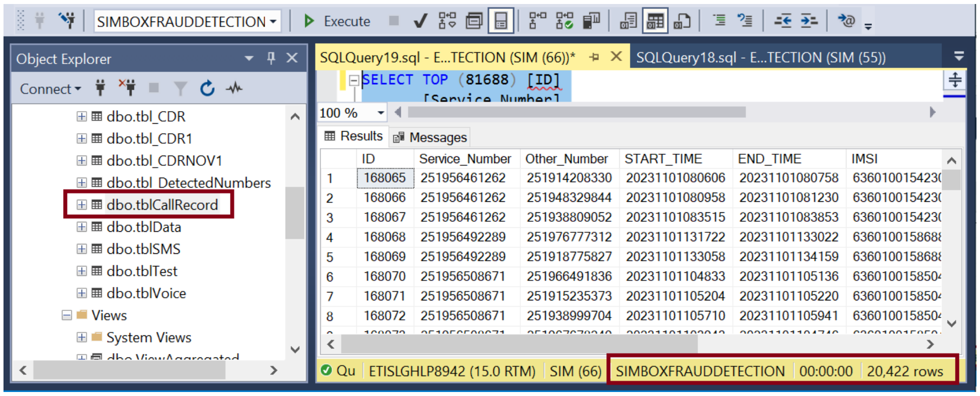
3.3.1 Manage Sampling Size

Sampling is a fundamental concept in data science that involves selecting a subset of data from a larger population for analysis. As we can see from figure 4.4, in the vast landscape of data, it is often impractical or resource-intensive to analyze the entire dataset. Sampling allows data scientists to draw meaningful insights and make inferences about the population based on a representative subset. There are various sampling techniques, each with their advantages and drawbacks, such as random sampling, stratified sampling, and cluster sampling. The choice of sampling method depends on the research objectives, available resources, and the nature of the data. Proper sampling is crucial to ensure that the subset accurately reflects the characteristics of the entire population, thereby enabling more efficient and cost-effective inference.

In this thesis work, we have used random sampling technique, due to all the population CDR has equal weight to the inference contribution and it ensures that every tuple of the CDR has an equal chance of being selected, minimizing bias, and ensuring the sample is truly representative. In order to determine the sample size of normal subscriber number and fraudulent number, there is no one-size-fits-all recommendation to manage the classification task and instead it depends on different criteria of complexity of the problem, number of features, desired level of accuracy and availability of the data. From the mentioned criteria's the complexity of the problem and desired level of accuracy is visible in this thesis work, due to the SIM-box detection task might be a bit complex because of the recorded behavior of fraudulent numbers are quite similar with normal subscribers in various attributes. In addition to this, the required level of accuracy is high and to deal with this situation larger sample dataset needs to be processed different from the work of [16], [17] and [18].

Therefore, fraudulent numbers are collected from Ethio telecom FMS, and using expert judgement, we have utilized here with the proportion of around 80/20 % normal/fraudulent subscriber ratio.

Normal Subscriber	81,688 records
Fraudulent Number	20,422 records
Total Record Base (Join table between CDR, Voice, SMS, and Data)	4,963,563 records

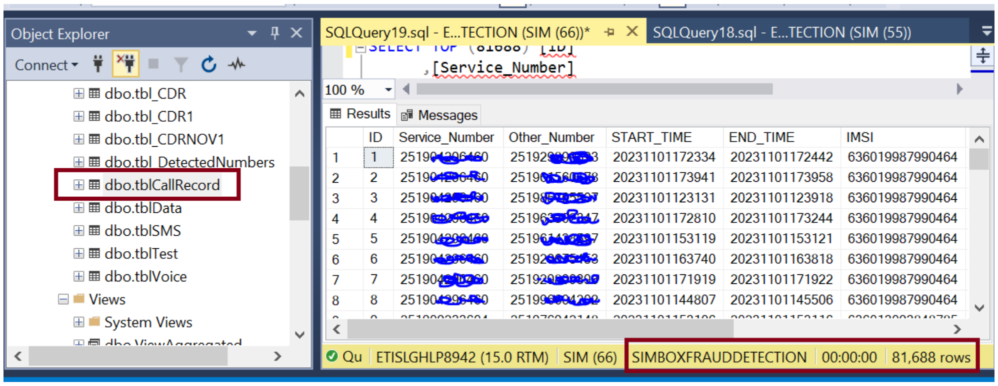


```
SELECT TOP (81688) [ID]
      [Service_Number]
```

	ID	Service_Number	Other_Number	START_TIME	END_TIME	IMSI
1	168065	251956461262	251914208330	20231101080606	20231101080758	636010015423
2	168066	251956461262	251948329844	20231101080958	20231101081230	636010015423
3	168067	251956461262	251938809052	20231101083515	20231101083853	636010015423
4	168068	251956492289	251976777312	20231101131722	20231101133022	636010015868
5	168069	251956492289	251918775827	20231101133058	20231101134159	636010015868
6	168070	251956508671	251966491836	20231101104833	20231101105136	636010015850
7	168071	251956508671	251915235373	20231101105204	20231101105220	636010015850
8	168072	251956508671	251938999704	20231101105710	20231101105941	636010015850

Qu | ETISLGHLP8942 (15.0 RTM) | SIM (66) | SIMBOXFRAUDDETECTION | 00:00:00 | 20,422 rows

Figure 3. 6 Sample Fraudulent Call Record

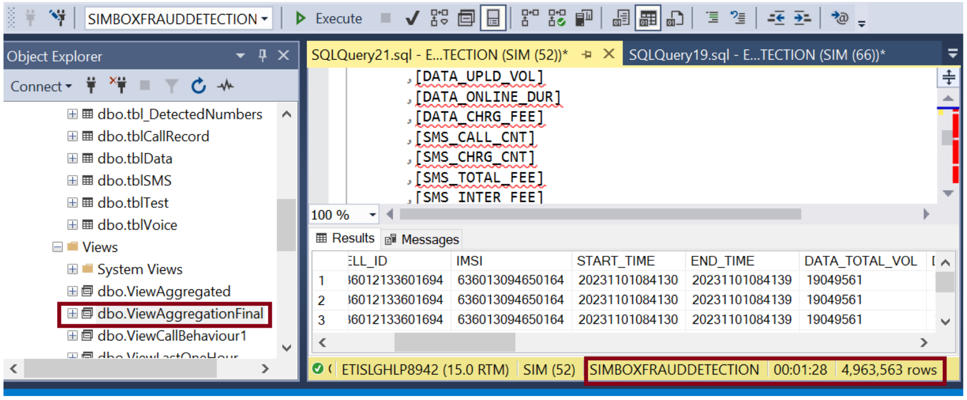


```
SELECT TOP (81688) [ID]
      ,[Service_Number]
```

	ID	Service_Number	Other_Number	START_TIME	END_TIME	IMSI
1	1	25190[redacted]0	25192[redacted]3	20231101172334	20231101172442	636019987990464
2	2	25190[redacted]0	25196[redacted]8	20231101173941	20231101173958	636019987990464
3	3	25190[redacted]0	25198[redacted]	20231101123131	20231101123918	636019987990464
4	4	2519[redacted]	25196[redacted]	20231101172810	20231101173244	636019987990464
5	5	25190[redacted]	25196[redacted]7	20231101153119	20231101153121	636019987990464
6	6	25190[redacted]0	25192[redacted]3	20231101163740	20231101163818	636019987990464
7	7	25190[redacted]0	25192[redacted]	20231101171919	20231101171922	636019987990464
8	8	25190[redacted]0	25199[redacted]	20231101144807	20231101145506	636019987990464

Qu | ETISLGHLP8942 (15.0 RTM) | SIM (66) | SIMBOXFRAUDDETECTION | 00:00:00 | 81,688 rows

Figure 3. 7 Sample Normal Subscriber Call Record



```
      ,[DATA_UPLD_VOL]
      ,[DATA_ONLINE_DUR]
      ,[DATA_CHRG_FEE]
      ,[SMS_CALL_CNT]
      ,[SMS_CHRG_CNT]
      ,[SMS_TOTAL_FEE]
      ,[SMS_INTER_FEE]
```

	ELL_ID	IMSI	START_TIME	END_TIME	DATA_TOTAL_VOL
1	36012133601694	636013094650164	20231101084130	20231101084139	19049561
2	36012133601694	636013094650164	20231101084130	20231101084139	19049561
3	36012133601694	636013094650164	20231101084130	20231101084139	19049561

ETISLGHLP8942 (15.0 RTM) | SIM (52) | SIMBOXFRAUDDETECTION | 00:01:28 | 4,963,563 rows

Figure 3. 8 ALL Aggregated Records

3.4 Data Preprocessing

Clear, useful data is essential for detecting SIM box fraud effectively and preprocessing this data is essential. The raw data, which is frequently Call Detail Records (CDRs), must first be cleaned up by eliminating duplicates, irregularities, and missing entries. Potential red flags, or outliers, are found and examined. The next step is feature engineering, which creates new features by modifying the existing ones. For instance, a SIM card's call frequency in relation to its age or unusual distribution of local and international call. Data normalization is one technique that makes sure features in fraud models contribute equally.

In order to avoid model bias, data balancing—which modifies the ratio of fraudulent to legit calls—might be required. This painstaking preprocessing, like enhancing a detective's toolkit, prepares models to identify the minute patterns that reveal SIM box fraudulent.

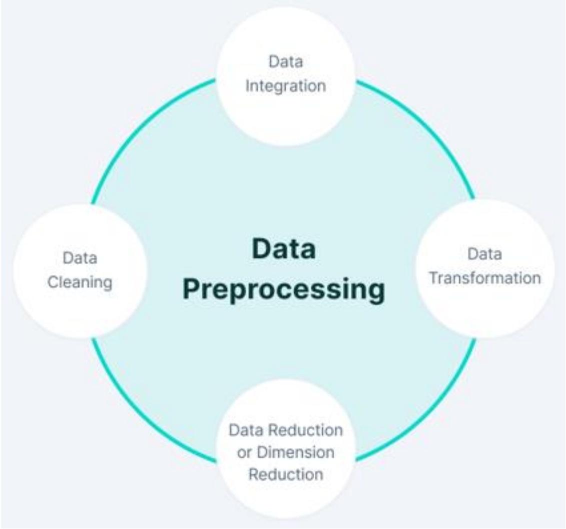
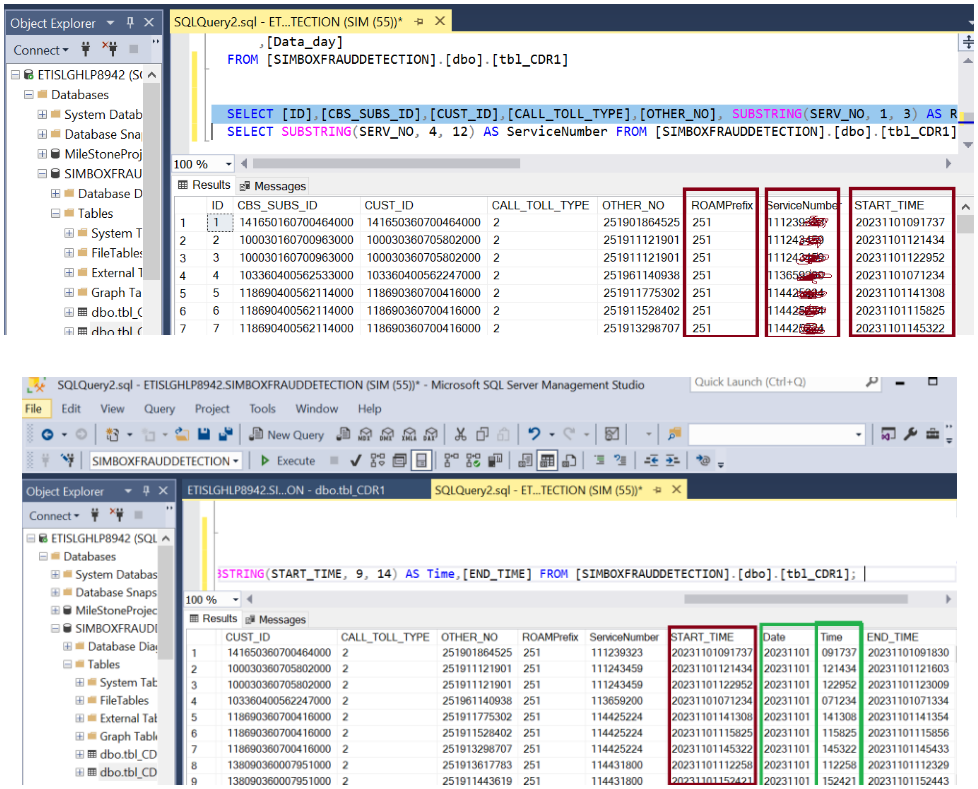


Figure 3. 9 Data Preprocessing in Machine Learning [20]

3.4.1 Data Cleaning

In order to build a viable model for detecting SIM box fraud, data cleansing is an essential first step in the process. Imagine sorting through a disorganized call record warehouse where duplicates hide in corners, inconsistent data confuses, and missing values shadowy patterns like smoke. Then the cleaner appears, eliminating these obstructions one by one. Missing values are carefully patched with well-informed imputations, duplicates are found and eliminated, and discrepancies are straightened up with corrections and standards. Potential red flags, or outliers, are marked for additional examination. After cleaning, the data shines, clear of noise and irregularities and ready to show the complex patterns that reveal the fraudulent secret tactics.

In this thesis, call detail records with missing or incorrect values, like a calling number length that doesn't have 12 characters are removed and adjusted accordingly. As we can see from figure 4.9, records that were not included in the country code (251) prefix are altered by appending a prefix to those records. Record timestamp substring task has done with “yyyy,dd,mm” and “h:m:s” format. Table 4.3 describes removed duplicate records and irrelevant features, due to retaining only one instance of each duplication and valuable columns makes the model effective. Additionally, the target data's validity and quality are examined in accordance with the planned data driven methodologies.



```
      ,[Data_day]
  FROM [SIMBOXFRAUDDETECTION].[dbo].[tbl_CDR1]


  SELECT [ID],[CBS_SUBS_ID],[CUST_ID],[CALL_TOLL_TYPE],[OTHER_NO], SUBSTRING(SERV_NO, 1, 3) AS R
  SELECT SUBSTRING(SERV_NO, 4, 12) AS ServiceNumber FROM [SIMBOXFRAUDDETECTION].[dbo].[tbl_CDR1]
```

	ID	CBS_SUBS_ID	CUST_ID	CALL_TOLL_TYPE	OTHER_NO	ROAMPrefix	ServiceNumbe	START_TIME
1	1	141650160700464000	141650360700464000	2	251901864525	251	111239[redacted]	20231101091737
2	2	100030160700963000	100030360705802000	2	251911121901	251	111243[redacted]	20231101121434
3	3	100030160700963000	100030360705802000	2	251911121901	251	111243[redacted]	20231101122952
4	4	103360400562533000	103360400562247000	2	251961140938	251	113659[redacted]	20231101071234
5	5	118690400562114000	118690360700416000	2	251911775302	251	114425[redacted]	20231101141308
6	6	118690400562114000	118690360700416000	2	251911528402	251	114425[redacted]	20231101115825
7	7	118690400562114000	118690360700416000	2	251913298707	251	114425[redacted]	20231101145322

```
3STRING(START_TIME, 9, 14) AS Time,[END_TIME] FROM [SIMBOXFRAUDDETECTION].[dbo].[tbl_CDR1];
```

	CUST_ID	CALL_TOLL_TYPE	OTHER_NO	ROAMPrefix	ServiceNumber	START_TIME	Date	Time	END_TIME
1	141650360700464000	2	251901864525	251	111239323	20231101091737	20231101	091737	20231101091830
2	100030360705802000	2	251911121901	251	111243459	20231101121434	20231101	121434	20231101121603
3	100030360705802000	2	251911121901	251	111243459	20231101122952	20231101	122952	20231101123009
4	103360400562247000	2	251961140938	251	113659200	20231101071234	20231101	071234	20231101071334
5	118690360700416000	2	251911775302	251	114425224	20231101141308	20231101	141308	20231101141354
6	118690360700416000	2	251911528402	251	114425224	20231101115825	20231101	115825	20231101115856
7	118690360700416000	2	251913298707	251	114425224	20231101145322	20231101	145322	20231101145433
8	138090360007951000	2	251913617783	251	114431800	20231101112258	20231101	112258	20231101112329
9	138090360007951000	2	251911443619	251	114431800	20231101152421	20231101	152421	20231101152443

Figure 3. 10 Data Cleansing Example

Table 3. 3 Removed Columns

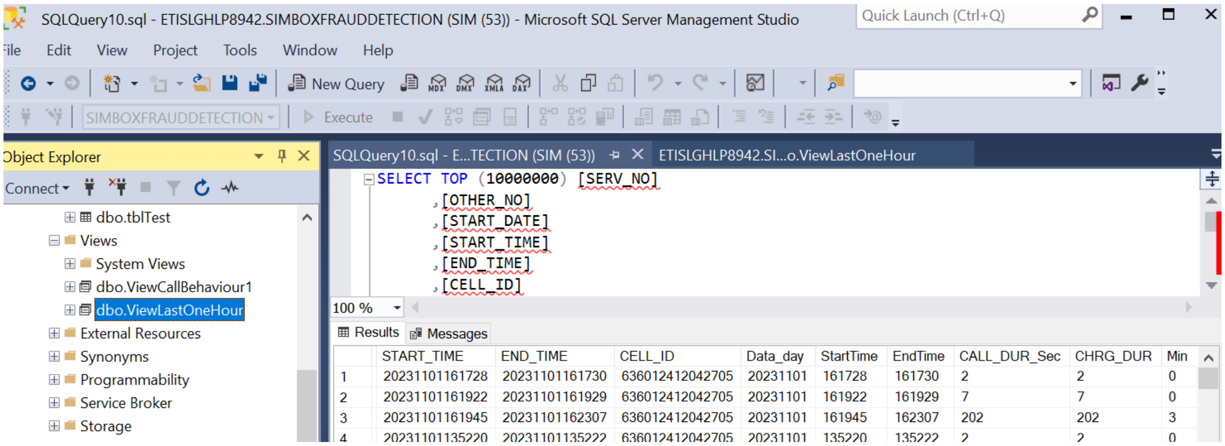
No	Removed Columns				
1	CNTRY_CODE	4	VOC_VIDEO_CALL_DUR	7	DATA_UPLD_FEE
2	ROAM_CNTRY_CODE	5	VOC_DATA_CALL_DUR	8	VOC_DATA_CHRG_DUR
3	MSC_ADDR	6	DATA_DLND_FEE		

After data cleansing, Call detail record (CDR) data aggregation plays a crucial role in detecting SIM box fraud, an illegal operation involving rerouting international calls for profit. By combining CDRs over time, location, and user behavior, telecom providers might uncover suspicious patterns indicative of fraud. A few warning signs identified by aggregate include daily call volumes significantly higher than average, abrupt increases in international calls made by individuals, or concentrated calling behavior towards known SIM box locations. This comprehensive data driven approach gives fraud detection systems the ability to identify fraudulent activities and act quickly to prevent revenue losses and illegal use of network resources. In this thesis, we used data aggregation parameters of time in 1-hour bases, Cell_ID and call behaviors to maximize the sense of near real-time and increase detection capabilities.

