Enhancing Over-the-Top (OTT) Bypass Fraud Detection and Classification in Telecom Networks Using Machine Learning Algorithms

Ramadan Abera Haso

A Thesis Submitted to

The Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfilment of the Requirement for the Degree of Master's in

Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

September, 2024

Adama Ethiopia

Enhancing Over-the-Top (OTT) Bypass Fraud Detection and Classification in Telecom Networks Using Machine Learning Algorithms

By Ramadan Abera Haso

Advisor: Dr.-Ing Frezewd Lemma

A Thesis Submitted to the Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfilment of the Requirement for the Degree of Master's in

Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

September, 2024

Adama, Ethiopia

Declaration

I declare that this thesis is entitled **"Enhancing Over-the-Top (OTT) Bypass Fraud Detection and Classification in Telecom Networks Using Machine Learning Algorithms"** is my work and has not been submitted to any university for a similar purpose. The references used in this proposal are duly recognized by proper citations.

Ramadan Abera Haso _____ _____

Name of student Signature Date

Advisor Recommendation

I, the advisor of this thesis, hereby certify that I have read the revised version of the thesis entitled **"Enhancing Over-the-Top (OTT) Bypass Fraud Detection and Classification in Telecom Networks Using Machine Learning Algorithms"** prepared under my guidance by **Ramadan Abera Haso** and submitted in partial fulfillment of the requirements for the degree of Masters of Science in Computer Science and Engineering (CSE). Therefore, I recommend the submission of the revised version of the thesis to the department following the applicable procedures.

Dr.-Ing Frezewd Lemma _____ _____

Major Advisor Signature Date

Approval Page for Advisor

I/we hereby certify that the recommendation and suggestion given by the proposal review committee are appropriately incorporated into the final thesis proposal entitled **"Enhancing Over-the-Top (OTT) Bypass Fraud Detection and Classification in Telecom Networks Using Machine Learning Algorithms"** by **Ramadan Abera Haso**

Dr.-Ing Frezewd Lemma _____ _____

Major Advisor Signature Date

Approval page of M.Sc. Thesis

I, the advisor of the thesis entitled **"Enhancing Over-the-Top (OTT) Bypass Fraud Detection and Classification in Telecom Networks Using Machine Learning Algorithms"** developed by Ramadan Abera Haso, hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis

Dr.-Ing. Frezewd Lemma _____ _____

Major Advisor Signature Date

We, the undersigned, members of the Board of Examiners of the thesis by Ramadan Abera Haso have read and evaluated the thesis entitled **"Enhancing Over-the-Top (OTT) Bypass Fraud Detection and Classification in Telecom Networks Using Machine Learning Algorithms"** and examined the candidate during the open defense. This is, therefore, to certify that the thesis is accepted for the partial fulfillment of the requirement of the degree of Master of Science in Computer Science and Engineering (CSE).

_____ _____ _____

Chairperson Signature Date

_____ _____ _____

Internal Examiner Signature Date

_____ _____ _____

External Examiner Signature Date

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

_____ _____ _____

Department Head Signature Date

_____ _____ _____

School Dean Signature Date

_____ _____ _____

Office of Postgraduate Studies Dean Signature Date

Approval of Board of Reviewers

We, the undersigned, members of the Board of Reviewers of the proposal open defense by **Ramadan Abera Haso** have read and evaluated the thesis proposal entitled **"Enhancing Over-the-Top (OTT) Bypass Fraud Detection and Classification in Telecom Networks Using Machine Learning Algorithms"** and assessed the understanding of the candidate about the proposed research. This is, therefore, to certify that the thesis proposal is accepted and we recommend the implementation of the proposal.

_____	_____	_____
Chairperson	Signature	Date
_____	_____	_____
Reviewer 1	Signature	Date
_____	_____	_____
Reviewer 2	Signature	Date

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

_____	_____	_____
Department Head	Signature	Date
_____	_____	_____
School Dean	Signature	Date
_____	_____	_____
Office of Postgraduate Studies	Dean Signature	Date

ACKNOWLEDGEMENTS

First and foremost, I wish to express my deepest gratitude to Almighty Allah for granting me the strength and opportunity to begin and complete this journey. His guidance and support have been my source of perseverance throughout this endeavor.

I would also like to extend my sincere thanks to my advisor, Dr. Frezewd Lemma, for his invaluable guidance and support from the very start to the conclusion of this work. His insightful ideas and constructive feedback have been a constant source of encouragement, and I will always remember the support he extended to all of us.

Finally, I am deeply grateful to my family and friends for their unwavering support and encouragement during the highs and lows of my time at the university. I also want to express my appreciation to all my classmates in the Computer Science and Engineering program at ASTU for their advice, motivation, and guidance as I navigated the academic research world.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

LIST OF ACRONYMS

AI	Artificial Intelligence
ANN	Artificial Neural Network
CDR	Call Detail Record
CFCA	China Financial Certification Authority
DRS	Disaster Recovery System
FMS	Fraud Management System
IP	Internet Protocol
IRSF	International Revenue Share Fraud
KNN	k-Nearest Neighbors
LR	Logistic Regression
ML	Machine Learning
MSC	Mobile Switching Centre
MSISDN	Mobile Station International Subscriber Directory Number
MT	Mobile Terminator
OTT	Over the Top
QoS	Quality of Service
RAM	Read Access Memory
RF	Random Forest
RIPPER	Repeated Incremental Pruning to Produce Error Reduction
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic Curve
RQ	Research Question
SIM	Subscriber Identity Module
SVM	Support Vector Machine
TB	Tera Byte
TCG	Test Call Generators
TN	True Negative
TP	True Positive
TPR	True Positive Rate
TV	Television

VOIP	Voice Over Internet Protocol
WEKA	Waikato Environment for Knowledge Analysis

Abstract

Telecom fraud has been a significant challenge for operators and organizations worldwide, with fraudsters leveraging technological advancements to perpetrate these crimes. Interconnect bypass fraud, particularly concerning over-the-top (OTT) services, has been one of the most prevalent and damaging forms of fraud, enabling fraudsters to avoid access fees and profit from international calls. The dynamic nature of this fraud has allowed it to circumvent traditional methods like Test Call Generators (TCG) and Fraud Management Systems (FMS), necessitating more sophisticated detection approaches. Our research utilized machine learning methods to improve the detection of OTT bypass fraud. We assessed the performance of three models: Random Forest (RF), Support Vector Machine (SVM), and Logistic Regression (LR). The models were tested using two validation approaches: an 80/20 split for training and testing, as well as 10-fold cross-validation. The experiments were conducted with Call Detail Record (CDR) data from Ethio-Telecom, following feature selection and preprocessing steps like data cleaning, integration, and aggregation. The findings showed that the Random Forest model delivered the highest accuracy across both validation methods. Specifically, RF reached 99.98%, and 99.33% *99.79% accuracy with the training and testing approach, outperforming SVM and LR, which attained 99.46% and 97.85% accuracy, respectively. By incorporating new features and leveraging machine learning, our approach significantly improved fraud detection capabilities, addressing the challenges associated with large datasets and the evolving nature of fraud patterns. Detecting and classifying OTT bypass fraud through these methods offers numerous advantages for telecom companies, including mitigating financial risks, enhancing revenue assurance, improving operational efficiency, safeguarding customer satisfaction, protecting brand reputation, ensuring regulatory compliance, and establishing industry leadership.*

Keywords Telecom Fraud**,** Interconnect Bypass Fraud**,** Over-the-Top (OTT) Services, Machine Learning

CHAPTER ONE

1. INTRODUCTION

1.1 Background of the Study

On a global basis, telecommunication businesses have lost billions of dollars of income due to scammers worldwide(Sujata et al., 2015a). With the evolution of network technologies and the widespread adoption of high-speed data plans, call termination methods have grown increasingly intricate, giving rise to bypass scenarios (Kinfmichael, 2021). Viber, a prominent OTT player, has further complicated this landscape by positioning itself as an interconnect partner, offering to terminate calls through its mobile application (Sujata et al., 2015). However, the legality of this interconnect arrangement is questionable, as no regulatory body recognizes OTT service providers as legitimate interconnect service providers. Despite these concerns, Viber offers its app and services at no cost. In the past, Viber IN calls were enabled by default, but now user consent is required during app installation. Once enabled, Viber IN allows Viber to terminate any call on the app, bypassing the conventional voice interconnect channel altogether. This bypass results in substantial revenue losses for telecommunications companies, especially for international calls, with significantly higher termination costs. Additionally, Viber utilizes the telco's data network to facilitate the landing of these calls (*OTT App Use Undermining SMS Revenue - Telecoms.Com*, n.d.).

Telecommunication fraud poses a significant threat to the revenue and quality of service for telecom operators. Various types of fraud, such as International Revenue Share Fraud (IRSF), Domestic Revenue Share (DRS), and Over-the-top (OTT) bypass fraud, impact the industry. OTT bypass fraud specifically involves using internet-based communication services like viper, WhatsApp, and Skype to evade call termination fees, especially in international calls routed through low-fee countries. Fraud is a growing challenge for communications service providers (CSPs) around the world. While CSPs have dealt with many of the more traditional fraud cases there are growing concerns around the use of voice over IP (VoIP) as a means to defraud CSP interconnect agreements(Sandvine, 2016)(Zhou & Lin, 2018). According to a recent survey by the Communications Fraud Control Association, interconnect bypass fraud is one of the largest causes of lost revenue, costing communications service providers across the globe an estimated $6 billion.

(United States Dollars) annually. Despite efforts to combat fraud, it remains a persistent challenge, with fraudsters, often business-minded individuals, seeking to exploit opportunities (*CFCA 2017 Global Fraud Loss Survey – CFCA*, n.d.).

Table 1. 1 Total Losses of the Companies for 2015 (Babaei et al., 2019).

Fraud type & Fraud methods		Fraud loss in Billon's of Dollars		
		Globally	Western Europe	North America
Fraud type	International Revenue Shared Fraud (IRSF)	10.75	2.07	3.21
	Interconnect Bypass Fraud	5.97	1.15	1.78
	Premium Rate Service Fraud	3.74	0.72	1.12
Fraud methods	Subscription Fraud	8.05	2.4	1.55
	PBX Hacking / IP PBX Hacking	7.47	*2.22*	1.44
	Wangiri Fraud	1.77	0.53	0.34
	Phishing	1.57	0.47	0.3
	Abuse of Service Terms and Conditions	1.17	0.53	0.34
	SMS Faking or Spoofing	0.79	0.23	0.15

Machine learning presents a promising solution for detecting OTT bypass fraud, with algorithms trained on datasets to distinguish between legitimate and fraudulent traffic. Studies, like one from Addis Ababa University, demonstrate the effectiveness of machine learning (Kinfmichael, 2021). However, challenges persist, including the diverse nature of OTT traffic, constant evolution of fraud techniques, and the computational expenses associated with implementing machine learning on large networks. Ongoing research aims to address these challenges by developing more robust algorithms capable of efficiently detecting OTT bypass fraud, contributing valuable insights to the field.

1.2 Motivation of the Study

OTT interconnect bypass fraud poses a growing threat to telecom operators worldwide, leading to significant revenue losses and operational inefficiencies. The dynamic and evolving tactics used by fraudsters make it difficult for traditional detection methods to keep pace. This problem needs to be addressed because failure to do so can result in substantial financial damage for telecom companies, degrade service quality, and erode customer trust(Babaei et al., 2019).

A notable gap in existing research is the reliance on synthetic or generated data, which restricts the applicability of fraud detection models to real-world scenarios. These studies often fail to address the complexities of fraud detection in telecom environments, limiting their practical effectiveness.

The motivation behind this study is to enhance both the detection and classification of fraud by utilizing real-world data from Ethio Telecom and incorporating appropriate features crucial for effective analysis. By deploying scalable machine learning models that leverage these relevant features and handle complex and dynamic data, this research aims to improve fraud detection accuracy, reduce financial losses, and enhance service quality. This approach seeks to bridge the gaps identified in previous research and develop robust, adaptable solutions for **OTT bypass** fraud detection and classification, addressing the growing threat and its impact on telecom operators.

1.3 Statement of the problem

Telecom service providers are significantly impacted by various types of telecom fraud, with Over-the-Top (OTT) bypass fraud emerging as a particularly pressing issue. This form of fraud exploits OTT services like Viber and WhatsApp, wherein fraudsters use a user's Mobile Station International Subscriber Directory Number (MSISDN) to route calls through the internet instead of traditional telecom networks(Cortesão et al., 2005). This circumvention of legitimate billing systems leads to substantial revenue losses for telecom operators, including Ethio Telecom, which currently lacks a specific detection and classification module for OTT bypass fraud in its Fraud Management System (FMS). The absence of such a module is critical because OTT bypass fraud results in the loss of international call termination fees—fees that are vital for the company's revenue and typically paid in foreign currency.