Table 3. 4 Derived Attributes for Call Behavior Identification

Aggregation Parameters	**Derived Attributes (Call Behaviors)**	**Description**
*Service Number *Last 1-Hour *Cell_ID	Total_Call_Made	Total number of calls made with in 1 hour, and Cell_ID.
	Total_Call_Made_toUnique_Num	Total number of outgoing calls to distinct numbers.
	Total_Incoming_Calls	Total number of incoming calls.
	Total_SMS_Local_Count	Total Number of SMS sent and received.
	Total_SMS_Inter_Count	Total Number of SMS sent and received from international network.
	Total_Data_Volume	Total amount of data upload and download.
	Total_Incoming_Duration	Total outgoing call duration in second within 1-hour and particular Cell_ID.
	Total_Outgoing_Duration	Total incoming call duration in second within 1-hour and particular Cell_ID.
	Total_Outgoing_Call_PEAK	Call established during daytime
	Total_Outgoing_Call_OFFPEAK	Call established during night
	Average_Call_Duration	Average time for established call

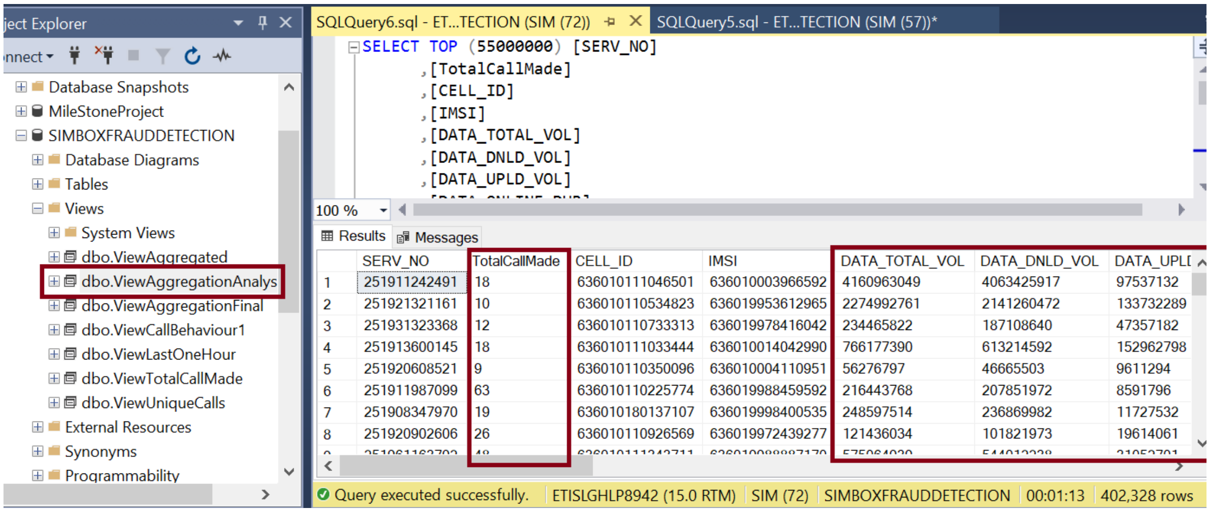
Table 4.3 describes, how the SIM-box detection model analyzes call patterns from the available call details record database and identifies possible fraudulent SIMs by going through call, SMS, and Data utilization activities with the considerations of fraudulent numbers initiate calls from the same location, made many short duration calls, less amount of SMS, Data usage and a smaller number of local incoming calls.



```
SELECT TOP (10000000) [SERV_NO]
      ,[OTHER_NO]
      ,[START_DATE]
      ,[START_TIME]
      ,[END_TIME]
      ,[CELL_ID]
```

	START_TIME	END_TIME	CELL_ID	Data_day	StartTime	EndTime	CALL_DUR_Sec	CHRG_DUR	Min
1	20231101161728	20231101161730	636012412042705	20231101	161728	161730	2	2	0
2	20231101161922	20231101161929	636012412042705	20231101	161922	161929	7	7	0
3	20231101161945	20231101162307	636012412042705	20231101	161945	162307	202	202	3
4	20231101135220	20231101135222	636012412042705	20231101	135220	135222	2	2	0

Figure 3. 11 Database View for Last One hour CDR

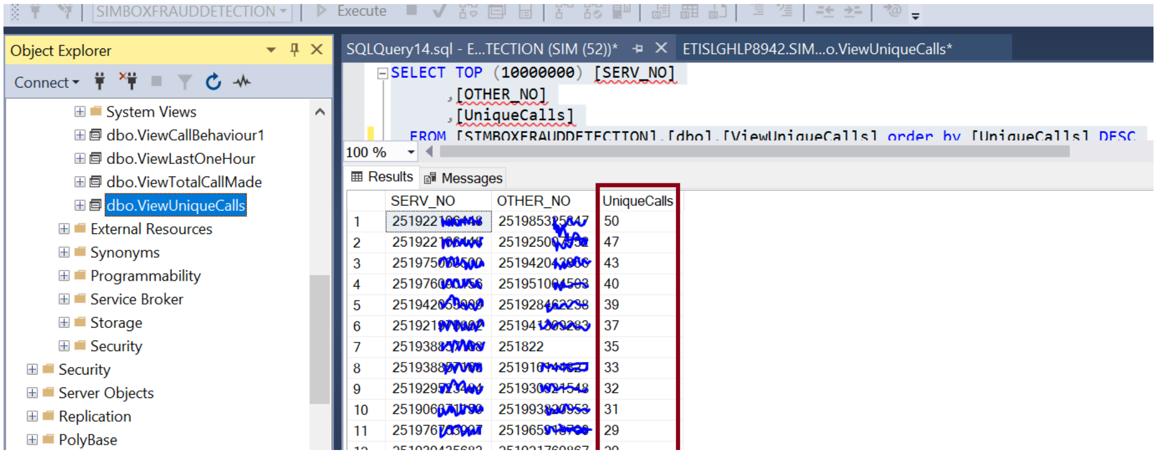


```
SELECT TOP (55000000) [SERV_NO]
      ,[TotalCallMade]
      ,[CELL_ID]
      ,[IMSI]
      ,[DATA_TOTAL_VOL]
      ,[DATA_DNLD_VOL]
      ,[DATA_UPLD_VOL]
```

	SERV_NO	TotalCallMade	CELL_ID	IMSI	DATA_TOTAL_VOL	DATA_DNLD_VOL	DATA_UPLI
1	251911242491	18	636010111046501	636010003966592	4160963049	4063425917	97537132
2	251921321161	10	636010110534823	636019953612965	2274992761	2141260472	133732289
3	251931323368	12	636010110733313	636019978416042	234465822	187108640	47357182
4	251913600145	18	636010111033444	636010014042990	766177390	613214592	152962798
5	251920608521	9	636010110350096	636010004110951	56276797	46665503	9611294
6	251911987099	63	636010110225774	636019988459592	216443768	207851972	8591796
7	251908347970	19	636010180137107	636019998400535	248597514	236869982	11727532
8	251920902606	26	636010110926569	636019972439277	121436034	101821973	19614061

```
Query executed successfully. | ETISLGHLP8942 (15.0 RTM) | SIM (72) | SIMBOXFRAUDDETECTION | 00:01:13 | 402,328 rows
```

Figure 3. 12 View Aggregated Data Analysis Per Subscriber



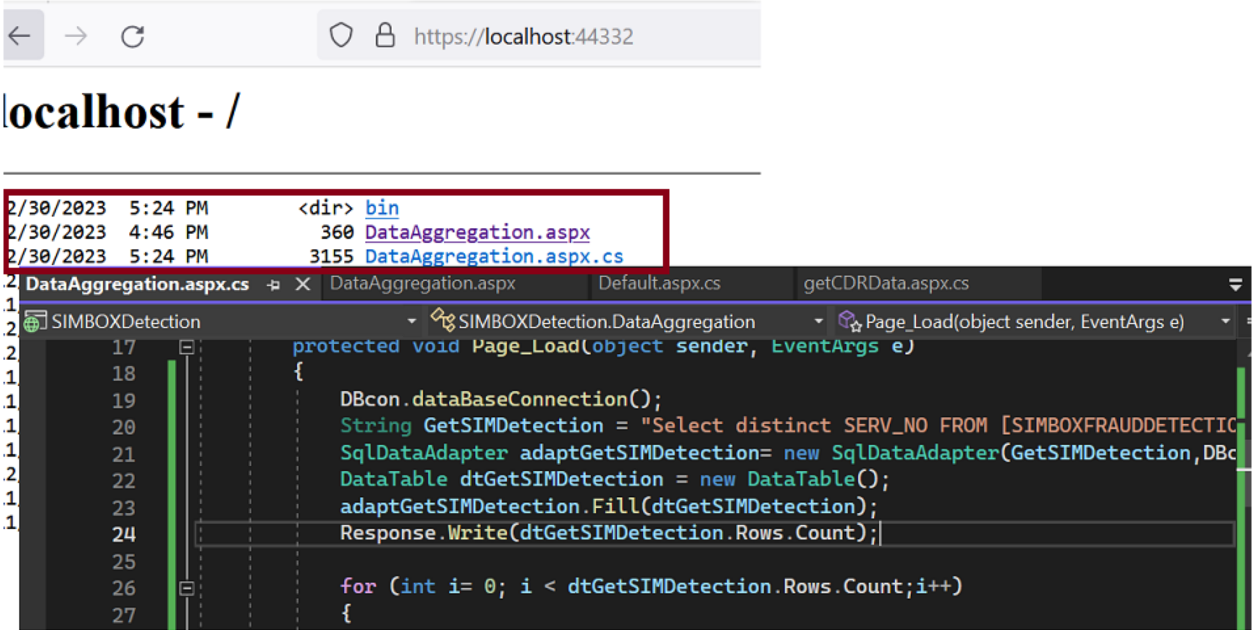
```sql
SELECT TOP (10000000) [SERV_NO]
      ,[OTHER_NO]
      ,[UniqueCalls]
  FROM [SIMBOXFRAUDDETECTION].[dbo].[ViewUniqueCalls] order by [UniqueCalls] DESC
```

	SERV_NO	OTHER_NO	UniqueCalls
1	251922[redacted]	2519853[redacted]	50
2	251922[redacted]	2519250[redacted]	47
3	251975[redacted]	2519420[redacted]	43
4	251976[redacted]	2519510[redacted]	40
5	251942[redacted]	251928[redacted]	39
6	251921[redacted]	25194[redacted]	37
7	251938[redacted]	251822	35
8	251938[redacted]	251916[redacted]	33
9	251929[redacted]	251930[redacted]	32
10	251906[redacted]	251993[redacted]	31
11	251976[redacted]	251965[redacted]	29

Figure 3. 13 DB View for Unique Calls Per Subscriber

3.4.2 Data Integration

As shown from Figure 4.10, 4.11 and 4.12, we have prepared database views using structural query language (SQL) to perform data aggregation on the collected CDR data, that compiles and display total call made and total unique call made by the available service number. In addition to this, as shown in Figure 4.13 and 4.14 we have developed a function called DataAggregation.cs using ASP.net C# to get total incoming calls, total SMS, total Data, total incoming duration, and total outgoing duration from different data models of CDR database.

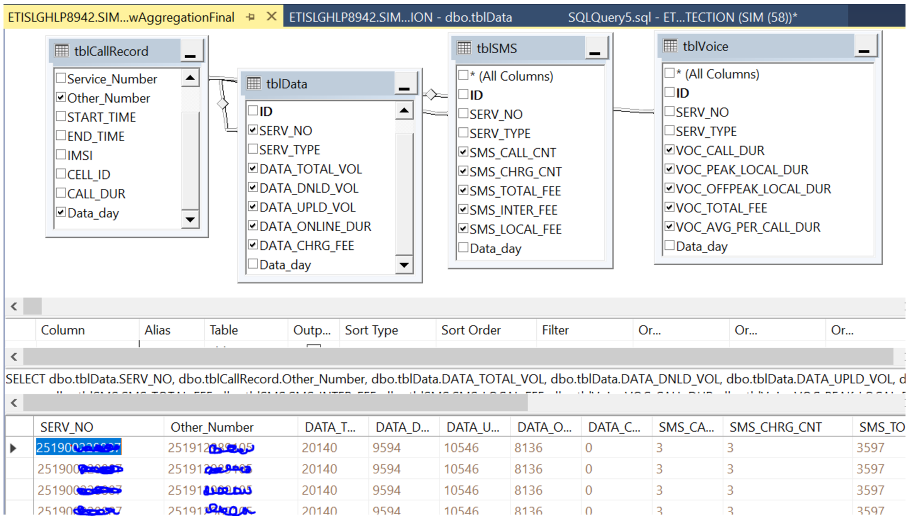


localhost - /

```
2/30/2023   5:24 PM        <dir> bin
2/30/2023   4:46 PM          360 DataAggregation.aspx
2/30/2023   5:24 PM         3155 DataAggregation.aspx.cs
```

```csharp
        protected void Page_Load(object sender, EventArgs e)
        {
            DBcon.dataBaseConnection();
            String GetSIMDetection = "Select distinct SERV_NO FROM [SIMBOXFRAUDDETECTIC
            SqlDataAdapter adaptGetSIMDetection= new SqlDataAdapter(GetSIMDetection,DBc
            DataTable dtGetSIMDetection = new DataTable();
            adaptGetSIMDetection.Fill(dtGetSIMDetection);
            Response.Write(dtGetSIMDetection.Rows.Count);

            for (int i= 0; i < dtGetSIMDetection.Rows.Count;i++)
            {
```

Figure 3. 14 Data Aggregation Function


ETISLGHLP8942.SIM...wAggregationFinal
ETISLGHLP8942.SIM...ION - dbo.tblData
SQLQuery5.sql - ET...TECTION (SIM (58))*

tblCallRecord
☐ Service_Number
☑ Other_Number
☐ START_TIME
☐ END_TIME
☐ IMSI
☐ CELL_ID
☐ CALL_DUR
☑ Data_day

tblData
☐ ID
☑ SERV_NO
☐ SERV_TYPE
☑ DATA_TOTAL_VOL
☑ DATA_DNLD_VOL
☑ DATA_UPLD_VOL
☑ DATA_ONLINE_DUR
☑ DATA_CHRG_FEE
☐ Data_day

tblSMS
☐ * (All Columns)
☐ ID
☐ SERV_NO
☐ SERV_TYPE
☑ SMS_CALL_CNT
☑ SMS_CHRG_CNT
☑ SMS_TOTAL_FEE
☑ SMS_INTER_FEE
☑ SMS_LOCAL_FEE
☐ Data_day

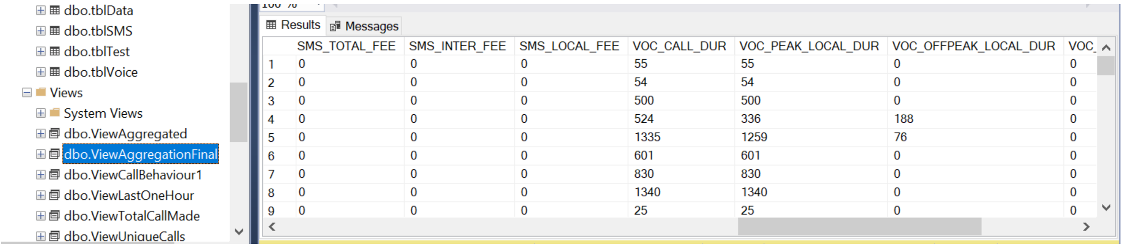
tblVoice
☐ * (All Columns)
☐ ID
☐ SERV_NO
☐ SERV_TYPE
☑ VOC_CALL_DUR
☑ VOC_PEAK_LOCAL_DUR
☑ VOC_OFFPEAK_LOCAL_DUR
☑ VOC_TOTAL_FEE
☑ VOC_AVG_PER_CALL_DUR
☐ Data_day

Column	Alias	Table	Outp...	Sort Type	Sort Order	Filter	Or...	Or...	Or...

```
SELECT dbo.tblData.SERV_NO, dbo.tblCallRecord.Other_Number, dbo.tblData.DATA_TOTAL_VOL, dbo.tblData.DATA_DNLD_VOL, dbo.tblData.DATA_UPLD_VOL, d
```

SERV_NO	Other_Number	DATA_T...	DATA_D...	DATA_U...	DATA_O...	DATA_C...	SMS_CA...	SMS_CHRG_CNT	SMS_TO
251900[redacted]	251912[redacted]	20140	9594	10546	8136	0	3	3	3597
251900[redacted]	251912[redacted]	20140	9594	10546	8136	0	3	3	3597
251900[redacted]	251912[redacted]	20140	9594	10546	8136	0	3	3	3597
25190[redacted]	25191[redacted]	20140	9594	10546	8136	0	3	3	3597

Figure 3. 15 Aggregated CDR Data



	SMS_TOTAL_FEE	SMS_INTER_FEE	SMS_LOCAL_FEE	VOC_CALL_DUR	VOC_PEAK_LOCAL_DUR	VOC_OFFPEAK_LOCAL_DUR	VOC_
1	0	0	0	55	55	0	0
2	0	0	0	54	54	0	0
3	0	0	0	500	500	0	0
4	0	0	0	524	336	188	0
5	0	0	0	1335	1259	76	0
6	0	0	0	601	601	0	0
7	0	0	0	830	830	0	0
8	0	0	0	1340	1340	0	0
9	0	0	0	25	25	0	0