Existing studies exploring machine learning techniques for telecom fraud detection, such as Decision Tree (DT) algorithms(Kinfmichael, 2021), semi-supervised approaches, AdaBoost with J48, and Support Vector Machines (SVM), have shown limitations in addressing the dynamic nature of OTT bypass fraud. A major gap in these methods is their reliance on generated data, often using tools like WEKA, which does not accurately represent real-world scenarios. Additionally, these methods are limited by a small number of features, reducing their effectiveness in complex fraud detection environments(Kinfmichael, 2021)(Tewodros, 2018). DT algorithms, while achieving high accuracy (99%), struggle with data variability. Semi-supervised techniques face challenges with limited labeled data and large volumes of unlabeled data. AdaBoost with J48 performs well in many metrics but suffers from slow classification times and potential overfitting. Similarly, SVM shows high accuracy (95.35%) but encounters difficulties with large, high-dimensional datasets, leading to increased computational complexity and slower processing times.

To address these critical gaps, this study focuses on using real-world data from Ethio Telecom and incorporating a broader set of relevant features. By employing advanced machine learning models that leverage this comprehensive dataset, the research aims to improve both detection and classification of OTT bypass fraud. This approach seeks to enhance accuracy, adaptability, and efficiency in fraud detection, directly addressing the limitations of existing methods and providing robust solutions for telecom companies.

1.4 Research Question

RQ1. What are the essential attributes to consider when implementing machine learning algorithms for detecting and classifying OTT bypass fraud?

RQ2. How should the key attributes be processed to improve the performance of machine learning algorithms in identifying OTT bypass fraud?

RQ3. How can we accurately evaluate the performance of the selected machine learning models to ensure their effectiveness?

RQ4. How do RF,SVM and LR machine learning algorithms compare in terms of their effectiveness in detecting and classifying OTT bypass fraud?

1.5 Objectives

1.5.1 General Objectives

The overall goal of the study is to improve over-the-top (OTT) bypass fraud detection and classification in telecom networks using machine learning algorithms.

1.5.2 Specific Objectives

The specific objectives are:

- To identify and preprocess key attributes from real-world datasets, including preparing data through feature extraction and engineering, to enhance OTT bypass fraud detection and improve model accuracy.
- To develop and implement a robust machine learning framework or architecture tailored for effective OTT bypass fraud detection and classification.
- To develop and test a range of machine learning models, including advanced techniques, to enhance the accuracy and effectiveness of OTT bypass fraud detection and classification.
- To establish and apply suitable evaluation metrics to accurately assess and compare the performance of different machine learning algorithms, and to recommend improvements for practical implementation.

1.6 Significance of the Study

Ensuring effective detection of OTT bypass fraud is crucial for safeguarding telecommunication companies against financial losses, reputational harm, and unauthorized access to customer data, ultimately contributing to enhanced overall customer satisfaction.

This research is designed to achieve several key objectives. Firstly, it aims to uncover hidden fraudulent activities associated with OTT bypass fraud. Additionally, the study seeks to raise awareness among telecom operators, equipping them with the knowledge needed to identify and stop potential fraudsters engaging in these activities. Furthermore, the research endeavors to offer practical guidance for fraud prevention, providing anti-fraud managers and employees with valuable insights into effective strategies for mitigating the impact of OTT bypass fraud. In a comprehensive evaluation, this study will assess the traditional fraud management systems used by telecom companies. Additionally, the research aims to advance the field by proposing and

testing customized machine learning algorithms, offering valuable insights for the development and implementation of effective fraud detection systems specifically designed for OTT bypass fraud.

1.7 Scope and Limitation of the Study

This study's scope is specifically concentrated on identifying and detecting OTT bypass fraud within the broad category of telecom fraud. Although there are many different kinds of telecom fraud, the study specifically focuses on OTT interconnect bypass fraud detection. The investigation includes applying several techniques and instruments designed especially to detect OTT bypass fraud. The selected strategy applies machine learning methods to customer data record (CDR) data. The study will utilize Ethiopia Telecom's CDR data as its major data source. This purposeful decision makes it possible to focus on OTT bypass fraud detection in the unique setting of Ethiopian Telecom.

1.8 Organization of the study

The structure of the paper is as follows: Chapter Two delves into telecom fraud, with a particular emphasis on interconnect bypass fraud, exploring its effects on revenue and network security. It also reviews related literature to identify existing gaps and the need for improved detection techniques. Chapter Three outlines the research methodology, including data collection and preprocessing, the selection of machine learning algorithms, and the metrics used for evaluating model performance. Chapter Four presents the architecture of the proposed solution, detailing system design, environment configuration, and implementation steps, with code snippets for various stages of model development. Chapter Five discusses the experimental findings, comparing the performance of machine learning models such as Random Forest (RF), Support Vector Machine (SVM), and Logistic Regression (LR), and evaluates the importance of feature selection and preprocessing. Finally, Chapter Six concludes by summarizing the main results, discussing the implications for telecom fraud detection, and offering suggestions for future research, including potential improvements to the model and its relevance to industry practices.

CHAPTER TWO

2. LITERATURE REVIEW AND RELATED WORK

2.1 Telecommunication Fraud

Telecommunication fraud, an ever-evolving challenge within the telecommunications industry, poses a substantial threat to operational integrity and financial stability(Kinfmichael, 2021). This illicit activity involves the unauthorized use of telecom services to circumvent legitimate payment obligations. Among the myriad forms of telecommunication fraud, a particular focus is directed toward Over-the-top (OTT) bypass fraud (Sahin et al., 2017). This sophisticated form of fraud leverages internet-based communication services, such as WhatsApp, Viber, and Skype, to exploit vulnerabilities in call termination fee structures. The impact of telecommunication fraud reverberates through the industry, leading to significant revenue losses and compromising the quality of service. (Kinfmichael, 2021)

The telecommunication system, a transformative invention, has become crucial for global development and governance. Unfortunately, its introduction lacked a robust security focus Despite its significance, the security of these systems has not been well-managed, posing a growing threat to their operation.(Wang, 2022)

A 2015 survey of telecom service providers estimated that fraud costs the industry $38.1 billion annually, or 1.69% of global revenue. In addition to the financial losses, fraud can also damage reputations and disrupt services, which can have devastating consequences given the reliance on telecommunications networks by millions of users worldwide. Consumers are also victims of telecom fraud, with the United States Federal Trade Commission receiving an average of 400,000 complaints per month (Commission, 2013).

Telecom fraud poses a significant threat to both telecom providers and their customers, with potentially severe consequences. It is relatively easy to carry out, as most attacks can be executed remotely with minimal equipment and technical knowledge. Detecting and preventing telecom fraud is challenging due to the vast amount of traffic and the wide range of services involved (Sahin et al., 2017; Veloso et al., 2020)..

2.2 OTT Bypass Fraud

OTT bypass fraud is a significant concern for telecom companies due to the widespread use of OTT service providers. Fraudsters exploit these platforms by routing international calls through servers in countries with lower termination fees, avoiding the charges imposed by telecom operators in the destination countries (Djomadji et al., 2023). This type of fraud, also known as OTT hijacking, occurs when a call initiated as a regular phone call is terminated on an OTT application installed on the recipient's device. The rerouting of the call is carried out by telephone transit operators in collaboration with the OTT service provider, without the knowledge or consent of the caller or the receiver's telecom provider (Tewodros, 2018).

OTT services operate on top of data networks, beyond the control of internet service providers, delivering voice, text, and video content using packet-switching technology (Sawe, 2016). Popular applications such as Viber, Skype, WhatsApp, and Google Messenger are widely used to provide these OTT services, often free of charge. Revenue for OTT service providers typically comes from advertisements, in-app purchases, and subscriptions (Sujata et al., 2015).

The global adoption of smartphones and the availability of fast mobile internet have significantly increased the use of OTT services (Sahin & Francillon, 2016). Due to their ease of use, content flexibility, and lower costs, users increasingly prefer OTT services over traditional carriers. This shift threatens the revenue of telecom providers from voice, SMS, and multimedia services, and adds to data traffic congestion on their networks, which outweighs the revenue gains from data usage (Kinfmichael, 2021).

Some OTT applications require user accounts for registration, while others use the user's MSISDN (Mobile Station International Subscriber Directory Number). For instance, Skype and Yahoo Messenger require accounts, whereas Viber and Tango use MSISDNs for registration. OTT bypass fraud is often conducted through applications that rely on MSISDNs during registration (Kinfmichael, 2021).

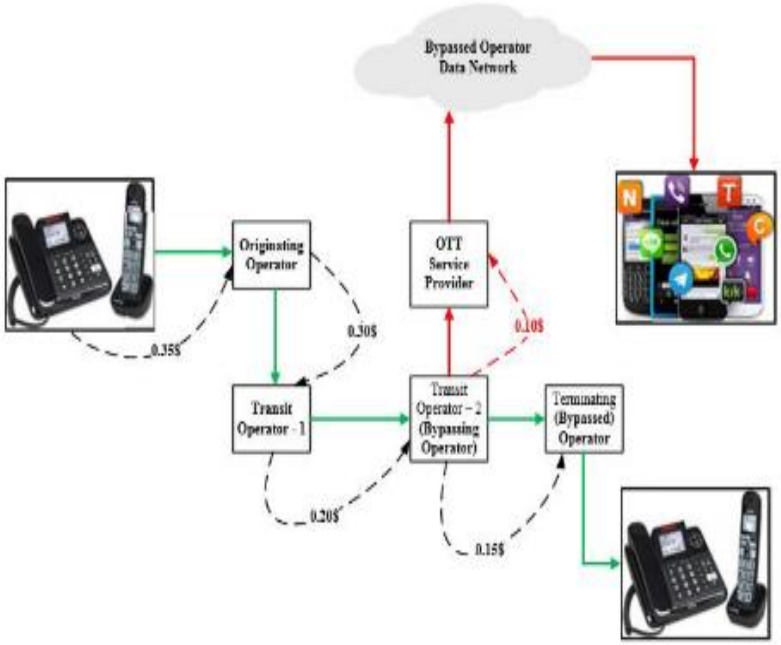


Figure 2. 1 OTT bypass fraud scenario. (Sahin & Francillon, 2016)

As illustrated in Figure 2.1, the caller initiates a regular call, assuming it will follow the standard telephony route and pay a specified fee to the originating operator. However, the calls are redirected to the OTT network (depicted by the red line) rather than the conventional cellular network (depicted by the green line), facilitated by a transit operation (Sahin et al., 2017)r. OTT service providers use the MSISDN to verify the receiver's online status and divert calls to OTT applications if the customer is online. The bypassed terminating operator incurs a loss in call termination fees since the calls deviate from the standard telephony route. Meanwhile, the OTT service provider and the transit operator, involved in rerouting the call, share the termination fee. In this fraudulent scenario, the terminating operator only receives a data usage fee from the receiver(Sahin & Francillon, 2016).

2.3 Understand the pattern with the flow of OTT bypass fraud.

OTT bypass fraud involves rerouting voice and messaging traffic over the internet (OTT) rather than the straight telecom network. This enables fraudsters to avoid the billing system of telecom operators and escape payment of termination fees.

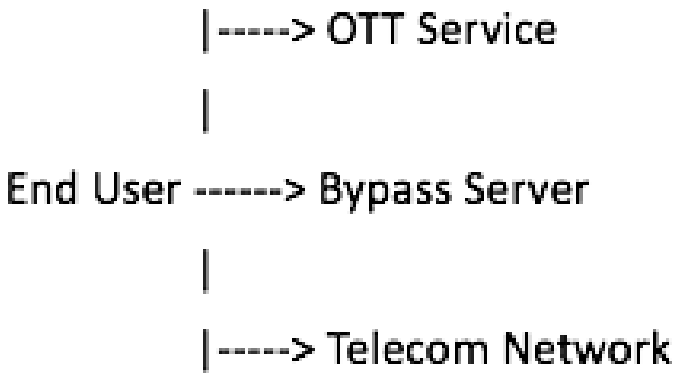


Figure 2. 2 Network flow diagram

1. The fraudster gains access to a third-party provider offering more cheap international voice and messaging services over the Internet.

2. Using tactics such as SIM card manipulation, VoIP gateways, call-routing overrides, phishing, and spoofing, the fraudster redirects voice and messaging traffic over the internet.

3. By bypassing the traditional telecom network and the operator's billing system, the fraudster manages to avoid paying termination fees to the telecom operator.

4. The end-user may remain unaware that their voice and messaging traffic is being rerouted over the internet, mistakenly continuing to use the third-party provider's service, thinking they are still on the telecom operator's network.

5. The telecom operator experiences significant revenue losses as a result of the fraudulent traffic being directed over the internet instead of through their network.

Fraudsters employ diverse techniques for executing OTT bypass fraud, including:

- **SIM card manipulation:** Using modified or cloned SIM cards, fraudsters redirect voice and messaging traffic over the internet, bypassing the conventional telecom network.
- **VoIP gateways:** By utilizing VoIP (Voice over Internet Protocol) gateways, fraudsters convert voice traffic from the traditional telecom network into internet-based traffic, enabling them to circumvent the operator's billing system.

- **Overriding call-routing:** Fraudsters can reprogram network equipment or deploy rogue signaling nodes to override call-routing, rerouting voice traffic over the internet.
- **Phishing:** Through phishing attacks, fraudsters acquire user credentials, allowing them unauthorized access to customer accounts and rerouting traffic over the internet without the user's awareness.
- **Spoofing:** Fraudsters can manipulate caller identification information, creating the illusion that calls originate from a legitimate source while rerouting traffic over the internet.