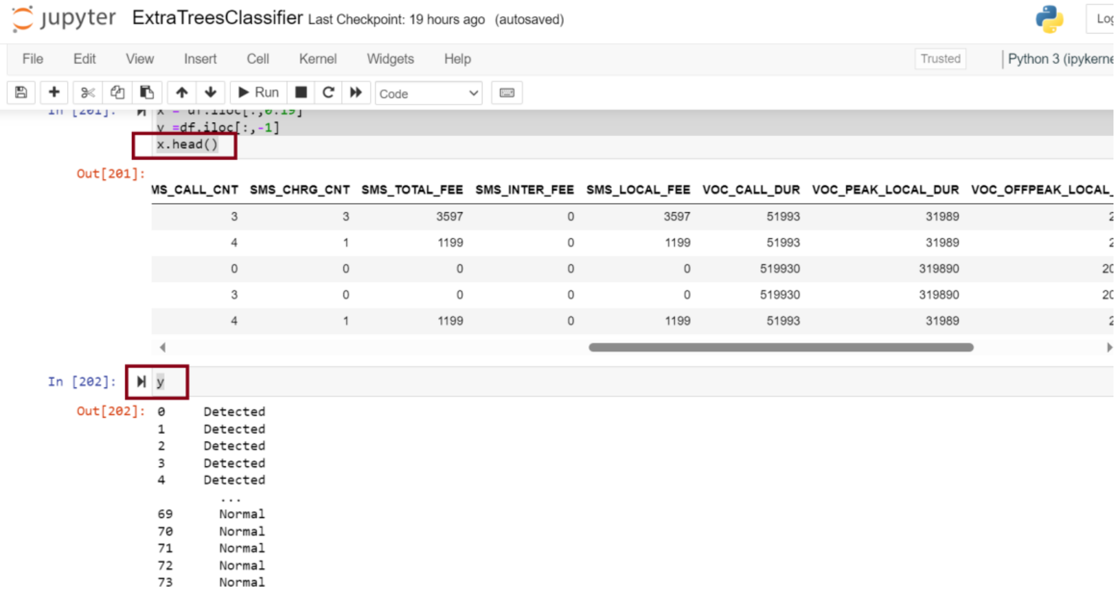
3.4.3 Feature Selection

Feature selection is a critical step in machine learning, particularly in the realm of SIM box fraud detection. It involves choosing the most relevant features from your dataset that contribute significantly to predicting SIM box fraud, while removing redundant or irrelevant ones, and this could increase the proposed fraud detection model's effectiveness and performance. There are different feature selection methodologies are available that identified namely as filter method, wrapper method, and embedded method [21].

Filter methods rely on statistical measures to assess the individual relevance of each feature to the target variable (in this case, SIM box fraud). They are computationally efficient and easy to implement but may fail to capture complex relationships between features. Common filter methods include correlation coefficients, mutual information, and variance threshold. Wrapper method evaluates different subsets of features by training a machine learning model on each subset and choosing the one that performs best on a validation set.

They are more flexible and can identify complex interactions between features but can be computationally expensive and prone to overfitting.

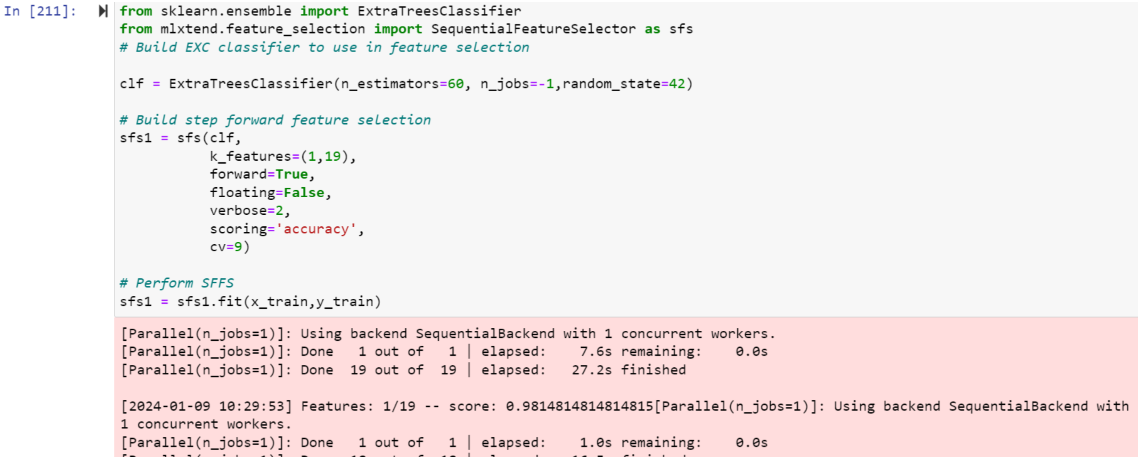
In the context of this thesis, we have used embedded method of ExtraTreesClassifier algorithm for feature selection due to its exceptional combination of accuracy, efficiency, interpretability, and resilience makes it stand out as an effective option in machine learning. It solves complicated tasks with an ensemble of randomized decision trees and provides insightful information about the significance of features [22].



jupyter ExtraTreesClassifier Last Checkpoint: 19 hours ago (autosaved)
Log
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykerne
Run
Code

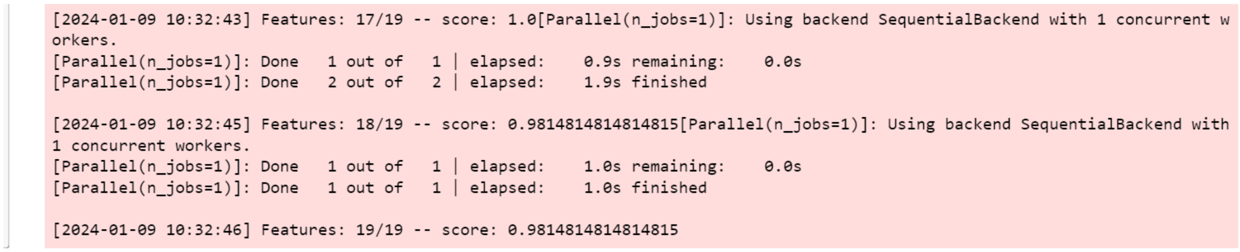
```
In [201]:  x = df.iloc[:,0:19]
           y =df.iloc[:,-1]
           x.head()

Out[201]:
MS_CALL_CNT  SMS_CHRG_CNT  SMS_TOTAL_FEE  SMS_INTER_FEE  SMS_LOCAL_FEE  VOC_CALL_DUR  VOC_PEAK_LOCAL_DUR  VOC_OFFPEAK_LOCAL_
          3             3           3597              0           3597         51993               31989                   2
          4             1           1199              0           1199         51993               31989                   2
          0             0              0              0              0        519930              319890                  20
          3             0              0              0              0        519930              319890                  20
          4             1           1199              0           1199         51993               31989                   2

In [202]:  y

Out[202]: 0     Detected
          1     Detected
          2     Detected
          3     Detected
          4     Detected
                  ...
          69      Normal
          70      Normal
          71      Normal
          72      Normal
          73      Normal
```

Figure 3. 16 Feature Selection Data Set

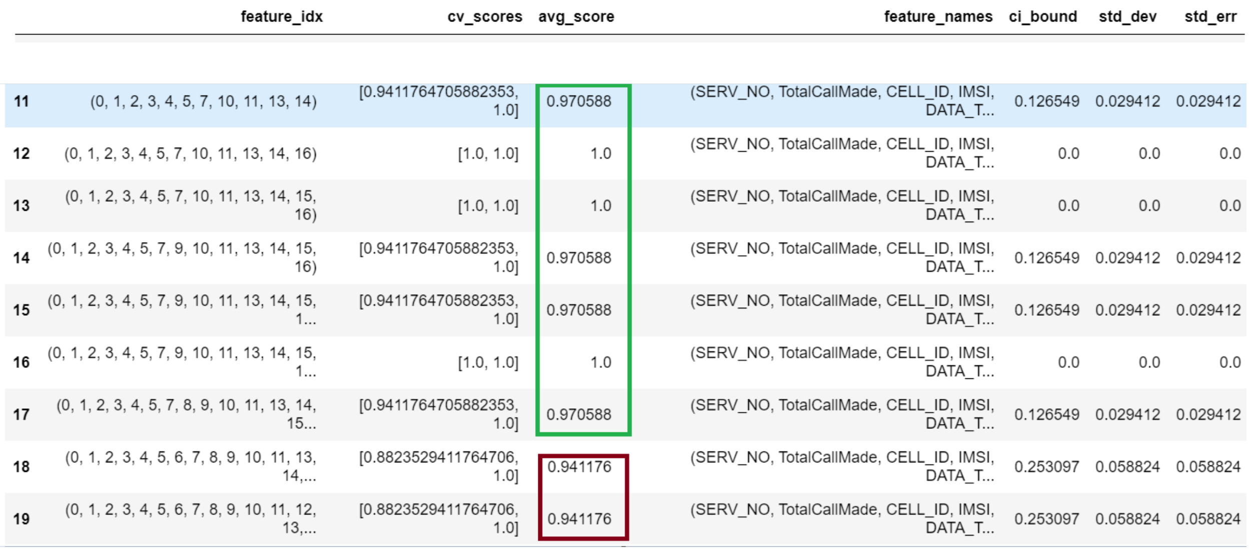


```
In [211]:  from sklearn.ensemble import ExtraTreesClassifier
           from mlxtend.feature_selection import SequentialFeatureSelector as sfs
           # Build EXC classifier to use in feature selection

           clf = ExtraTreesClassifier(n_estimators=60, n_jobs=-1,random_state=42)

           # Build step forward feature selection
           sfs1 = sfs(clf,
                      k_features=(1,19),
                      forward=True,
                      floating=False,
                      verbose=2,
                      scoring='accuracy',
                      cv=9)

           # Perform SFFS
           sfs1 = sfs1.fit(x_train,y_train)

           [Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
           [Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    7.6s remaining:    0.0s
           [Parallel(n_jobs=1)]: Done  19 out of  19 | elapsed:   27.2s finished

           [2024-01-09 10:29:53] Features: 1/19 -- score: 0.9814814814814815[Parallel(n_jobs=1)]: Using backend SequentialBackend with
           1 concurrent workers.
           [Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    1.0s remaining:    0.0s
```

Figure 3. 17 ExtraTreesClassifier Algorithm

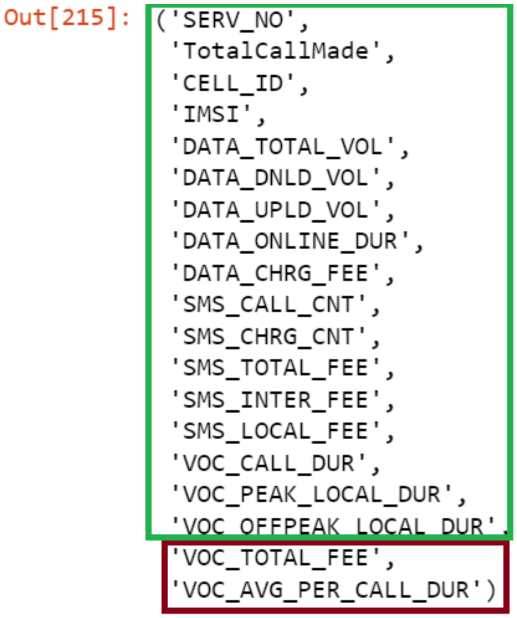


```
[2024-01-09 10:32:43] Features: 17/19 -- score: 1.0[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent w
orkers.
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    0.9s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   2 out of   2 | elapsed:    1.9s finished

[2024-01-09 10:32:45] Features: 18/19 -- score: 0.9814814814814815[Parallel(n_jobs=1)]: Using backend SequentialBackend with
1 concurrent workers.
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    1.0s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    1.0s finished

[2024-01-09 10:32:46] Features: 19/19 -- score: 0.9814814814814815
```



	feature_idx	cv_scores	avg_score	feature_names	ci_bound	std_dev	std_err
11	(0, 1, 2, 3, 4, 5, 7, 10, 11, 13, 14)	[0.9411764705882353, 1.0]	0.970588	(SERV_NO, TotalCallMade, CELL_ID, IMSI, DATA_T...	0.126549	0.029412	0.029412
12	(0, 1, 2, 3, 4, 5, 7, 10, 11, 13, 14, 16)	[1.0, 1.0]	1.0	(SERV_NO, TotalCallMade, CELL_ID, IMSI, DATA_T...	0.0	0.0	0.0
13	(0, 1, 2, 3, 4, 5, 7, 10, 11, 13, 14, 15, 16)	[1.0, 1.0]	1.0	(SERV_NO, TotalCallMade, CELL_ID, IMSI, DATA_T...	0.0	0.0	0.0
14	(0, 1, 2, 3, 4, 5, 7, 9, 10, 11, 13, 14, 15, 16)	[0.9411764705882353, 1.0]	0.970588	(SERV_NO, TotalCallMade, CELL_ID, IMSI, DATA_T...	0.126549	0.029412	0.029412
15	(0, 1, 2, 3, 4, 5, 7, 9, 10, 11, 13, 14, 15, 1...	[0.9411764705882353, 1.0]	0.970588	(SERV_NO, TotalCallMade, CELL_ID, IMSI, DATA_T...	0.126549	0.029412	0.029412
16	(0, 1, 2, 3, 4, 5, 7, 9, 10, 11, 13, 14, 15, 1...	[1.0, 1.0]	1.0	(SERV_NO, TotalCallMade, CELL_ID, IMSI, DATA_T...	0.0	0.0	0.0
17	(0, 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 13, 14, 15...	[0.9411764705882353, 1.0]	0.970588	(SERV_NO, TotalCallMade, CELL_ID, IMSI, DATA_T...	0.126549	0.029412	0.029412
18	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14,...	[0.8823529411764706, 1.0]	0.941176	(SERV_NO, TotalCallMade, CELL_ID, IMSI, DATA_T...	0.253097	0.058824	0.058824
19	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,...	[0.8823529411764706, 1.0]	0.941176	(SERV_NO, TotalCallMade, CELL_ID, IMSI, DATA_T...	0.253097	0.058824	0.058824

Figure 3. 18 Feature Selection Scores



```
Out[215]: ('SERV_NO',
           'TotalCallMade',
           'CELL_ID',
           'IMSI',
           'DATA_TOTAL_VOL',
           'DATA_DNLD_VOL',
           'DATA_UPLD_VOL',
           'DATA_ONLINE_DUR',
           'DATA_CHRG_FEE',
           'SMS_CALL_CNT',
           'SMS_CHRG_CNT',
           'SMS_TOTAL_FEE',
           'SMS_INTER_FEE',
           'SMS_LOCAL_FEE',
           'VOC_CALL_DUR',
           'VOC_PEAK_LOCAL_DUR',
           'VOC_OFFPEAK_LOCAL_DUR',
           'VOC_TOTAL_FEE',
           'VOC_AVG_PER_CALL_DUR')
```

Figure 3. 19 Selected Features for Model Development

Chapter Four

Model Development and Evaluation

This chapter discusses about model development and evaluation for near real-time SIM-box fraud detection problem. The identified problem is categorized as a classification problem and in this chapter, we will experiment with different classification model development algorithm of random forest (RF), support vector machine (SVM), and neural network (NN). In order to achieve this, we have applied much effort on the dataset management of CDR data collection, preprocessing, integrating different data from various sources, perform SQL analysis and feature selection (described in chapter 4).

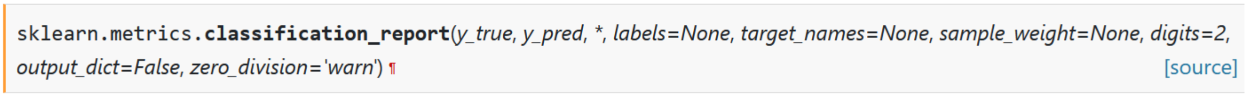
For the model development and evaluation task, we have utilized Scikit-learn (sklearn), it is a powerful and popular Python library for building and utilizing machine learning models. It provides a comprehensive set of algorithms, tools, and utilities for tasks like data preprocessing, feature engineering, model selection, and training, model evaluation and analysis, model deployment and load into production.

4.1 Evaluation Methods

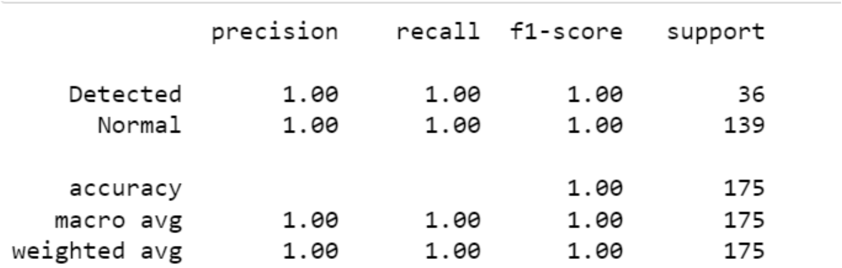
Scikit-learn, often abbreviated as sklearn, is a widely used machine learning library in Python, providing a comprehensive set of tools for building and evaluating machine learning models. Within sklearn, metrics play a crucial role in assessing the performance of models across various tasks, including classification, regression, and clustering. These metrics serve as quantitative measures that help practitioners gauge the effectiveness of their models by comparing predicted outcomes to true values. Common metrics in sklearn include accuracy, precision, recall, F1 score, and mean squared error, among others. These metrics aid in making informed decisions about model selection, hyperparameter tuning, and overall model optimization.

The ***sklearn*** metrics library also includes functions for generating confusion matrices, computing area under the receiver operating characteristic curve (AUC-ROC) and assessing clustering quality. Scikit-learns metrics play a crucial role in helping researchers and practitioners understand and improve the effectiveness of their model using the function classification_report.