Using an above-mentioned mechanism, fraudsters are constantly adapting their tactics to evade detection and carry out OTT bypass fraud. To combat this type of fraud, telecom operators need to implement robust fraud detection systems and regularly monitor traffic patterns to detect and prevent fraudulent activity. They also need to collaborate with industry bodies, regulators, and law enforcement agencies to share information and develop new strategies to combat OTT bypass fraud.

2.4 Possible Consequences of OTT Bypass

Until recently, telecom operators relied heavily on voice calls and SMS as their primary revenue streams, with data services lagging behind. However, unlike their swift adaptation to previous disruptive innovations such as the internet boom and the expansion of mobile communications in the 1990s, telecom companies have been slower to respond to the latest challenge: Over-the-Top (OTT) service providers. It is widely accepted that OTT services are increasingly impacting telecom revenue from voice and messaging services. Additionally, their influence on data revenue and the growing demand for mobile data traffic have become crucial factors that telecom operators must now consider (Sujata et al., 2015)

2.4.1 Impact of OTT Services on Telecom Revenues

The global telecommunications industry has widely recognized the significant impact of Over-The-Top (OTT) services on telecom operators' revenues. Shifting consumer preferences and advancements in technology have led to a decline in traditional income sources for operators. Various studies, including Inform a's World Cellular Revenue Forecasts and Spirit DSP's report on the future of voice, reveal that the rise of OTT messaging apps and VoIP technologies is causing a notable reduction in SMS and voice revenues. For instance, global SMS revenues, which stood at US$120 billion in 2013, are expected to fall to US$96.7 billion by 2018(Farooq & Raju, 2019).

Similarly, global voice revenues, valued at $970.4 billion in 2012, are projected to drop to $799.6 billion by 2020, with around $479 billion of this loss attributed to VoIP(*What Is OTT and How Is It Impacting Telecom Service Providers? - Carritech*, n.d.). The OTT VoIP market is experiencing rapid growth, estimated at 20 percent annually, and is predicted to generate 1.7 trillion minutes of usage by 2018, leading to $63 billion in lost revenue(Farooq & Raju, 2019). These trends underscore the significant financial challenges facing telecom operators, emphasizing the need for strategic adjustments to thrive in the evolving telecommunications environment (OTT App Use Undermining SMS Revenue - Telecoms.Com, n.d.).

2.4.2 Impact of OTT Players on Data Traffic

The surge in OTT services has not only impacted telecom operators' voice and messaging revenue but has also led to a significant rise in data traffic, resulting in severe network congestion. One of the primary drivers of this data traffic surge is the increasing consumer appetite for video content (Sujata et al., 2015).

According to a Cisco study, mobile data traffic was projected to grow by 61% annually between 2013 and 2018, increasing from 1.5 exabytes to 15.9 exabytes per month. This growth was largely fueled by mobile video consumption, which was expected to rise from 633 PB to 9103 PB per month, with a CAGR of 70%. Such growth created immense pressure on telecom networks, with signaling traffic surpassing data traffic by 40%. To manage this demand, telecom operators were compelled to invest in expanding their network capacity, including spectrum and small cells (Pepper, 2013).

Heavy Reading's report, "Internet TV, Over-the-Top Video, & the Future of IPTV Services," highlights the growing threat telecom operators face from internet service providers and OTT video platforms. OTT providers can utilize existing IP networks without significant capital investment, while telecom companies fear that OTT video services will diminish the value of their VoIP offerings, relegating them to mere providers of broadband infrastructure(Gudimalla et al., 2019).

The widespread adoption of OTT services has been driven by their cost-effectiveness, convenience, wide range of features, and abundant content, all easily accessible via smartphones and mobile internet. These factors have collectively contributed to a decrease in traditional telecom

revenue from voice and messaging services. To stay competitive, telecom operators must adapt to these market shifts and leverage technological advancements to offer enhanced value to customers (Sujata et al., 2015)

2.5 Techniques used by Telecom Operators

2.5.1 Test Call Analysis

Telecom operators can use different commercially available Test Call Generation (TCG) platforms to make international test calls to detect OTT bypass fraud. By making repetitive calls from the networks of other operators, telecom operators can identify the transit operators that are redirecting the calls to OTT networks. Once the fraudulent transit operators have been identified, the telecom operators can take corrective actions (Tewodros, 2018).

One of the main problems with the Test Call Analysis is that it can be very time-consuming and resource-intensive. Telecom operators need to make a large number of test calls from different operators in order to identify the transit operators that are redirecting calls to OTT networks. This can be a very expensive and time-consuming process, especially for large telecom operators. Another problem with the Test Call Analysis is that it can be easily circumvented by fraudsters.

TCG systems are accessible for $1–10 per test call and are also utilized for QoS evaluation. They aren't made especially to identify OTT bypass fraud, though. The bypassed calls found in test calls can be the result of different interconnect bypass frauds in addition to OTT bypass fraud. In a recent study, used two TCG platforms to perform 15,872 test calls in European countries and found that 83% of the calls were fraudulently obtained by OTT bypass (Sahin & Francillon, 2016).

2.5.2 Network Traffic Analysis

OTT bypass fraud is also detected by employing Deep Packet Inspection (DPI) techniques to analyze the data flow. This approach does provide certain difficulties, though. The first is the matter of network neutrality, which requires operators to handle all data traffic equally, regardless of the content (Sawe, 2016). The inability of DPI techniques to collect only OTT bypass traffic is the second challenge. The other difficulties with network traffic analysis are the various encryption techniques and proprietary protocols employed by the over-the-top (OTT) service providers (Tewodros, 2018)

2.5.3 Statistical-based Techniques

Statistical-based classification techniques are a promising approach for OTT bypass fraud detection. They use machine learning algorithms and packet flow features to classify traffic, analyzing payload-independent traffic attributes such as byte frequencies, packet length, and packet inter-arrival time. They classify traffic based on protocol communication patterns instead of payload content, which offers several advantages over port-based and payload-based techniques.

Statistical-based classification techniques are more accurate, have faster computation time, and preserve data privacy. They are also more robust to changes in the network environment and can detect a wider range of OTT bypass fraud techniques. Overall, statistical-based classification techniques are a powerful tool for OTT bypass fraud detection (Tewodros, 2018).

2.6 Machine Learning Algorithm for Fraud Detection

Machine learning(ML), which is a subfield of the larger field of artificial intelligence (AI), is the process of acquiring knowledge or expertise from experience.(Shanthamallu et al., 2017) Machine learning is a computational approach that uses certain guidelines and rules to extract meaningful concepts from vast amounts of data to provide services and information. But those ideas were not created by computer programmers (Shalev-Shwartz & Ben-David, 2013) (Shanthamallu et al., 2017) Through the use of experience sample data, machine learning algorithms can gather knowledge or information without the need for sequential instruction. Machine learning was developed to carry out complex tasks that are beyond the scope of human expertise (Shalev-Shwartz & Ben-David, 2013). Adaptive nature solutions are required for machine learning algorithms to tackle these challenging problems. Two further reasons machine learnings are preferred over other programming languages are fraud detection from any transaction and forecasts. Machine learning algorithms are used for a variety of tasks, including predicting telecom Business Support System (BSS) failures, identifying telecom frauds (Tewodros, 2018), creating customer churn prediction models that identify potential customers who might switch telecom service providers by analyzing customer satisfaction levels, and creating fraud detection systems that identify anomalous traffic (Tewodros, 2018) (Kinfmichael, 2021).

2.6.1 Supervised Learning

Supervised machine learning techniques use a labeled training dataset, and after training is over, the generated model can predict the supplied class labels for new datasets that were not used in the training process. Classification and regression algorithms are the core class of supervised learning algorithms. The K-Nearest Neighbors Classifier, Decision Trees, Naive Bayesian Classifiers, Rule-Based Classifiers, Neural Networks, Linear Discriminant Analysis, and Support Vector Machines are a few instances of supervised learning methods (Shalev-Shwartz & Ben-David, 2013).

2.6.2 Unsupervised Learning

Since unsupervised machine learning techniques do not use labeled datasets for model training, model evaluation is a challenge. Using an unlabeled input dataset, clustering is a popular density estimate technique that attempts to determine the number of clusters or groupings. Two popular methods for unsupervised machine learning are K-Means and Gaussian Mixture Model (GMM) (Shalev-Shwartz & Ben-David, 2013; Rehman et al., 2024).

2.6.3 Deep Learning

In the near future, deep learning algorithms will be included in many different applications related to human aid. A machine is more likely to be able to test its hypotheses using deep learning methods. By simulating the neuronal connections found in the human brain, most deep learning techniques enable the integration of artificial intelligence into computer systems. This improves operational accuracy and speed for a variety of crucial operations. Currently, computers are built to carry out particular functions. In recent times, scientists have endeavored to provide a computer system with artificial intelligence, enabling it to do autonomous analysis and work. Artificial intelligence is used in many robotic applications to solve difficulties (Ahmad et al., 2022).

CNNs and RNNs are examples of deep learning models that notice things in data that are not visible to the human eye. By capturing the passage of time in sequences and deriving insights from image pixels, they unveil intricate relationships that are hidden from view by conventional models. This provides extremely accurate forecasts for sectors such as banking, language, telecom, and medicine; however, caution and high-quality data are still needed. Deep learning is the key for individuals who can read the whispers in the data; those people will rule the future (Ahmad et al., 2022).

2.7 Related works

By reviewing journals, publications, periodicals, and the Internet, it was possible to gain a thorough understanding of this research topic, bypassing fraud detection. Numerous scholarly research articles and investigations on the identification of telecom fraud and telecom fraudsters' effects on telecom firms have been carried out. One of the most prevalent telecom frauds that has severely impacted telecom firms is OTT Bypass fraud. As mentioned below, numerous investigations have been carried out in an effort to develop an intelligent system for OTT Bypass fraud detection.

The research by Kinfmichael Wubalem (2021) employs semi-supervised machine learning algorithms, specifically Decision Tree (DT) and Collective Filter, to detect MSISDN-based OTT bypass fraud. Data collection and preprocessing are conducted using Wireshark and WEKA, focusing on both labeled and unlabeled datasets for comprehensive training and evaluation. The performance of these algorithms is assessed using ten-fold cross-validation, a method where the dataset is divided into ten parts, with each part serving as both training and testing data in turn. The results show that the Decision Tree algorithm excels with a remarkable 99% accuracy in detecting OTT bypass fraud, significantly outperforming the Collective Filter algorithm, which achieves 91% accuracy. This demonstrates the superior efficacy of the Decision Tree in classifying network traffic related to OTT fraud. The study concludes that machine learning, particularly through the Decision Tree algorithm, can greatly enhance the detection of OTT bypass fraud, providing a valuable tool for telecom companies to mitigate revenue losses from such fraudulent activities. However, the authors note the importance of expanding the feature set beyond the four tested attributes to improve scalability and overall performance(Kinfmichael, 2021).

In order to detect fraud, another study (Raghavan & Gayar, 2019) evaluates several deep learning and machine learning models using distinct data sets. Determining which approaches would be most effective for different kinds of datasets was the main objective of this investigation. This could potentially help practitioners and businesses better understand how different methods operate on different types of datasets, as many corporations invest in innovative strategies to increase their bottom line. The work is limited to fraud detection in supervised learning methods, even though it provides guidance on choosing the right machine learning and deep learning algorithms depending on the provided data. Though they seem attractive and produce good results,

CNN, KNN, and Random Forest do not work well in dynamic contexts. Due to the fact that fraudulent practices fluctuate and are difficult to detect over time (Raghavan & Gayar, 2019).

Tewodros (Tewodros, 2018) conducted pioneering research on detecting Over-The-Top (OTT) applications based on Mobile Station International Subscriber Directory Number (MSISDN) using machine learning algorithms applied to network traffic data. The study employs various machine learning algorithms, including Adaptive Booster (AdaBoost) + J48, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), and Support Vector Machine (SVM). The efficacy of these algorithms is evaluated using metrics such as the F-measure, confusion matrix, classification accuracy, and ROC curve. Additionally, attribute significance is assessed through correlation and information gain ratio methods. Despite these advancements, the study acknowledges limitations due to the restricted number of network packet attributes and suggests that future research should address these constraints to improve data transmission and processing parameters.

Kasra Babaei's research delves into the intricacies of fraud detection in the telecom industry, examining various fraud types and challenges like data quality and rule-based limitations. The study underscores the significance of high-quality data for effective detection, proposing promising machine learning solutions. An SVM model achieved a remarkable 99.06% accuracy in detecting SIMboxing fraud, utilizing carefully selected features from call data. Furthermore, the research showcases a real-time detection system analyzing call audio, successfully identifying 87% of fraudulent calls within 30 seconds without false positives. This offers a valuable tool for telecom operators to combat fraud swiftly. However, the ever-evolving landscape of fraud, with new techniques like social engineering and deepfake-based scams emerging regularly, demands continuous adaptation and expansion of fraud detection strategies(Babaei et al., 2019).