In scikit-learn, the ***classification_report*** function is used to generate a text report showing the main classification metrics for a classification model. It is particularly useful for evaluating the performance of a classification algorithm by providing information such as precision, recall, f1-score, and support for each class.



```
sklearn.metrics.classification_report(y_true, y_pred, *, labels=None, target_names=None, sample_weight=None, digits=2,
output_dict=False, zero_division='warn') ¶                                                                    [source]
```



```
              precision    recall  f1-score   support

    Detected       1.00      1.00      1.00        36
      Normal       1.00      1.00      1.00       139

    accuracy                           1.00       175
   macro avg       1.00      1.00      1.00       175
weighted avg       1.00      1.00      1.00       175
```

Precision is one of the metrics included in the classification report, and it measures the accuracy of the positive predictions made by the model. It is calculated as the ratio of true positive predictions to the sum of true positives and false positives. The formula for precision is:

$$\text{Precission} = \frac{\text{True Postives (TR)}}{\text{True Postives (TR)+ False Postive (FP)}} \qquad 5.1$$

Recall is one of the metrics included in this report. Recall, also known as sensitivity or true positive rate, measures the ability of a classifier to correctly identify all relevant instances, or in other words, it's the ratio of true positive predictions to the total number of actual positive instances.

$$\text{Recall} = \frac{\text{True Postives (TR)}}{\text{True Postives (TR)+ False Negative (FN)}} \qquad 5.2$$

F1-score is the mean of precision and recall and is often used as a single metric to evaluate a classifier's performance.

$$\mathrm{F1-score} = 2 * (\mathrm{Precision} * \mathrm{Recall}) / (\mathrm{Precision} + \mathrm{Recall}) \quad 5.3$$

In sklearn, **accuracy** is represented in **accuracy_score** function that used to evaluate the accuracy of a classification model. It compares the predicted labels to the true labels and calculates the accuracy as the ratio of correctly predicted instances to the total number of instances.

$$\text{Accuracy} = \frac{\text{True Postivess (TP)} + \text{True Negaives (TN)}}{\text{True Postivess (TP)} + \text{True Negaives (TN)} + \text{False Negative (FN)} + \text{False Postive (FP)}} \quad 5.4$$

4.2 Model Development

Model development process involves problem definition, identify the type of data that shall be consumed in the process, perform data preprocessing, and feature selection that we discussed in chapter 1 and 4 of this thesis work. In this chapter we will discuss model selection, evaluation, and other required techniques to achieve better SIM-box fraud detection.

Model selection in machine learning is a critical step that involves choosing the most appropriate algorithm or model architecture for a given task. The goal is to identify the model that will generalize well to unseen data and provide accurate predictions. Model selection is a balancing act between underfitting and overfitting. Underfit models may fail to capture the complexities of the data, while overfit models may memorize the training data and perform poorly on new instances. Researchers and practitioners typically explore a range of algorithms and architectures, considering factors such as the complexity of the model, the size and nature of the dataset, computational resources, and the desired interpretability of the model.

The goal of this thesis work is to identify and develop the best fit model for SIM-box fraud detection operation that can detect possible fraudulent service numbers in near real-time manner using the available machine learning classification technique or custom solution that best fit to the problem domain. Therefore, we will experiment random forest, support vector machine, and neural network (NN) algorithms on the available preprocessed datasets using python. Depending on the classification report we will develop a custom model that can learn the preprocessed datasets pattern and identify SIM-box fraudulent activities using ASP.net C#.

Hence, we will utilize a python library called ***sklearn.model_selection*** that offers tools for model selection and evaluation. This module includes functions for ***train_test_split*** and ***cross_val_score.***

The ***train_test_split*** function is used to split a dataset into training and testing sets and its essential to evaluate the performance of a model on unseen data. The ***cross_val_score (***k-fold CV***)*** evaluates a model's performance and reduces the possibility of overfitting. The process entails dividing the dataset into several subsets, using part of these subsets to train the model, and then assessing it using the remaining subset. The final evaluation metric is the average performance after this process is conducted several times with various splits.

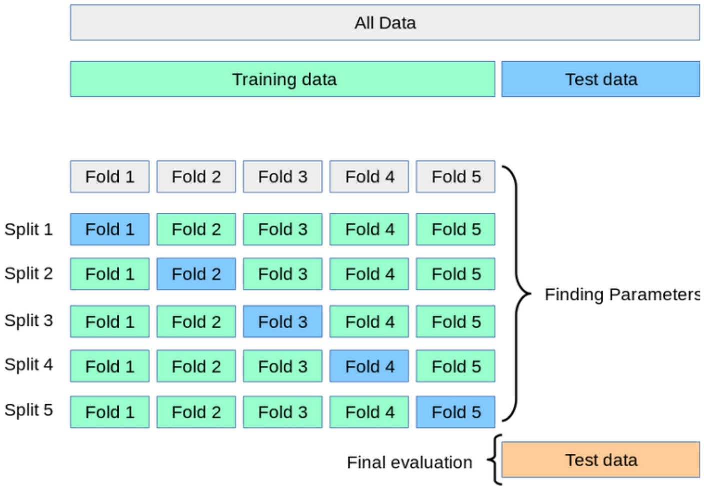


Figure 4. 1 k-fold cross-validation [23]

Hence, in this thesis experimentation, we have applied cross-validation techniques with various parameters and iterations to bring optimal performance and avoid overfitting.

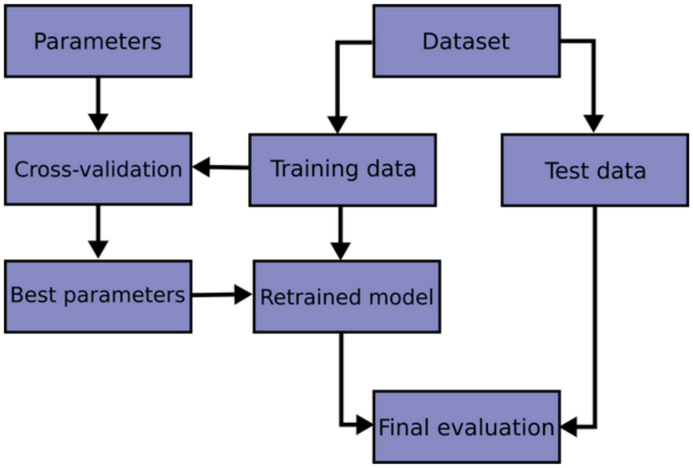


Figure 4. 2 Cross-validation: evaluating estimator performance [23].

4.2.1 Dataset Categorization

Before feeding the dataset into the machine learning algorithm, it's preferable to split the data to hourly bases to have the sense of near real-time detection. But for the purpose of developing a better model we will run the algorithm with different dataset groups for 1hour, 1day and 7days. Once we have identified the best performance model, it will be deployed to the production server, will process 1-hour CDR data iteratively and classify service numbers as ***normal*** and ***detected*** to indicate further procedural action for Ethio-telecom.

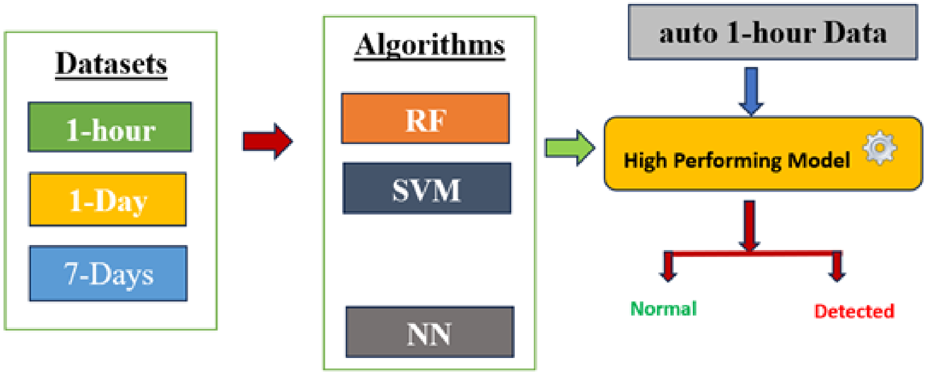
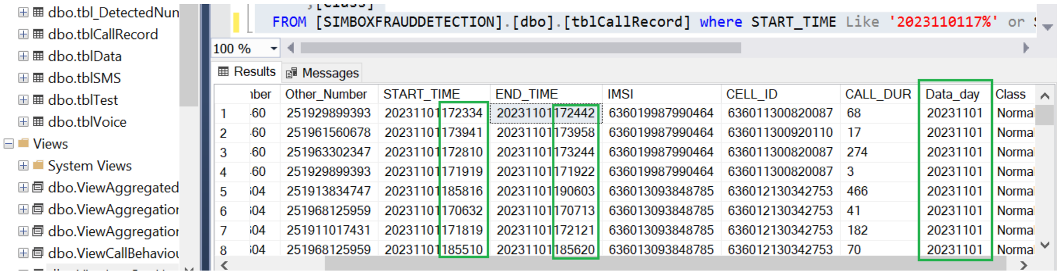


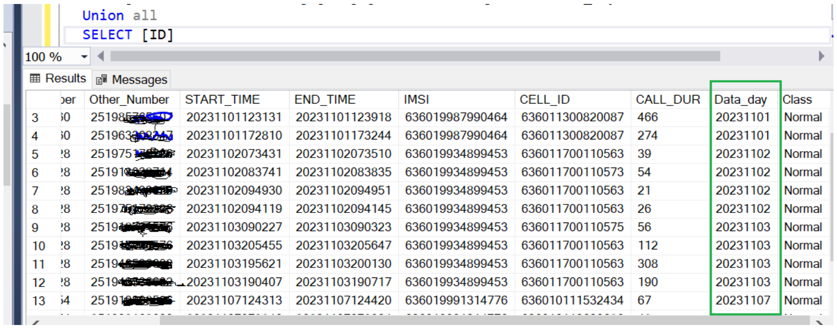
Figure 4. 3 Model Selection



```
FROM [SIMBOXFRAUDDETECTION].[dbo].[tblCallRecord] where START_TIME Like '2023110117%' or
```

	ıber	Other_Number	START_TIME	END_TIME	IMSI	CELL_ID	CALL_DUR	Data_day	Class
1	60	251929899393	20231101172334	20231101172442	636019987990464	636011300820087	68	20231101	Norma
2	60	251961560678	20231101173941	20231101173958	636019987990464	636011300920110	17	20231101	Norma
3	60	251963302347	20231101172810	20231101173244	636019987990464	636011300820087	274	20231101	Norma
4	60	251929899393	20231101171919	20231101171922	636019987990464	636011300820087	3	20231101	Norma
5	04	251913834747	20231101185816	20231101190603	636013093848785	636012130342753	466	20231101	Norma
6	04	251968125959	20231101170632	20231101170713	636013093848785	636012130342753	41	20231101	Norma
7	04	251911017431	20231101171819	20231101172121	636013093848785	636012130342753	182	20231101	Norma
8	04	251968125959	20231101185510	20231101185620	636013093848785	636012130342753	70	20231101	Norma

Figure 4. 4 Retrieve Sample 1-hour data from Database.



```
Union all
SELECT [ID]
```

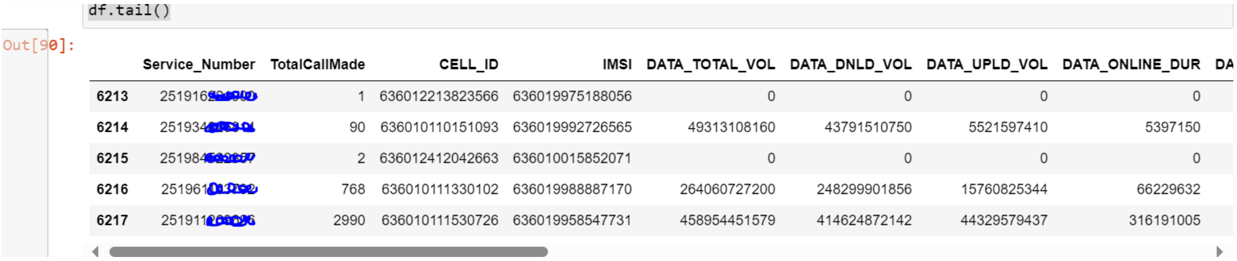
	ɔer	Other_Number	START_TIME	END_TIME	IMSI	CELL_ID	CALL_DUR	Data_day	Class
3	0	25198[redacted]	20231101123131	20231101123918	636019987990464	636011300820087	466	20231101	Normal
4	0	25196[redacted]	20231101172810	20231101173244	636019987990464	636011300820087	274	20231101	Normal
5	8	251975[redacted]	20231102073431	20231102073510	636019934899453	636011700110563	39	20231102	Normal
6	8	25191[redacted]	20231102083741	20231102083835	636019934899453	636011700110573	54	20231102	Normal
7	8	25198[redacted]	20231102094930	20231102094951	636019934899453	636011700110563	21	20231102	Normal
8	8	25197[redacted]	20231102094119	20231102094145	636019934899453	636011700110563	26	20231102	Normal
9	8	2519[redacted]	20231103090227	20231103090323	636019934899453	636011700110575	56	20231103	Normal
10	8	2519[redacted]	20231103205455	20231103205647	636019934899453	636011700110563	112	20231103	Normal
11	8	2519[redacted]	20231103195621	20231103200130	636019934899453	636011700110563	308	20231103	Normal
12	8	2519[redacted]	20231103190407	20231103190717	636019934899453	636011700110563	190	20231103	Normal
13	4	2519[redacted]	20231107124313	20231107124420	636019991314776	636010111532434	67	20231107	Normal

Figure 4. 5 Retrieve Sample 7-days Data from Database.

4.2.2 Random Forest (RF) Model Experiment

Random Forest is a powerful machine learning algorithm that can be employed for SIM box fraud detection. Therefore, in this thesis work, we will feed three different datasets of 1_hour, 1_day and 7_days to the random forest (RF) algorithm. For the experimentation, we used Jupiter notebook IDE with Python engine, below are all the executed options and related results.

We started the experiment by loading the analyzed 1-hour dataset using pandas' library. Pandas is a powerful and widely used Python library for data manipulation and analysis. It provides data structures like Series and Data Frame, which are efficient for handling and analyzing structured data.



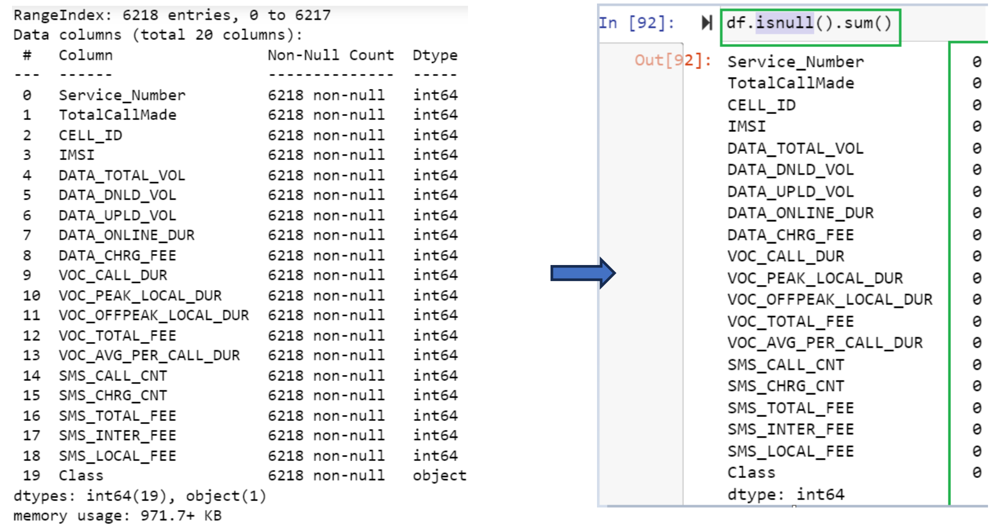
```
df.tail()
```

Out[90]:

	Service_Number	TotalCallMade	CELL_ID	IMSI	DATA_TOTAL_VOL	DATA_DNLD_VOL	DATA_UPLD_VOL	DATA_ONLINE_DUR	DA
6213	251916[redacted]	1	636012213823566	636019975188056	0	0	0	0	
6214	251934[redacted]	90	636010110151093	636019992726565	49313108160	43791510750	5521597410	5397150	
6215	251984[redacted]	2	636012412042663	636010015852071	0	0	0	0	
6216	251961[redacted]	768	636010111330102	636019988887170	264060727200	248299901856	15760825344	66229632	
6217	251911[redacted]	2990	636010111530726	636019958547731	458954451579	414624872142	44329579437	316191005	

Figure 4. 6 Loading data set using pandas.

To deal with null values, the data set has been processed and analyzed well in SQL Server relational database management system, that helps to improve the performance and accuracy of the model critically.

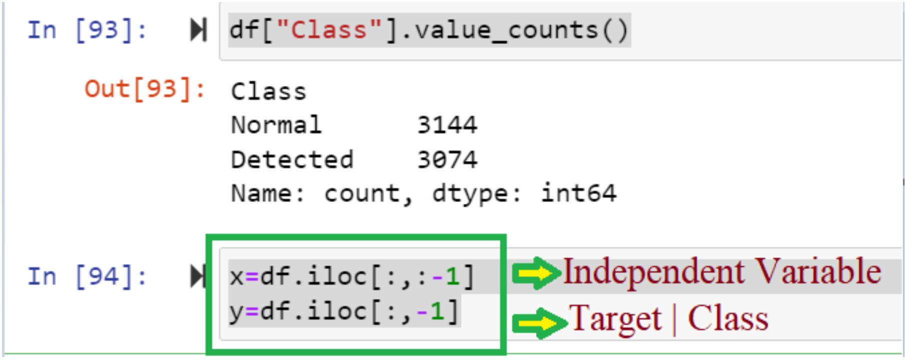


```
RangeIndex: 6218 entries, 0 to 6217
Data columns (total 20 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Service_Number        6218 non-null   int64
 1   TotalCallMade         6218 non-null   int64
 2   CELL_ID               6218 non-null   int64
 3   IMSI                  6218 non-null   int64
 4   DATA_TOTAL_VOL        6218 non-null   int64
 5   DATA_DNLD_VOL         6218 non-null   int64
 6   DATA_UPLD_VOL         6218 non-null   int64
 7   DATA_ONLINE_DUR       6218 non-null   int64
 8   DATA_CHRG_FEE         6218 non-null   int64
 9   VOC_CALL_DUR          6218 non-null   int64
 10  VOC_PEAK_LOCAL_DUR    6218 non-null   int64
 11  VOC_OFFPEAK_LOCAL_DUR 6218 non-null   int64
 12  VOC_TOTAL_FEE         6218 non-null   int64
 13  VOC_AVG_PER_CALL_DUR  6218 non-null   int64
 14  SMS_CALL_CNT          6218 non-null   int64
 15  SMS_CHRG_CNT          6218 non-null   int64
 16  SMS_TOTAL_FEE         6218 non-null   int64
 17  SMS_INTER_FEE         6218 non-null   int64
 18  SMS_LOCAL_FEE         6218 non-null   int64
 19  Class                 6218 non-null   object
dtypes: int64(19), object(1)
memory usage: 971.7+ KB
```

```
In [92]:  df.isnull().sum()

Out[92]:  Service_Number           0
          TotalCallMade            0
          CELL_ID                  0
          IMSI                     0
          DATA_TOTAL_VOL           0
          DATA_DNLD_VOL            0
          DATA_UPLD_VOL            0
          DATA_ONLINE_DUR          0
          DATA_CHRG_FEE            0
          VOC_CALL_DUR             0
          VOC_PEAK_LOCAL_DUR       0
          VOC_OFFPEAK_LOCAL_DUR    0
          VOC_TOTAL_FEE            0
          VOC_AVG_PER_CALL_DUR     0
          SMS_CALL_CNT             0
          SMS_CHRG_CNT             0
          SMS_TOTAL_FEE            0
          SMS_INTER_FEE            0
          SMS_LOCAL_FEE            0
          Class                    0
          dtype: int64
```

Figure 4. 7 Managing Missing Values

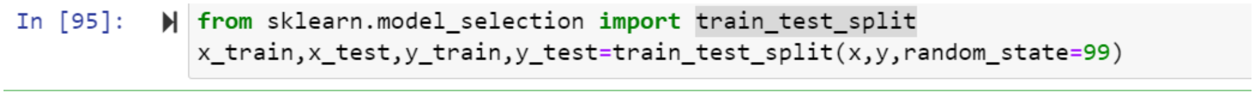
For manipulating the data, we have used pandas iloc function, a popular data manipulation library in Python. It stands for "integer location" and is used for integer-location based indexing for selection by position.



```
In [93]:  df["Class"].value_counts()

Out[93]:  Class
          Normal      3144
          Detected    3074
          Name: count, dtype: int64

In [94]:  x=df.iloc[:,:-1]
          y=df.iloc[:,-1]
```

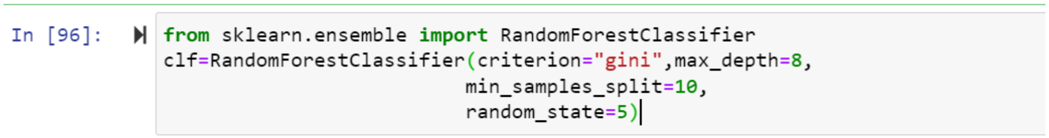
Figure 4. 8 Positioning Variables

The next operation is splitting the dataset into two parts of training and testing data, using a function called ***train_test_split.*** It is a function commonly used in machine learning to split a dataset into two or more parts for training and testing purposes. This function is typically employed during the model development process to assess how well a trained model generalizes to new, unseen data. In Python, the train_test_split function is often part of the scikit-learn library, which is a popular machine learning library. The basic usage of train_test_split involves providing the input features and corresponding labels, and the function randomly splits the data into training and testing sets.

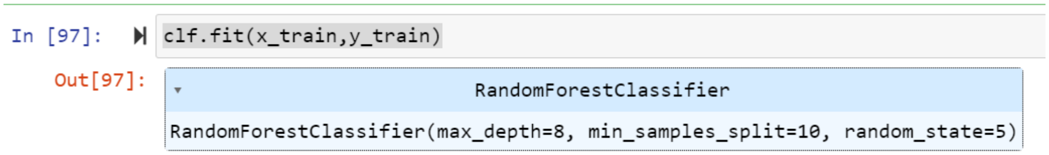


```
In [95]:  from sklearn.model_selection import train_test_split
          x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=99)
```

Then after variable and class positioning to x, y and splitting of the dataset to training and testing, the next task is calling the ***RandomForestClassifier*** python class to consume the dataset and compute predictions using ***sklearn.ensemble*** module. The ensemble module focuses on joint learning methods. Ensemble methods combine multiple individual models to create a stronger and more robust model. The idea is that by combining different models, you can often achieve better predictive performance than with any individual model.

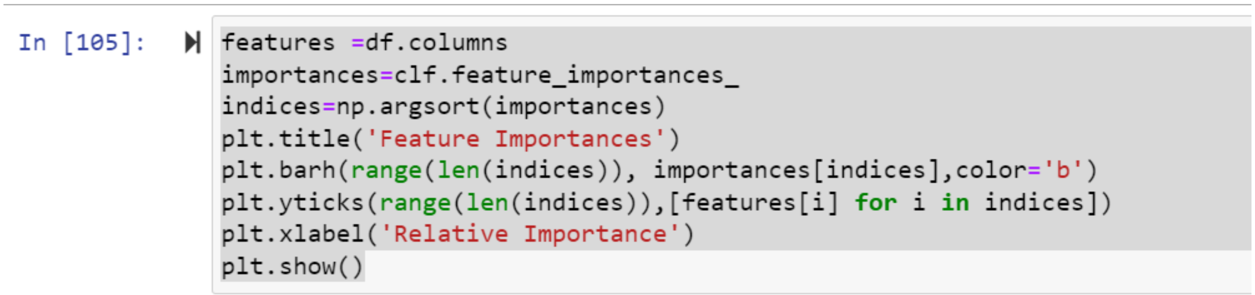


```
In [96]:  from sklearn.ensemble import RandomForestClassifier
          clf=RandomForestClassifier(criterion="gini",max_depth=8,
                                     min_samples_split=10,
                                     random_state=5)
```



```
In [97]:  clf.fit(x_train,y_train)

Out[97]:                      RandomForestClassifier
          RandomForestClassifier(max_depth=8, min_samples_split=10, random_state=5)
```

To identify the importance of all available features, we have used ***feature_importances_,*** is a property of a ***RandomForestClassifier*** object that returns a ***numpy*** array of feature importance's. Each value in the array represents the importance of the corresponding feature in the model.



```
In [105]:  features =df.columns
           importances=clf.feature_importances_
           indices=np.argsort(importances)
           plt.title('Feature Importances')
           plt.barh(range(len(indices)), importances[indices],color='b')
           plt.yticks(range(len(indices)),[features[i] for i in indices])
           plt.xlabel('Relative Importance')
           plt.show()
```

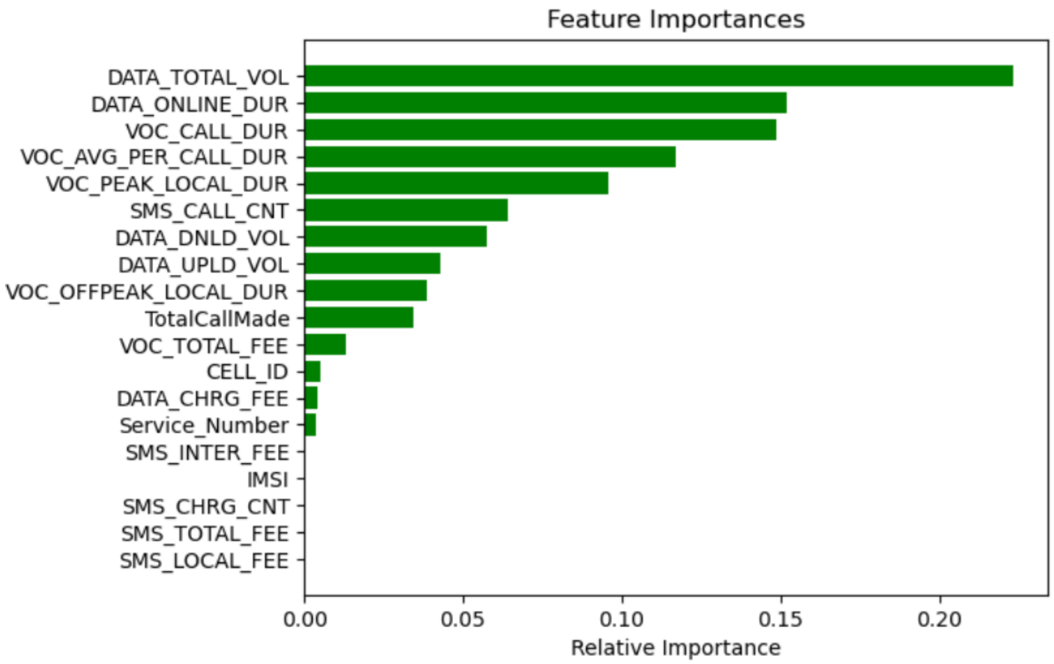
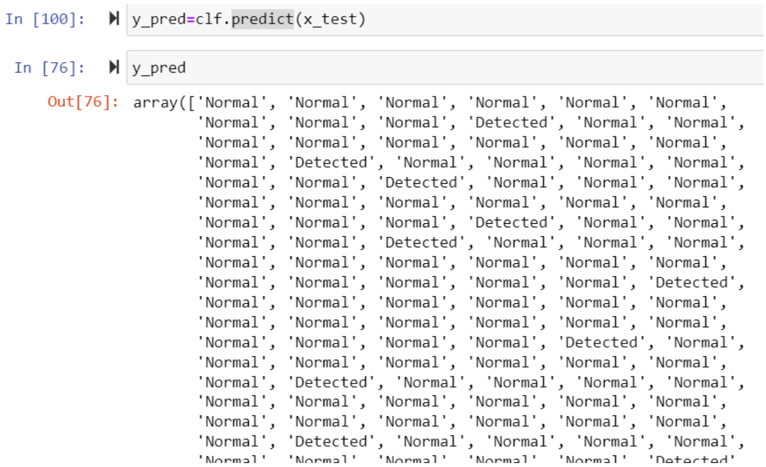


Figure 4. 9 Feature Importance using Random Forest Classifier

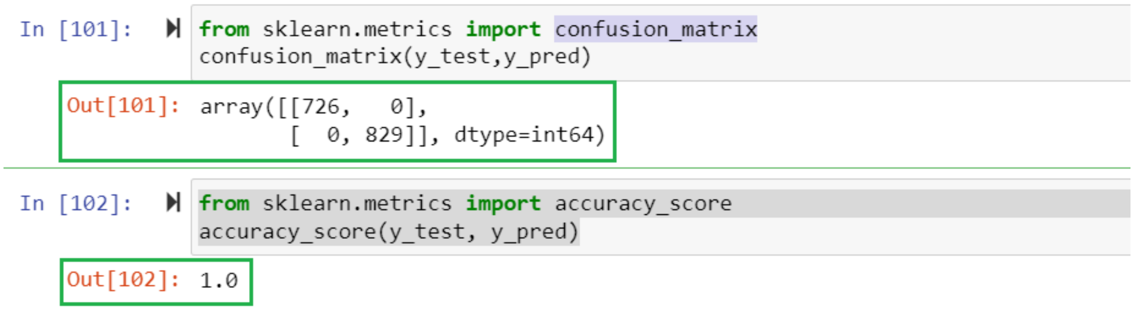
Then, we run prediction with split test data using a function called predict(). The ***predict*** function is used to make predictions on new, unseen data. The predict method takes an input dataset and returns the predicted outcomes based on the trained Random Forest model. In a Random Forest algorithm, the prediction is made through an ensemble of decision trees. Each tree independently makes a prediction, and the final prediction is determined by a majority vote (for classification) or an average (for regression) of the individual tree predictions.



```
In [100]:  y_pred=clf.predict(x_test)

 In [76]:  y_pred

Out[76]: array(['Normal', 'Normal', 'Normal', 'Normal', 'Normal', 'Normal',
                'Normal', 'Normal', 'Normal', 'Detected', 'Normal', 'Normal',
                'Normal', 'Normal', 'Normal', 'Normal', 'Normal', 'Normal',
                'Normal', 'Detected', 'Normal', 'Normal', 'Normal', 'Normal',
                'Normal', 'Normal', 'Detected', 'Normal', 'Normal', 'Normal',
                'Normal', 'Normal', 'Normal', 'Normal', 'Normal', 'Normal',
                'Normal', 'Normal', 'Normal', 'Detected', 'Normal', 'Normal',
                'Normal', 'Normal', 'Detected', 'Normal', 'Normal', 'Normal',
                'Normal', 'Normal', 'Normal', 'Normal', 'Normal', 'Normal',
                'Normal', 'Normal', 'Normal', 'Normal', 'Normal', 'Detected',
                'Normal', 'Normal', 'Normal', 'Normal', 'Normal', 'Normal',
                'Normal', 'Normal', 'Normal', 'Normal', 'Normal', 'Normal',
                'Normal', 'Normal', 'Normal', 'Normal', 'Detected', 'Normal',
                'Normal', 'Normal', 'Normal', 'Normal', 'Normal', 'Normal',
                'Normal', 'Detected', 'Normal', 'Normal', 'Normal', 'Normal',
                'Normal', 'Normal', 'Normal', 'Normal', 'Normal', 'Normal',
                'Normal', 'Normal', 'Normal', 'Normal', 'Normal', 'Normal',
                'Normal', 'Detected', 'Normal', 'Normal', 'Normal', 'Normal',
                'Normal'  'Normal'  'Normal'  'Normal'  'Normal'  'Detected'
```

Figure 4. 10 Random Forest Prediction with test data

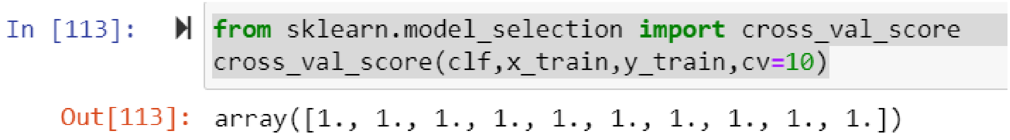
Then, we used confusion matrix to evaluate the performance of a predictive model. It summarizes the predictions of a model on a set of data for comparison with the actual outcomes. The matrix has four entries of true positive (TP), true negative (TN), false positive (FP), and false negative (FN).



```
In [101]:   from sklearn.metrics import confusion_matrix
            confusion_matrix(y_test,y_pred)

Out[101]: array([[726,   0],
                 [  0, 829]], dtype=int64)

In [102]:   from sklearn.metrics import accuracy_score
            accuracy_score(y_test, y_pred)

Out[102]: 1.0
```

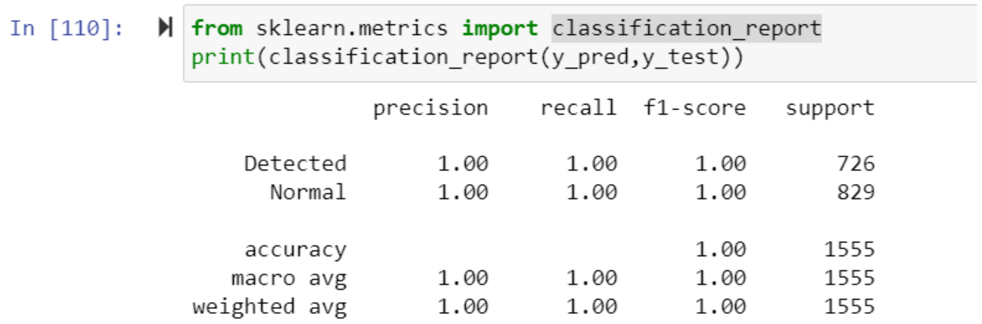
Figure 4. 11 Random Forest Confusion Matrix

To validate the performance of the model, we have used cross validation technique using ***cross_val_score,*** python function that simplifies the process of cross-validation by processing of the data splitting and model evaluation. Cross-validation is a technique used in machine learning to assess the performance of a model by splitting the dataset into multiple subsets (folds) and using each fold as a testing set while the remaining folds are used for training.



```
In [113]:   from sklearn.model_selection import cross_val_score
            cross_val_score(clf,x_train,y_train,cv=10)

Out[113]: array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

Finally, we have generated performance report of the model using ***classification_report***, a tool in sklearn that used for evaluating the performance of a classification model and provides a inclusive summary of various metrics of precision, recall, F1-score, and support for each class in a classification problem.