Ighneiwa et al. collaborated with a Tier 1 mobile operator in Libya for their (Ighneiwa & Mohamed, 2017). The study's objective was to raise awareness about SIM-box fraud to mitigate revenue losses, denial of service, diminished service quality, and network congestion. The experiment utilized Call Detail Record (CDR) data, employing two supervised algorithms, SVM and Decision Trees (Random Forest), with accuracy and precision serving as the evaluation

metrics for model performance. Additionally, the author employed SVM and Artificial Neural Network (ANN) algorithms in their study.

The paper (Raghavan & Gayar, 2019) provides a comprehensive overview of various machine learning and deep learning methods for fraud detection, including K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Naïve Bayes, and Random Forest, each noted for their specific strengths. Deep learning techniques like Convolutional Neural Networks (CNN), Autoencoders, Deep Belief Networks (DBN), and Restricted Boltzmann Machines (RBM) are also explored. The study highlights the effectiveness of ensemble methods, particularly combining KNN, SVM, and Random Forest, which yielded the best results in terms of evaluation metrics like Matthews Correlation Coefficient (MCC) and Area Under the Curve (AUC). Overall, the paper emphasizes the importance of using a combination of techniques to improve the accuracy and reliability of fraud detection models(Raghavan & Gayar, 2019).

The Paper (Li et al., 2018)presents a structured approach for detecting telecom fraud using two modules: a Machine Learning Module and a Template Detection Module. The Machine Learning Module employs a Support Vector Machine (SVM) to classify users based on Call Detail Records (CDRs), identifying potential fraudsters. The Template Detection Module uses a Finite State Machine (FSM) to further analyze the behavior of users flagged as suspicious by the SVM, improving detection by focusing on user actions. The method, tested on a real-world dataset, achieved a high detection accuracy of 93.56%. However, the limited scope of the data may impact the generalizability of the results. Despite this, the SVM and FSM framework shows strong potential for minimizing revenue losses and protecting subscribers from fraud.

In the research by Tesfaye, various machine learning algorithms, including Random Forest (RF), Artificial Neural Networks (ANN), and Support Vector Machine (SVM), were employed to detect SIM-box fraud, with RF highlighted for its high classification accuracy. The study applied two data aggregation techniques, finding that the Sliding Window (SW) mode significantly enhanced detection accuracy while managing detection delays. WEKA was used for training the algorithms and analyzing performance through metrics such as accuracy, confusion matrix, F-measure, and Receiver Operating Characteristic (ROC) curve. Extensive data preprocessing, including cleaning and integration, was conducted to ensure the quality of Call Detail Record (CDR) data. The results revealed that the RF classifier achieved a notable 96.2% accuracy with the SW aggregation mode,

surpassing ANN and SVM in performance. The decision tree algorithm was noted for its superior accuracy, while SVM faced challenges with longer detection delays, highlighting a trade-off between accuracy and speed(F. Tesfaye, 2020).

The paper (Hu et al., 2023)presents significant findings on the effectiveness of the proposed fraud detection method using Augmented Graph Neural Networks (GNNs). Extensive experiments were conducted on two real-world telecom fraud detection datasets, Sichuan and BUPT, both characterized by their imbalanced nature, which challenges traditional detection methods. The method's performance was evaluated using key metrics such as accuracy, area under the ROC curve (AUC), and F1 score, providing a comprehensive assessment of its ability to classify fraudsters versus non-fraudsters. Additionally, the proposed method effectively addressed two major challenges associated with GNNs: the graph imbalance problem, where traditional methods often fail to accurately classify minority classes, and other existing limitations in handling such imbalanced datasets, leading to improved classification performance(Hu et al., 2023).

Table 2. 1 Summary of Related Work

Author and Year	Type of Fraud	Techniques Used	Data Employed	Machine Learning /Models Applied	Research Gaps/Aspects Not Addressed
Kinfmichael Wubalem, 2021	MSISDN-based OTT bypass fraud	Semi-supervised ML algorithms, Decision Tree (DT), Collective Filter	Generated data collected with Wireshark and evaluated using ten-fold cross-validation	Decision Tree (99% accuracy), Collective Filter (91% accuracy)	- **Limited features**: Only 5 features used. - **Data limitation**: Lack of real-world data, reliance on generated data. - **Techniques**: Lack of ensemble methods or more advanced ML techniques.
Tewodros, 2018	OTT voice call fraud	Network traffic generation, feature selection	Controlled environment using Wireshark; converted to .ARFF format	Support Vector Machine (95.35% accuracy), AdaBoost + J48	- **Fraud type**: Limited to OTT voice calls, not focusing on MSISDN-based OTT applications. - **Algorithm testing**: Insufficient testing and comparison of multiple algorithms. - **Data limitation**: Lack of real-world data.
Kasra Babei, 2019	SIM-boxing fraud	SVM, real-time detection	Call data, features selected manually	Support Vector Machine (99.06% accuracy), real-time detection	- **Emerging fraud techniques**: Lack of focus on new fraud trends (e.g., deepfake, social engineering). - **Small dataset**: Limited dataset and features used.
Ighneiwa & Mohamad, 2017	Fraudulent transactions	Machine learning, deep learning	Compared large and small datasets,	SVM, CNN, ensemble approaches (SVM, RF, KNN), Autoencoders	- **Dataset size**: Small dataset affecting generalization. - **Feature analysis**: Lack of detailed feature importance.

			features unspecified		- **Fraud type**: General fraud detection, not specific to telecom fraud like OTT bypass.
Tesfaye, 2019	SIM-box (interconnect bypass) fraud	Data aggregation (Sliding Window mode), ML training via WEKA	Aggregated data using Sliding Window mode	Random Forest (96.2% accuracy), ANN, SVM	- **Fraud type**: Focus limited to SIM-box fraud, not generalized to other telecom fraud types.
					- **Real-time detection**: Continuous updates needed for real-time detection.
Xinxin Hu, 2023	Telecom fraud (general)	Graph imbalance technique, minority class focus	Real-world datasets (Sichuan, BUPT)	Accuracy, AUC, F1 score	- **Fraud type**: General telecom fraud focus, not specific to OTT bypass or SIM-boxing.
					- **Techniques**: Focused only on graph imbalance, lack of comparison with other advanced ML methods.

2.8 Baseline paper

The research by Kinfmichael Wubalem (2021) uses semi-supervised machine learning algorithms, specifically Decision Tree (DT) and Collective Filter, to detect MSISDN-based OTT bypass fraud. The study employs datasets collected and preprocessed with Wireshark and WEKA, and evaluates the algorithms using ten-fold cross-validation. While the Decision Tree algorithm achieves a high accuracy of 99% in detecting OTT bypass fraud, outperforming the Collective Filter algorithm with 91% accuracy, the study has a significant gap: it relies on generated data and a limited set of features, without using real network traffic data from Ethio Telecom. This lack of real-world data limits the insights into the effectiveness of the detection mechanisms.