```
In [110]:  from sklearn.metrics import classification_report
            print(classification_report(y_pred,y_test))
```

```
              precision    recall  f1-score   support

    Detected       1.00      1.00      1.00       726
      Normal       1.00      1.00      1.00       829

    accuracy                           1.00      1555
   macro avg       1.00      1.00      1.00      1555
weighted avg       1.00      1.00      1.00      1555
```

Figure 4. 12 Random Forest Classification Report

Figure 5.12 shows that 100% accuracy has been reported with the random forest algorithm and we just used 1-day and 7-days dataset, and the result keep the same for the three dataset options.

Table 4. 1 Random Forest Classifier Summary

	precision	recall	f1-score	accuracy
1_Hour	1.00	1.00	1.00	1.00
1_Day	1.00	1.00	1.00	1.00
7_Day	1.00	1.00	1.00	1.00

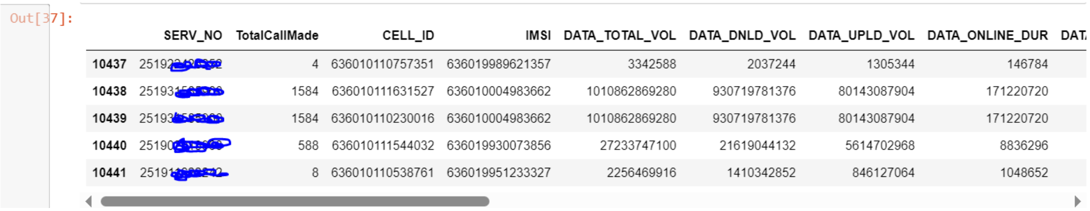
4.2.2 Support Vector Machine(SVM) Model Experiment

Support Vector Machine also one of the most used algorithms to solve different classification problems. In this thesis, we have experimented with 1_hour, 1_day and 7_days dataset and the result and procedures described as follow. To train the model properly, we have followed the step to choose an appropriate kernel function that fit to the problem characteristics, train the model using labeled data of ***normal*** and ***detected*** subscribers and we try to optimize the hyperparameters using cross validation to generate the maximum optimum result.

The kernel function is a crucial component of SVMs as it allows them to operate in a higher-dimensional space without explicitly computing the coordinates of the data in that space. The kernel function is a mathematical function that computes the dot product of the data points in a transformed feature space. The choice of the kernel function significantly influences the performance of the SVM.

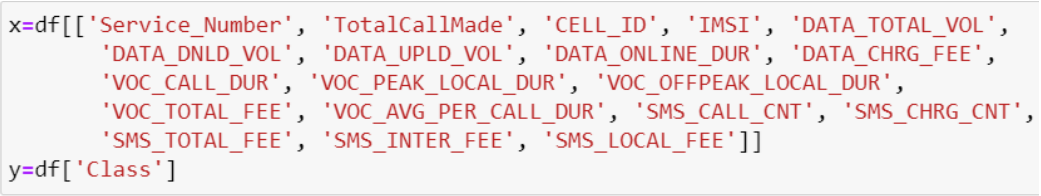
In this thesis work, we have used linier and polynomial kernel with various hyperparameters to check the performance due to the mechanism of data point distribution in the space might have possibilities to generate different accuracy performance. In addition, we have experimented different hyperparameter tuning settings to check for optimal performance, due to hyperparameters also control the behavior of the algorithm and can significantly impact accuracy, generalization, and even training time.

Therefore, as we had in the previous experimentation, we have loaded the different datasets to the python data structure using panda's module for further processing. then, independent and class features are represented in x, y variable.



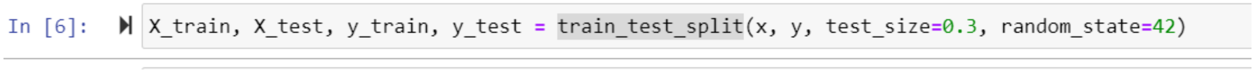
Out[37]:

	SERV_NO	TotalCallMade	CELL_ID	IMSI	DATA_TOTAL_VOL	DATA_DNLD_VOL	DATA_UPLD_VOL	DATA_ONLINE_DUR	DAT
10437	25192[redacted]	4	636010110757351	636019989621357	3342588	2037244	1305344	146784	
10438	25193[redacted]	1584	636010111631527	636010004983662	1010862869280	930719781376	80143087904	171220720	
10439	25193[redacted]	1584	636010110230016	636010004983662	1010862869280	930719781376	80143087904	171220720	
10440	25190[redacted]	588	636010111544032	636019930073856	27233747100	21619044132	5614702968	8836296	
10441	25191[redacted]	8	636010110538761	636019951233327	2256469916	1410342852	846127064	1048652	



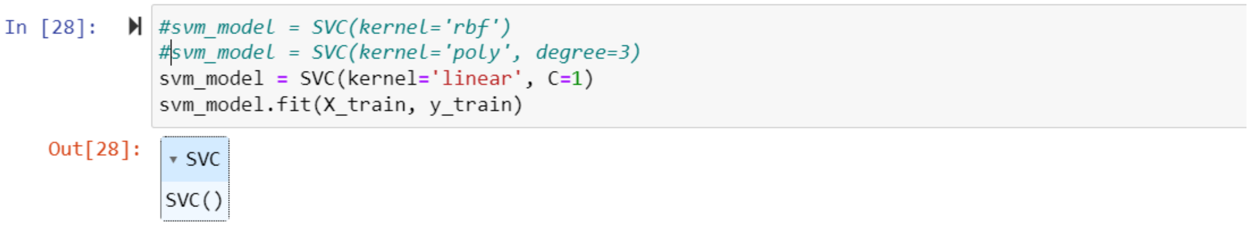
```
x=df[['Service_Number', 'TotalCallMade', 'CELL_ID', 'IMSI', 'DATA_TOTAL_VOL',
       'DATA_DNLD_VOL', 'DATA_UPLD_VOL', 'DATA_ONLINE_DUR', 'DATA_CHRG_FEE',
       'VOC_CALL_DUR', 'VOC_PEAK_LOCAL_DUR', 'VOC_OFFPEAK_LOCAL_DUR',
       'VOC_TOTAL_FEE', 'VOC_AVG_PER_CALL_DUR', 'SMS_CALL_CNT', 'SMS_CHRG_CNT',
       'SMS_TOTAL_FEE', 'SMS_INTER_FEE', 'SMS_LOCAL_FEE']]
y=df['Class']
```

Next, the data has been split in to training and test set using sklearn python library called ***train_test_split*** and here we applied percent split method to consume 30% of the data to test the model and 70% of the data for training the model.



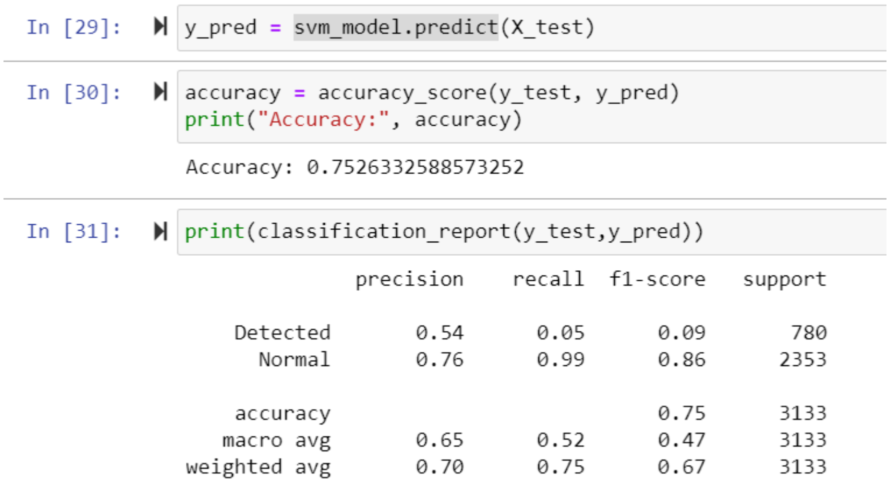
```
In [6]:  X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=42)
```

Then, the SVM algorithm is applied to the 1_hour, 1_day and 7_days dataset with ***linear***, ***poly*** and ***rbf*** kernel option and hyperparameter tunings (Regularization, degree, gamma).



```
In [28]:  #svm_model = SVC(kernel='rbf')
          #svm_model = SVC(kernel='poly', degree=3)
          svm_model = SVC(kernel='linear', C=1)
          svm_model.fit(X_train, y_train)

Out[28]:  ▾ SVC
          SVC()
```

After the model creation, we have utilized the ***svm_model.predict()*** method to perform prediction with the help of consumed training data.



```
In [29]:  y_pred = svm_model.predict(X_test)

In [30]:  accuracy = accuracy_score(y_test, y_pred)
          print("Accuracy:", accuracy)

          Accuracy: 0.7526332588573252

In [31]:  print(classification_report(y_test,y_pred))

                        precision    recall  f1-score   support

              Detected       0.54      0.05      0.09       780
                Normal       0.76      0.99      0.86      2353

              accuracy                           0.75      3133
             macro avg       0.65      0.52      0.47      3133
          weighted avg       0.70      0.75      0.67      3133
```

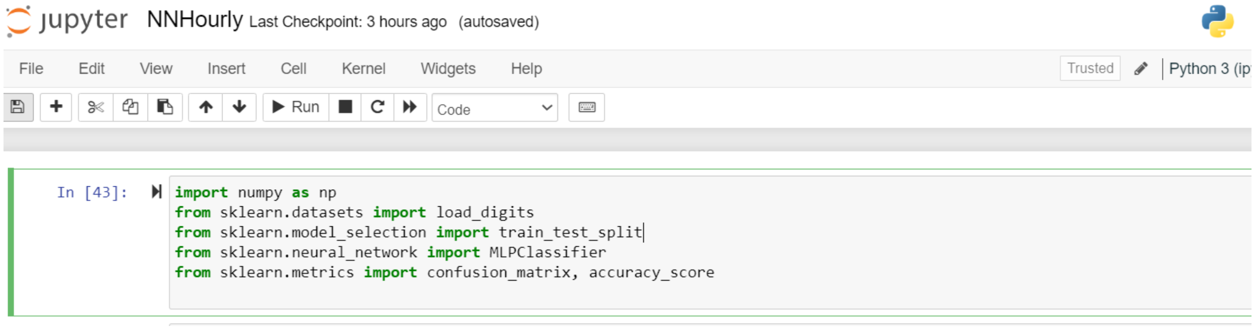
Table 4. 2 SVM Summary

	precision	recall	f1-score	accuracy
1_Hour	0.73	0.54	0.42	54%
1_Day	0.65	0.52	0.47	75%
7_Day	0.67	0.52	0.47	75%

4.2.3 Neural Network (NN) Model Experiment

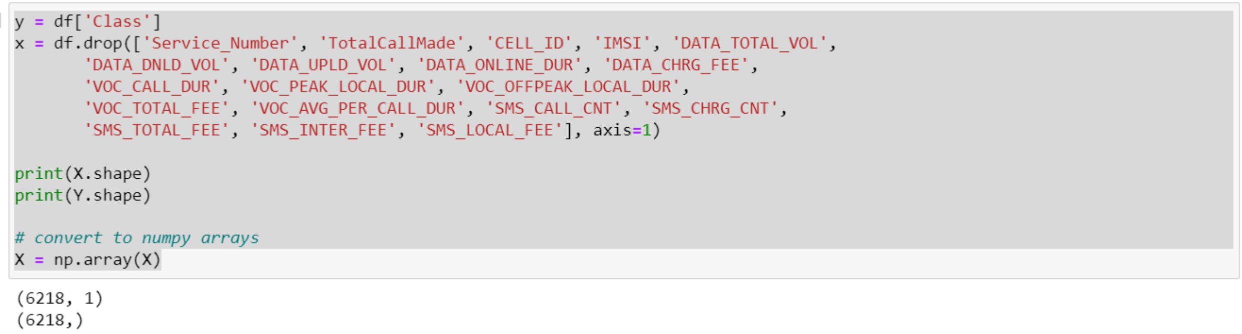
Neural Networks is one of the supervised machine learning algorithms, that require labeled data to train and used for both classification and regression problems. In this thesis work, we have implemented Neural Network for classification, using the scikit-learn toolkit. Initially, the dataset is prepared, preprocessed, integrated, and analyzed using SQL server DBMS and we split it into test and train datasets. The model will be trained on the training data, and we will use the test data to evaluate the model.

We have used the ***train_test_split*** function and the accuracy and confusion matrix metrics from the sklearn library to split the data into train and test samples and to evaluate the results, respectively. Then, the sklearn library built in model for Neural Network has utilized accordingly.


jupyter NNHourly Last Checkpoint: 3 hours ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ip
Run
Code

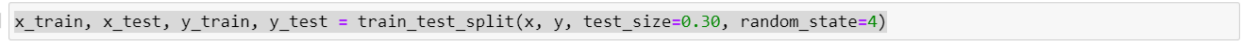
```
In [43]:  import numpy as np
          from sklearn.datasets import load_digits
          from sklearn.model_selection import train_test_split
          from sklearn.neural_network import MLPClassifier
          from sklearn.metrics import confusion_matrix, accuracy_score
```

Then after loading of the dataset into data frame using sklearn toolkit pandas module, the dataset has split into train and test using the sklearn, ***train_test_split*** library. We have applied percent split approach using 70/30 split, in which 70 percent data will be train data and 30 percent of the data shall be test data.



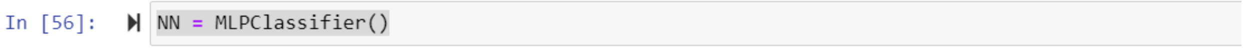
```
y = df['Class']
x = df.drop(['Service_Number', 'TotalCallMade', 'CELL_ID', 'IMSI', 'DATA_TOTAL_VOL',
        'DATA_DNLD_VOL', 'DATA_UPLD_VOL', 'DATA_ONLINE_DUR', 'DATA_CHRG_FEE',
        'VOC_CALL_DUR', 'VOC_PEAK_LOCAL_DUR', 'VOC_OFFPEAK_LOCAL_DUR',
        'VOC_TOTAL_FEE', 'VOC_AVG_PER_CALL_DUR', 'SMS_CALL_CNT', 'SMS_CHRG_CNT',
        'SMS_TOTAL_FEE', 'SMS_INTER_FEE', 'SMS_LOCAL_FEE'], axis=1)

print(X.shape)
print(Y.shape)

# convert to numpy arrays
X = np.array(X)
```

```
(6218, 1)
(6218,)
```



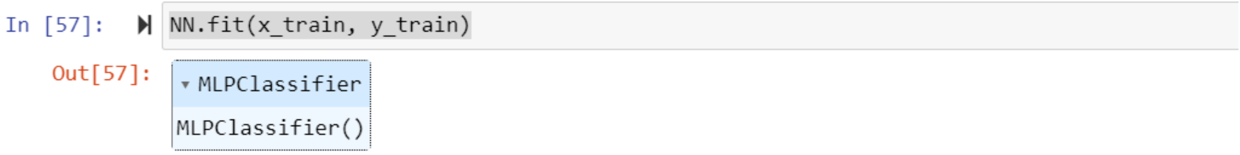
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.30, random_state=4)
```

Then, we have applied ***MLPClassifier*** class for creating a neural network classifier. The MLPClassifier class in scikit-learn serves as a versatile tool for constructing neural network-based classifiers. Short for Multi-Layer Perceptron, this class provides a flexible implementation of artificial neural networks, allowing users to configure various parameters to tailor the model to specific tasks. One of the key parameters is hidden_layer_sizes, enabling the specification of the number of neurons in each hidden layer, thereby influencing the model's capacity to capture complex patterns in the data. With the ability to set parameters such as activation functions, optimization algorithms, and regularization techniques, the MLPClassifier accommodates diverse machine learning scenarios.

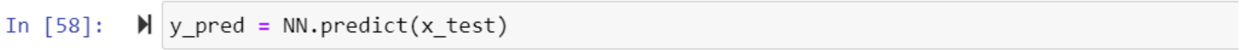


```
In [56]:  NN = MLPClassifier()
```

To test the model, we will use the testing data and testing labels using ***predict()*** method of ***MLPClassifier*** class.

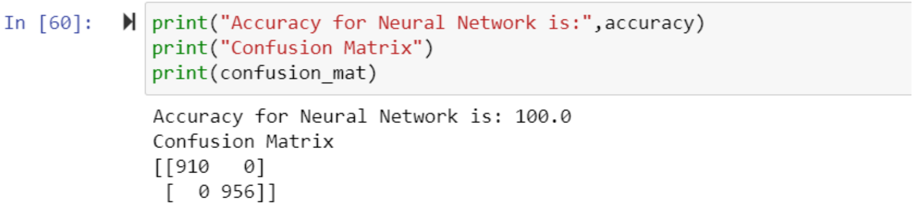


```
In [57]:  NN.fit(x_train, y_train)

Out[57]:  ▾ MLPClassifier
          MLPClassifier()
```

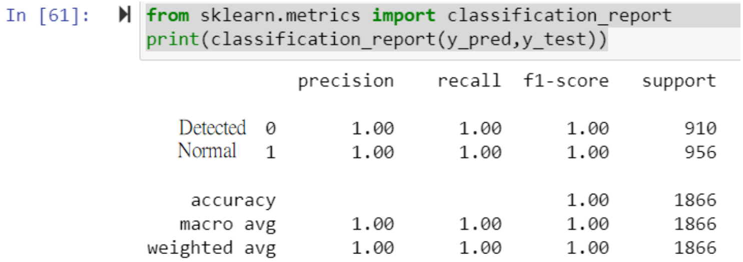


```
In [58]:  y_pred = NN.predict(x_test)
```

Finally, we applied ***accuracy_score*** function with prediction function as a parameter and 100% accuracy score has recorded with the cautiously prepared three dataset.



```
In [60]:  print("Accuracy for Neural Network is:",accuracy)
          print("Confusion Matrix")
          print(confusion_mat)

          Accuracy for Neural Network is: 100.0
          Confusion Matrix
          [[910   0]
           [  0 956]]
```



```
In [61]:  from sklearn.metrics import classification_report
          print(classification_report(y_pred,y_test))

                        precision    recall  f1-score   support

           Detected  0       1.00      1.00      1.00       910
           Normal    1       1.00      1.00      1.00       956

              accuracy                           1.00      1866
             macro avg       1.00      1.00      1.00      1866
          weighted avg       1.00      1.00      1.00      1866
```

Hence, in this chapter we have experimented the collected, preprocessed, aggregated, and analyzed 1_hour, 1_day, 7_days datasets with three machine learning algorithms of Random Forest (RF), Support Vector Machine (SVM) and Neural network (NN) using scikit-learn python library. In the final classification report, there is no result difference between the provided dataset categories and 100% accuracy has recorded by RF and NN algorithms. In contrast, the SVM algorithm returned 54% accuracy for 1_hour dataset and

75% accuracy has recorded for 1_day and 7_days datasets for different kernel and hyperparameter options.

In the provided dataset, the fraudulent numbers SMS, Voice and Data utilization is extremely low and that might create class imbalances at some level and prevent the SVM to draw clear decision boundary and score less accuracy comparing with RF and NN.

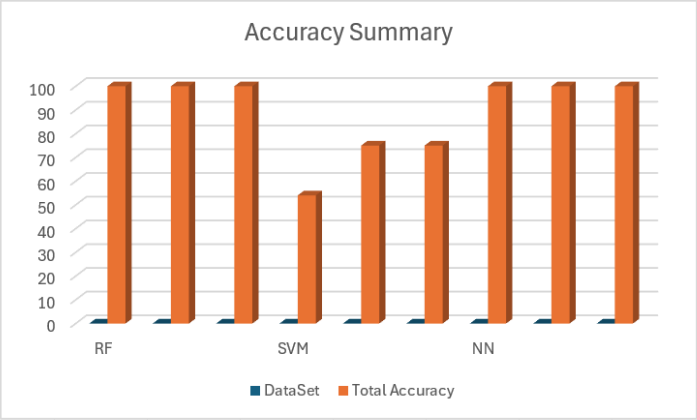


Figure 4. 13 Accuracy Summary

At the initial stage of the thesis, we have defined the below research questions and now it addressed well with the support of experimentation.

1. What are the key features that can effectively distinguish between legitimate and fraudulent SIM-box traffic?

    - In the experiment, useful features for SIM-Box fraud detection were identified, derived, and used.

2. What machine learning method can effectively use provided datasets to predict the behavioral patterns of SIM-Box fraud ?

    - Based on the executed experimentation RF and NN is best fit to classify fraudulent and legit subscribers.

Chapter Five

Conclusion and Recommendations

The primary objective of this thesis was to create a model that would use machine learning algorithms to identify and forecast SIM-Box fraud. In order to accomplish this goal, CDR data was gathered, preprocessed, aggregated and analyzed. Better performing models were suggested after they had been assessed and experimented. The research's results and conclusions are covered in this chapter. A conclusion and recommendations were drawn from these data.

5.1 Conclusion

Telecom operators play a crucial role in providing communication services, enabling connectivity worldwide. However, they often face challenges, one of which is the persistent threat of SIM box fraud. SIM box fraud occurs when fraudulent individuals use devices known as SIM boxes to manipulate telecommunications networks. These fraudulent activities involve routing international calls through these devices to simulate local calls, thus avoiding international tariffs. This deceptive practice not only results in substantial financial losses for telecom operators but also undermines the integrity of their networks [24]. To combat SIM box fraud, operators employ advanced monitoring systems and collaborate with regulatory authorities to detect and block those involved subscribers in these illegal activities.

This thesis is also one of the contributions to finding a solution for the spotted problem that SIM box fraud causes to the telecom business. Hence, we have collected CDR data from the telecom systems of CRM and CBS, then we have run preprocessing, aggregate data of different sources, derived new attributes from the existing attributes, identify important features to the target classification and prepared the final dataset as 1_hour, 1_day and 7_day category. Then we have feed those datasets to RF, SVM and NN supervised machine learning classification algorithms.

For the experimentation use have used phyton sklearn, a powerful machine learning library in Python, that provides efficient tools for data analysis and modeling, including various algorithms of classification. Hence, based on the experimentation result RF and NN algorithms generated 100% accuracy for the three different datasets and SVM generated 54% accuracy for the 1_hour dataset and 75% accuracy for the 1_day and 7_day datasets.

5.2 Recommendation

As extending work of this thesis, we recommend that future research in SIM box fraud detection explore the synergistic integration of machine learning models with expert-based custom solutions. By combining the analytical power of machine learning algorithms with the defined insights of expert-designed solutions, a more robust and adaptive solution can be developed to combat the evolving landscape of SIM box fraud. Machine learning models can effectively analyze large datasets, identifying patterns and anomalies that may be indicative of fraudulent activities. Meanwhile, expert-based custom solutions can provide domain-specific knowledge, incorporating industry expertise and real-world insights that machine learning models may lack. This collaborative approach has the potential to enhance the accuracy, efficiency, and adaptability of SIM box fraud detection systems to various telecom operators.

Reference

1 Roger L. Freeman, "Fundamentals of Telecommunications", 1999, John Wiley & Sons, Inc

2 ADAM ANDERSON, ERIK WYNTER & TAYLER WOODALL, "Cyber Crime and Cyber Insurance", 2019

3 Dean Armstrong, Thomas Steward, Shyam Thakerar, "Cyber Risks and insurance: The legal principles" 2020

4 Chubb Insurance Australia Limited, "Cyber Enterprise Risk Management", 2021

5 Ethio Telecom, https://www.ethiotelecom.et/ethio-telecom-2022-23-annual-business-performance/, accessed:12/2/2023

6 R. Abhari, D. Goldof, M. Lanzerotti, T. Samad, "TELECOMMUNICATIONS SYSTEM RELIABILITY ENGINEERING, THEORY, AND PRACTICE" IEEE Press, 2012

7 Gordon L.Stuber, "Principles of Mobile Communication" 2016

8 P.GNANASIVAM, "TELECOMMUNICATION SWITCHING AND NETWORKS", NEW AGE INTERNATIONAL LIMITED PUBLISHIERS

9 European Network Integration for Railways group, FFFS for Voice and Data Services
Interconnection & Roaming between GSM-R networks, 2016

10 Tutorial Point, https://www.tutorialspoint.com/telecom-billing/system-architecture.htm, accessed 12/2/2023

11 Andriy Burkov, "The Hundred Page Machine Learning" 2019

12 Aurélien Géron, "Hands - On Machine Learning with Scikit - Learn and TensorFlow
Concepts, Tools, and Techniques to Build Intelligent Systems" O'Reilly Media, 2017

13 Anshul Saini," Decision Tree Algorithm – A Complete Guide"
https://www.analyticsvidhya.com/blog/2021/08/decision-tree-algorithm/
accessed:10/10/2023

14 Sruthi E R , “Understand Random Forest Algorithms With Examples”, https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/; last- accessed:10/01/2023

15 Anshul Saini, Guide on Support Vector Machine (SVM) Algorithm, https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/ last- accessed:10/05/2023

16 Frehiwot Mola, “Analysis and Detection Mechanisms of SIM Box Fraud in The Case of Ethio Telecom”, Addis Ababa University, 2017

17 Kahsu Hagos, “SIM-Box Fraud Detection Using Data Mining Techniques: The Case of ethio telecom”, Addis Ababa University, 2018

18 Fitsum Tesfaye, “Near-Real Time SIM-box Fraud Detection Using Machine Learning in the case of ethio telecom”, Addis Ababa University, 2020

19 Roselina Sallehuddin*, Subariah Ibrahim, Azlan Mohd Zain, Abdikarim Hussein Elmi, “Detecting SIM Box Fraud by Using Support Vector Machine and Artificial Neural Network”, Jurnal Teknologi, 2015

20 Pragati Baheti, “A Simple Guide to Data Preprocessing in Machine Learning”, https://www.v7labs.com/blog/data-preprocessing-guide, last-accessed:12/26/2023

21 Suhang Wang , Jiliang Tang, “Feature Selection”, Springer, January 2016

22 GeeksforGeeks, https://www.geeksforgeeks.org/ml-extra-tree-classifier-for-feature-selection/ last-accessed:01/10/2024

23 ScikitLearn, https://scikit-learn.org/stable/modules/cross_validation.html, last-accessed:01/15/2024

24 Dmitry Sumin, “Enterprise Telecom Fraud in Focus: Challenges and Solutions”, June 27 2023

Appendices

```
=============================Source Code for Get  CDR Data=================
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="getCDRData.aspx.cs"
Inherits="SIMBOXDetection.getCDRData" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>CDR API</title>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-
awesome.min.css">
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
<link rel="stylesheet" href="https://cdn.datatables.net/1.10.16/css/dataTables.bootstrap4.min.css" />
<script src="https://cdn.datatables.net/1.10.16/js/jquery.dataTables.min.js"
type="text/javascript"></script>
<script src="https://cdn.datatables.net/1.10.16/js/dataTables.bootstrap4.min.js"
type="text/javascript"></script>

<script type="text/javascript">
        $(document).ready(function () {
            $("#GridView1").prepend($("<thead></thead>").append($(this).find("tr:first"))).dataTable();
        });
</script>
</head>
<body>
    <form id="form1" runat="server">
        <div class="container py-3">
            <h2 class="text-center text-uppercase">API to Get CDR Data for SIM-BOX Fraud
Detection</h2>
            <div class="card">
                <div class="card-header bg-primary text-uppercase text-white">
                    <h5>Telecom CDR Data</h5>
                </div>
                <div class="card-body">
                    <button style="margin-bottom:10px;" type="button" class="btn btn-primary" data-
toggle="modal" data-target="#myModal">
                        <i class="fa fa-plus-circle"></i> Connect to CDR DB</button>
                     <asp:Button ID="btnSave" runat="server" CssClass="btn btn-outline-primary" Text="Save
to SIM-BOX SOlUTION" OnClick="btnSave_Click" />
                    <div class="modal fade" id="myModal">
                        <div class="modal-dialog">
                            <div class="modal-content">
                                <div class="modal-header">
```

```
                                    <h4 class="modal-title">Connect to Telecom CDR DB</h4>
                                    <button type="button" class="close" data-dismiss="modal">×</button>
                                </div>
                                <div class="modal-body">
                                    <div class="row">
                                        <div class="col-md-12">
                                            <div class="form-group">
                                                <label>Provide Database Path</label>
                                                <div class="input-group">
                                                    <div class="custom-file">
                                                        <asp:FileUpload ID="FileUpload1" CssClass="custom-file-input" runat="server" />

                                                        <label class="custom-file-label"></label>
                                                    </div>
                                                    <label id="filename"></label>
                                                    <div class="input-group-append">
                                                        <asp:Button ID="btnUpload" runat="server" CssClass="btn btn-outline-primary" Text="Connect" OnClick="btnUpload_Click" />
                                                    </div>
                                                </div>
                                                <asp:Label ID="lblMessage" runat="server"></asp:Label>
                                            </div>
                                        </div>
                                    </div>
                                </div>
                                <div class="modal-footer">
                                    <button type="button" class="btn btn-danger" data-dismiss="modal">Close</button>
                                </div>
                            </div>
                        </div>
                    </div>

                    <asp:GridView ID="GridView1" HeaderStyle-CssClass="bg-primary text-white" ShowHeaderWhenEmpty="true" runat="server" AutoGenerateColumns="false" CssClass="table table-bordered``">
                        <EmptyDataTemplate>
                            <div class="text-center">No record found</div>
                        </EmptyDataTemplate>
                        <Columns>
                           <%-- <asp:BoundField HeaderText="ID" DataField="ID" /> --%>
                            <asp:BoundField HeaderText="CBS_SUBS_ID" DataField="CBS_SUBS_ID" ItemStyle-Width="40%"/>
                            <asp:BoundField HeaderText="ACCT_ID" DataField="ACCT_ID" />
                            <asp:BoundField HeaderText="SERV_NO" DataField="SERV_NO" />
                            <asp:BoundField HeaderText="OTHER_NO" DataField="OTHER_NO" />
                            <asp:BoundField HeaderText="CUST_ID" DataField="CUST_ID" />
                           <asp:BoundField HeaderText="CALL_TOLL_TYPE" DataField="CALL_TOLL_TYPE" />
                           <asp:BoundField HeaderText="START_DATE" DataField="START_DATE" />
                           <asp:BoundField HeaderText="START_TIME" DataField="START_TIME" />
                           <asp:BoundField HeaderText="END_TIME" DataField="END_TIME" />
```

```
                              <asp:BoundField HeaderText="SPECL_NO" DataField="SPECL_NO" />
                             <asp:BoundField HeaderText="NET_TECH_TYPE" DataField="NET_TECH_TYPE" />
                             <asp:BoundField HeaderText="IMSI" DataField="IMSI" />
                             <asp:BoundField HeaderText="CELL_ID" DataField="CELL_ID" />
                             <asp:BoundField HeaderText="SGMT_TYPE" DataField="SGMT_TYPE" />
                             <asp:BoundField HeaderText="PROFILE_ID" DataField="PROFILE_ID" />
                            <asp:BoundField HeaderText="SERV_TYPE" DataField="SERV_TYPE" />
                            <asp:BoundField HeaderText="CALL_DUR" DataField="CALL_DUR" />
                            <asp:BoundField HeaderText="CHRG_DUR" DataField="CHRG_DUR" />
                            <asp:BoundField HeaderText="FREE_DUR" DataField="FREE_DUR" />
                            <asp:BoundField HeaderText="ACTUAL_USG_DUR" DataField="ACTUAL_USG_DUR" />
                            <asp:BoundField HeaderText="Data_day" DataField="Data_day" />
                        </Columns>
                    </asp:GridView>
                </div>
            </div>
        </div>
    </form>
</body>
</html>
```

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Data.OleDb;
using System.Data.Common;
using System.Web.UI.WebControls;
using static System.Collections.Specialized.BitVector32;
using System.Reflection;

namespace SIMBOXDetection
{
    public partial class getCDRData : System.Web.UI.Page
    {
        SqlConnection con;
        SqlCommand cmd;
        DBConnection DBcon = new DBConnection();
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                BindGridview();
            }
        }

        private void BindGridview()
        {
            string CS = ConfigurationManager.ConnectionStrings["SIMBOXDetection"].ConnectionString;
            using (SqlConnection con = new SqlConnection(CS))
            {
                SqlCommand cmd = new SqlCommand("GetAllCDR", con);
                cmd.CommandType = CommandType.StoredProcedure;
                con.Open();
                GridView1.DataSource = cmd.ExecuteReader();
                GridView1.DataBind();
            }
        }

        protected void btnUpload_Click(object sender, EventArgs e)
        {
            DBcon.dataBaseConnection();
            String filename;
            if (FileUpload1.PostedFile != null)
            {
                //try
                //{
                DBcon.dataBaseConnection();

                filename = FileUpload1.PostedFile.FileName;
                filename = filename.Substring(filename.LastIndexOf("\\") + 1);
                FileUpload1.PostedFile.SaveAs(Server.MapPath(".") + "/UploadFile/" + filename);
                DataSet dsRecords = new DataSet();
                string strConn = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source= " +
Server.MapPath(".") + "/UploadFile/" + filename + ";Extended Properties=Excel 8.0;";
                OleDbDataAdapter daGetExcel = new OleDbDataAdapter("SELECT * FROM [Sheet1$]",
strConn);
                daGetExcel.Fill(dsRecords, "Sheet1");
                System.Threading.Thread.Sleep(20000);
                int numofRecords = dsRecords.Tables.Count;
                if (numofRecords > 0)
                {
                    GridView1.DataSource = dsRecords;
                    GridView1.DataBind();
                    GridView1.Visible = true;

                }
                /*
                 *
                    string path = string.Concat(Server.MapPath("~/UploadFile/" + FileUpload1.FileName));
                    FileUpload1.SaveAs(path);
                    // Connection String to Excel Workbook
                    string excelCS = string.Format("Provider=Microsoft.ACE.OLEDB.12.0;Data
Source={0};Extended Properties=Excel 8.0", path);
                    using (OleDbConnection con = new OleDbConnection(excelCS))
                    {
                        OleDbCommand cmd = new OleDbCommand("select * from [Sheet1$]", con);
                        con.Open();
                        // Create DbDataReader to Data Worksheet
                        DbDataReader dr = cmd.ExecuteReader();
                        // SQL Server Connection String
                        string CS =
ConfigurationManager.ConnectionStrings["SIMBOXDetection"].ConnectionString;
                    // Bulk Copy to SQL Server
                        SqlBulkCopy bulkInsert = new SqlBulkCopy(CS);
                        bulkInsert.DestinationTableName = "tbl_CDRNOV1";
                        bulkInsert.WriteToServer(dr);
                        BindGridview();
                        lblMessage.Text = "Your file uploaded successfully";
                        lblMessage.ForeColor = System.Drawing.Color.Green;
                    }*/
                //}
                //catch (Exception)
                //{
                //    lblMessage.Text = "Your file not uploaded";
                //    lblMessage.ForeColor = System.Drawing.Color.Red;
                //}
            }
        }

        protected void btnSave_Click(object sender, EventArgs e)
        {
            DBcon.dataBaseConnection();
            foreach (GridViewRow row in GridView1.Rows)
            {

                String CBS_SUBS_ID = Convert.ToString(row.Cells[0].Text.ToString());
                String ACCT_ID = Convert.ToString(row.Cells[1].Text.ToString());
                Int64 SERV_NO = Convert.ToInt64(row.Cells[2].Text.ToString());
                Int64 OTHER_NO = Convert.ToInt64(row.Cells[3].Text.ToString());
                String CUST_ID = Convert.ToString(row.Cells[4].Text.ToString());

                Int64 CALL_TOLL_TYPE = onvert.ToInt64(row.Cells[5].Text.ToString());
                Int64 START_DATE = Convert.ToInt64(row.Cells[6].Text.ToString());
                Int64 START_TIME = Convert.ToInt64(row.Cells[7].Text.ToString());
                Int64 END_TIME = Convert.ToInt64(row.Cells[8].Text.ToString());
```

```
      String SPECL_NO = Convert.ToString(row.Cells[9].Text.ToString());

      Int64 NET_TECH_TYPE =Convert.ToInt64(row.Cells[10].Text.ToString());
      String IMSI = Convert.ToString(row.Cells[11].Text.ToString());
      String CELL_ID = Convert.ToString(row.Cells[12].Text.ToString());
      Int64 SGMT_TYPE = Convert.ToInt64(row.Cells[13].Text.ToString());
      Int64 PROFILE_ID = Convert.ToInt64(row.Cells[14].Text.ToString());

      Int64 SERV_TYPE = Convert.ToInt64(row.Cells[15].Text.ToString());
      Int64 CALL_DUR = Convert.ToInt64(row.Cells[16].Text.ToString());
      Int64 CHRG_DUR = Convert.ToInt64(row.Cells[17].Text.ToString());
      Int64 FREE_DUR = Convert.ToInt64(row.Cells[18].Text.ToString());
      Int64 ACTUAL_USG_DUR = connvert.ToInt64(row.Cells[19].Text.ToString());

      Int64 Data_day = Convert.ToInt64(row.Cells[20].Text.ToString());


      String InsertCDR = "insert into tbl_CDRNOV1
(CBS_SUBS_ID,ACCT_ID,SERV_NO,OTHER_NO,CUST_ID,CALL_TOLL_TYPE,START_DATE,START_TIME,END_
TIME,SPECL_NO,NET_TECH_TYPE,IMSI,CELL_ID,SGMT_TYPE,PROFILE_ID,SERV_TYPE,CALL_DUR,CHRG_DU
R,FREE_DUR,ACTUAL_USG_DUR,Data_day)
values(@CBS_SUBS_ID,@ACCT_ID,@SERV_NO,@OTHER_NO,@CUST_ID,@CALL_TOLL_TYPE,@START_D
ATE,@START_TIME,@END_TIME,@SPECL_NO,@NET_TECH_TYPE,@IMSI,@CELL_ID,@SGMT_TYPE,@PRO
FILE_ID,@SERV_TYPE,@CALL_DUR,@CHRG_DUR,@FREE_DUR,@ACTUAL_USG_DUR,@Data_day)";
      SqlCommand cmd = new SqlCommand(InsertCDR, DBcon.con);
      cmd.Parameters.AddWithValue(@"CBS_SUBS_ID ", CBS_SUBS_ID);
      cmd.Parameters.AddWithValue(@"ACCT_ID", ACCT_ID);
      cmd.Parameters.AddWithValue(@"SERV_NO", SERV_NO);
      cmd.Parameters.AddWithValue(@"OTHER_NO", OTHER_NO);
      cmd.Parameters.AddWithValue(@"CUST_ID", CUST_ID);
      cmd.Parameters.AddWithValue(@"CALL_TOLL_TYPE", CALL_TOLL_TYPE);
      cmd.Parameters.AddWithValue(@"START_DATE", START_DATE);
      cmd.Parameters.AddWithValue(@"START_TIME ", START_TIME);
      cmd.Parameters.AddWithValue(@"END_TIME", END_TIME);
      cmd.Parameters.AddWithValue(@"SPECL_NO", SPECL_NO);
      cmd.Parameters.AddWithValue(@"NET_TECH_TYPE", NET_TECH_TYPE);
      cmd.Parameters.AddWithValue(@"IMSI", IMSI);

      cmd.Parameters.AddWithValue(@"CELL_ID", CELL_ID);
      cmd.Parameters.AddWithValue(@"SGMT_TYPE", SGMT_TYPE);
      cmd.Parameters.AddWithValue(@"PROFILE_ID", PROFILE_ID);
      cmd.Parameters.AddWithValue(@"SERV_TYPE", SERV_TYPE);

      cmd.Parameters.AddWithValue(@"CALL_DUR", CALL_DUR);
      cmd.Parameters.AddWithValue(@"CHRG_DUR", CHRG_DUR);
      cmd.Parameters.AddWithValue(@"FREE_DUR", FREE_DUR);
      cmd.Parameters.AddWithValue(@"ACTUAL_USG_DUR", ACTUAL_USG_DUR);
      cmd.Parameters.AddWithValue(@"Data_day", Data_day);
      cmd.ExecuteNonQuery();

    }
```

```
        }
    }
}

==============================Source Code for Data Aggregation=================
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="DataAggregation.aspx.cs"
Inherits="SIMBOXDetection.DataAggregation" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
        </div>
    </form>
</body>
</html>
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Data.OleDb;
using System.Data.Common;
using System.Web.UI.WebControls;
using static System.Collections.Specialized.BitVector32;
using System.Reflection;
using System.Drawing;

namespace SIMBOXDetection
{
    public partial class DataAggregation : System.Web.UI.Page
    {
        DBConnection DBcon = new DBConnection();
        protected void Page_Load(object sender, EventArgs e)
        {
            DBcon.dataBaseConnection();
            String GetSIMDetection = "Select distinct SERV_NO FROM
[SIMBOXFRAUDDETECTION].[dbo].[ViewLastOneHour]";
            SqlDataAdapter adaptGetSIMDetection= new SqlDataAdapter(GetSIMDetection,DBcon.con);
            DataTable dtGetSIMDetection = new DataTable();
            adaptGetSIMDetection.Fill(dtGetSIMDetection);
            Response.Write(dtGetSIMDetection.Rows.Count);

            for (int i= 0; i < dtGetSIMDetection.Rows.Count;i++)
            {
                // Response.Write(dtGetSIMDetection.Rows[i]["SERV_NO"].ToString());
                String ServiceNumber = dtGetSIMDetection.Rows[i]["SERV_NO"].ToString();
            String CountIncomingCall = "Select Count ([OTHER_NO]) as IncomingCalls FROM
[SIMBOXFRAUDDETECTION].[dbo].[ViewLastOneHour] where [OTHER_NO]='"+ ServiceNumber + "'";
            SqlCommand cmdGetSIMDetection = new SqlCommand(CountIncomingCall, DBcon.con);
            String GetIncomingCount = Convert.ToString(cmdGetSIMDetection.ExecuteScalar());

            String SUMIncomingCallDuration = "Select SUM ([CALL_DUR_Sec]) as IncomingDuration
FROM [SIMBOXFRAUDDETECTION].[dbo].[ViewLastOneHour] where [OTHER_NO]='" + ServiceNumber
+ "'";
            SqlCommand cmdSUMIncomingCallDuration = new
SqlCommand(SUMIncomingCallDuration, DBcon.con);
            String GetIncomingCallDuSum =
Convert.ToString(cmdSUMIncomingCallDuration.ExecuteScalar());

            String SUMOutgoingCallDuration = "  Select SUM ([CALL_DUR_Sec]) as OutgoingDuration
FROM [SIMBOXFRAUDDETECTION].[dbo].[ViewLastOneHour] where SERV_NO='" + ServiceNumber +
"'";
            SqlCommand cmdSUMOutgoingCallDuration = new
SqlCommand(SUMOutgoingCallDuration, DBcon.con);
            String GetOutgoingCallSum =
Convert.ToString(cmdSUMOutgoingCallDuration.ExecuteScalar());

            String InsertCDR = "insert into tbl_Aggregation
(SERV_NO,Total_Incoming_Calls,Total_Incoming_Duration,Total_Outgoing_Duration)
values(@ServiceNumber,@GetIncomingCount,@GetIncomingCallDuSum,@GetOutgoingCallSum)";
            SqlCommand cmd = new SqlCommand(InsertCDR, DBcon.con);
            cmd.Parameters.AddWithValue(@"ServiceNumber", ServiceNumber);
            cmd.Parameters.AddWithValue(@"GetIncomingCount", GetIncomingCount);
            cmd.Parameters.AddWithValue(@"GetIncomingCallDuSum", GetIncomingCallDuSum);
            cmd.Parameters.AddWithValue(@"GetOutgoingCallSum", GetOutgoingCallSum);
            cmd.ExecuteNonQuery();

        }


    }
  }
}
```

============================Database Preprocessing and Analysis================

```
USE [SIMBOXFRAUDDETECTION]
GO

/****** Object:  Table [dbo].[tblCallRecord]    Script Date: 1/21/2024 12:59:22 PM
******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TABLE [dbo].[tblCallRecord](
       [ID] [int] IDENTITY(1,1) NOT NULL,
       [Service_Number] [nvarchar](50) NULL,
       [Other_Number] [nvarchar](50) NULL,
       [START_TIME] [nvarchar](50) NULL,
       [END_TIME] [nvarchar](50) NULL,
       [IMSI] [nvarchar](50) NULL,
       [CELL_ID] [nvarchar](50) NULL,
       [CALL_DUR] [nvarchar](50) NULL,
       [Data_day] [nvarchar](50) NULL,
       [Class] [nvarchar](50) NULL,
 CONSTRAINT [PK_tblCallRecord] PRIMARY KEY CLUSTERED
(
       [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO


USE [SIMBOXFRAUDDETECTION]
GO

/****** Object:  Table [dbo].[tblVoice]    Script Date: 1/21/2024 12:59:43 PM
******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[tblVoice](
       [ID] [int] IDENTITY(1,1) NOT NULL,
       [SERV_NO] [bigint] NULL,
       [SERV_TYPE] [int] NULL,
       [VOC_CALL_DUR] [bigint] NULL,
       [VOC_PEAK_LOCAL_DUR] [bigint] NULL,
       [VOC_OFFPEAK_LOCAL_DUR] [bigint] NULL,
       [VOC_TOTAL_FEE] [bigint] NULL,
       [VOC_AVG_PER_CALL_DUR] [bigint] NULL,
       [Data_day] [bigint] NULL,
       [Class] [nvarchar](50) NULL,
 CONSTRAINT [PK_tblVoice] PRIMARY KEY CLUSTERED
(
       [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO


USE [SIMBOXFRAUDDETECTION]
GO

/****** Object:  Table [dbo].[tblData]    Script Date: 1/21/2024 12:59:59 PM
******/
```

```
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[tblData](
        [ID] [int] IDENTITY(1,1) NOT NULL,
        [SERV_NO] [nvarchar](50) NULL,
        [SERV_TYPE] [int] NULL,
        [DATA_TOTAL_VOL] [bigint] NULL,
        [DATA_DNLD_VOL] [bigint] NULL,
        [DATA_UPLD_VOL] [bigint] NULL,
        [DATA_ONLINE_DUR] [bigint] NULL,
        [DATA_CHRG_FEE] [bigint] NULL,
        [Data_day] [bigint] NULL,
        [Class] [nvarchar](50) NULL,
 CONSTRAINT [PK_tblSMS] PRIMARY KEY CLUSTERED
(
        [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO


USE [SIMBOXFRAUDDETECTION]
GO

/****** Object:  Table [dbo].[tblSMS]    Script Date: 1/21/2024 1:00:19 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[tblSMS](
        [ID] [int] IDENTITY(1,1) NOT NULL,
        [SERV_NO] [bigint] NULL,
        [SERV_TYPE] [int] NULL,
        [SMS_CALL_CNT] [bigint] NULL,
        [SMS_CHRG_CNT] [bigint] NULL,
        [SMS_TOTAL_FEE] [bigint] NULL,
        [SMS_INTER_FEE] [bigint] NULL,
        [SMS_LOCAL_FEE] [bigint] NULL,
        [Data_day] [bigint] NULL,
        [Class] [nvarchar](50) NULL,
 CONSTRAINT [PK_tblSMS_1] PRIMARY KEY CLUSTERED
(
        [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
```

```sql
USE [SIMBOXFRAUDDETECTION]
GO

/****** Object:  View [dbo].[ViewAggregationFinal]    Script Date: 1/21/2024
1:01:11 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE VIEW [dbo].[ViewAggregationFinal]
AS
SELECT dbo.tblData.SERV_NO, dbo.tblCallRecord.Other_Number,
dbo.tblCallRecord.CALL_DUR, dbo.tblCallRecord.CELL_ID, dbo.tblCallRecord.IMSI,
dbo.tblCallRecord.START_TIME, dbo.tblCallRecord.END_TIME,
dbo.tblData.DATA_TOTAL_VOL, dbo.tblData.DATA_DNLD_VOL,
             dbo.tblData.DATA_UPLD_VOL, dbo.tblData.DATA_ONLINE_DUR,
dbo.tblData.DATA_CHRG_FEE, dbo.tblSMS.SMS_CALL_CNT, dbo.tblSMS.SMS_CHRG_CNT,
dbo.tblSMS.SMS_TOTAL_FEE, dbo.tblSMS.SMS_INTER_FEE, dbo.tblSMS.SMS_LOCAL_FEE,
dbo.tblVoice.VOC_CALL_DUR,
             dbo.tblVoice.VOC_PEAK_LOCAL_DUR, dbo.tblVoice.VOC_OFFPEAK_LOCAL_DUR,
dbo.tblVoice.VOC_TOTAL_FEE, dbo.tblVoice.VOC_AVG_PER_CALL_DUR,
dbo.tblCallRecord.Data_day, dbo.tblCallRecord.Class
FROM   dbo.tblCallRecord INNER JOIN
             dbo.tblData ON dbo.tblCallRecord.Service_Number = dbo.tblData.SERV_NO
INNER JOIN
             dbo.tblSMS ON dbo.tblCallRecord.Service_Number = dbo.tblSMS.SERV_NO
INNER JOIN
             dbo.tblVoice ON dbo.tblCallRecord.Service_Number =
dbo.tblVoice.SERV_NO
GO



USE [SIMBOXFRAUDDETECTION]
GO

/****** Object:  View [dbo].[ViewAggregationAnalysis1]    Script Date: 1/21/2024
1:01:52 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE VIEW [dbo].[ViewAggregationAnalysis1]
AS
SELECT DISTINCT
             SERV_NO, COUNT(Other_Number) AS TotalCallMade, CELL_ID, IMSI,
DATA_TOTAL_VOL, DATA_DNLD_VOL, DATA_UPLD_VOL, DATA_ONLINE_DUR, DATA_CHRG_FEE,
SMS_CALL_CNT, SMS_CHRG_CNT, SMS_TOTAL_FEE, SMS_INTER_FEE, SMS_LOCAL_FEE,
             SUM(VOC_CALL_DUR) AS VOC_CALL_DUR, SUM(VOC_PEAK_LOCAL_DUR) AS
VOC_PEAK_LOCAL_DUR, SUM(VOC_OFFPEAK_LOCAL_DUR) AS VOC_OFFPEAK_LOCAL_DUR,
SUM(VOC_TOTAL_FEE) AS VOC_TOTAL_FEE, SUM(VOC_AVG_PER_CALL_DUR)
             AS VOC_AVG_PER_CALL_DUR, Class
FROM   dbo.ViewAggregationFinal
```

```
GROUP BY SERV_NO, CELL_ID, IMSI, DATA_TOTAL_VOL, DATA_DNLD_VOL, DATA_UPLD_VOL,
DATA_ONLINE_DUR, DATA_CHRG_FEE, SMS_CALL_CNT, SMS_CHRG_CNT, SMS_TOTAL_FEE,
SMS_INTER_FEE, SMS_LOCAL_FEE, Class
GO

USE [SIMBOXFRAUDDETECTION]
GO

/****** Object:  View [dbo].[View1_hourDataset]    Script Date: 1/21/2024 1:02:24
PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

create view [dbo].[View1_hourDataset]  as
SELECT TOP (55000000) [ID]
      ,[Service_Number]
      ,[Other_Number]
      ,[START_TIME]
      ,[END_TIME]
      ,[IMSI]
      ,[CELL_ID]
      ,[CALL_DUR]
      ,[Data_day]
      ,[Class]
  FROM [SIMBOXFRAUDDETECTION].[dbo].[tblCallRecord] where Class='Detected' and
Data_day='20231101' or START_TIME >= 20231101080606 and START_TIME <=
20231101095906

  union all
  SELECT TOP (55000000) [ID]
      ,[Service_Number]
      ,[Other_Number]
      ,[START_TIME]
      ,[END_TIME]
      ,[IMSI]
      ,[CELL_ID]
      ,[CALL_DUR]
      ,[Data_day]
      ,[Class]
  FROM [SIMBOXFRAUDDETECTION].[dbo].[tblCallRecord] where Class='Normal' and
Data_day='20231101' or START_TIME >= 20231101080606 and START_TIME <=
20231101095906


GO

USE [SIMBOXFRAUDDETECTION]
GO

/****** Object:  View [dbo].[view1_hourDatasetAll]    Script Date: 1/21/2024
1:02:53 PM ******/
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO

CREATE VIEW [dbo].[view1_hourDatasetAll]
AS
SELECT dbo.View1_hourDataset.Service_Number, dbo.View1_hourDataset.Other_Number,
dbo.View1_hourDataset.START_TIME, dbo.View1_hourDataset.END_TIME,
dbo.View1_hourDataset.IMSI, dbo.View1_hourDataset.CELL_ID,
dbo.View1_hourDataset.CALL_DUR,
             dbo.View1_hourDataset.Data_day, dbo.View1_hourDataset.Class,
dbo.tblData.DATA_TOTAL_VOL, dbo.tblData.DATA_DNLD_VOL, dbo.tblData.DATA_UPLD_VOL,
dbo.tblData.DATA_ONLINE_DUR, dbo.tblData.DATA_CHRG_FEE, dbo.tblVoice.VOC_CALL_DUR,
             dbo.tblVoice.VOC_PEAK_LOCAL_DUR, dbo.tblVoice.VOC_OFFPEAK_LOCAL_DUR,
dbo.tblVoice.VOC_TOTAL_FEE, dbo.tblVoice.VOC_AVG_PER_CALL_DUR,
dbo.tblSMS.SMS_CALL_CNT, dbo.tblSMS.SMS_CHRG_CNT, dbo.tblSMS.SMS_TOTAL_FEE,
dbo.tblSMS.SMS_INTER_FEE,
             dbo.tblSMS.SMS_LOCAL_FEE
FROM   dbo.View1_hourDataset INNER JOIN
             dbo.tblData ON dbo.View1_hourDataset.Service_Number =
dbo.tblData.SERV_NO INNER JOIN
             dbo.tblVoice ON dbo.View1_hourDataset.Service_Number =
dbo.tblVoice.SERV_NO INNER JOIN
             dbo.tblSMS ON dbo.View1_hourDataset.Service_Number =
dbo.tblSMS.SERV_NO
GO

USE [SIMBOXFRAUDDETECTION]
GO

/****** Object:  View [dbo].[View1_HourAnalysisFinal]    Script Date: 1/21/2024
1:03:13 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE VIEW [dbo].[View1_HourAnalysisFinal]
AS
SELECT DISTINCT
             Service_Number, COUNT(Other_Number) AS TotalCallMade, CELL_ID, IMSI,
SUM(DATA_TOTAL_VOL) AS DATA_TOTAL_VOL, SUM(DATA_DNLD_VOL) AS DATA_DNLD_VOL,
SUM(DATA_UPLD_VOL) AS DATA_UPLD_VOL, SUM(DATA_ONLINE_DUR) AS DATA_ONLINE_DUR,
             SUM(DATA_CHRG_FEE) AS DATA_CHRG_FEE, SUM(VOC_CALL_DUR) AS
VOC_CALL_DUR, SUM(VOC_PEAK_LOCAL_DUR) AS VOC_PEAK_LOCAL_DUR,
SUM(VOC_OFFPEAK_LOCAL_DUR) AS VOC_OFFPEAK_LOCAL_DUR, SUM(VOC_TOTAL_FEE) AS
VOC_TOTAL_FEE,
             SUM(VOC_AVG_PER_CALL_DUR) AS VOC_AVG_PER_CALL_DUR, SUM(SMS_CALL_CNT)
AS SMS_CALL_CNT, SUM(SMS_CHRG_CNT) AS SMS_CHRG_CNT, SUM(SMS_TOTAL_FEE) AS
SMS_TOTAL_FEE, SUM(SMS_INTER_FEE) AS SMS_INTER_FEE, SUM(SMS_LOCAL_FEE)
             AS SMS_LOCAL_FEE, Class
FROM   dbo.view1_hourDatasetAll
GROUP BY Service_Number, CELL_ID, IMSI, Class
GO
```