CHAPTER THREE

3. RESEARCH METHODOLOGY

3.1 Overview

This section of the thesis presents the research method, along with discussion on research approach, data collection methods, metrics for research, testing and assessment process and algorithm. To explore explanatory research questions, we try to specify methodologies or models that are used in a suggested solution for OTT/Bypass fraud detection and classification. Then we came up with machine learning models and eventually tested their performance.

3.2 Overall methodologies of the research

Research methodology refers to the overall plan and processes used to conduct research and answer your research questions. It's the roadmap that guides you from your initial research idea to collecting and analyzing data, interpreting results, and drawing conclusions.

The methodology in a research study on OTT fraud detection typically outlines the systematic approach and procedures undertaken to conduct the investigation. In the context of OTT fraud detection, the methodology section may include the following components: To fully comprehend the suggested approach, problem identification, suggested solution, design, implementation, and testing procedures, as well as the current method, conduct thorough research on it.

The research focuses on addressing limitations in existing OTT fraud detection methods, such as inadequate performance and high complexity in algorithms like Decision Trees and SVMs. It utilizes machine learning models like Random Forest, SVM, and Logistic Regression to detect fraud using Call Detail Records (CDRs) from Ethio Telecom. The research covers problem identification, literature review, data collection, preprocessing, algorithm selection, model evaluation, and results. The study concludes with suggestions for future work to further enhance fraud detection methods by refining techniques and exploring new approaches.

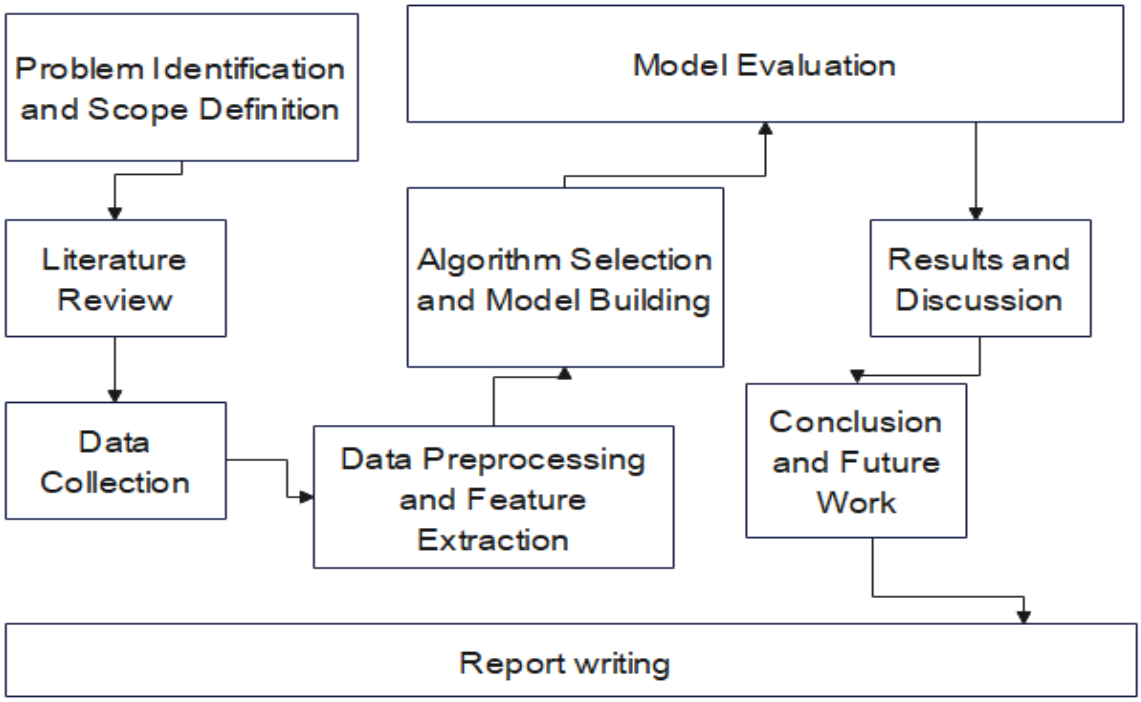


Figure 3. 1 Research methodology

3.3 Data Collection

When making calls through a telecom service provider, the call remains within the telecommunications network, and descriptive information about the call is stored as call detail records (CDRs). For this thesis, we obtained a dataset from Ethio Telecom, which included information from various mobile operator databases, including Cell ID and CDRs. our focus was on CDRs, which contain a subscriber's call history. To prepare. for training, we collected subscriber usage history and CDR data from the Ethio Telecom billing system. The CDRs generated from active customers' usage over a two-month period were stored in a database, processed, and outputted in the required training format.

When a subscriber initiates a call over the operator's network, a toll ticket is generated. This ticket contains comprehensive call information, including details such as the subscriber ID, called number, call duration, start and end times, destination location, and more. Specifically, this information is captured in what is known as a Call Detail Record (CDR). CDRs are produced on a per-call basis by telephone switches or other telecommunication equipment. They contain all the necessary data to describe the key characteristics of a telephone call or other telecommunication

transactions involving subscribers within the network. CDRs include fields required by billing systems to rate specific calls and bill the corresponding subscribers. These records provide a detailed account of each voice call, SMS, and Internet data usage transaction originating from and terminating on a subscriber's device. Fields within a CDR instance may encompass phone numbers involved in the call, the date and time of the call, call duration, the identity of the cell transmitting the call to the subscriber's phone, and other relevant information. Successful data preprocessing requires a deeper exploration of the meaning behind these records to gain a comprehensive understanding of the data.

3.3.1 Telecommunication Data

When a subscriber places a call over the operator's network, a toll ticket is generated. This ticket captures detailed call information, including the subscriber's ID, the number dialed, call duration, start and end times, destination, and other relevant data. This information is recorded in what is called a Call Detail Record (CDR). CDRs are generated for each call by telephone switches or other telecom equipment and contain essential data to describe the key characteristics of calls or other telecom activities involving subscribers. These records include fields necessary for billing systems to rate calls and charge subscribers accordingly. They provide a comprehensive account of each voice call, SMS, or data usage session initiated and received by a subscriber. Information in a CDR may include the phone numbers involved, the date and time of the call, its duration, the identity of the cell tower handling the call, and other pertinent details. Proper data preprocessing involves thoroughly understanding these records to gain meaningful insights into the data.

3.3.2 Call Detail Record

When a phone call is placed within a telecommunications network, the details of the call are recorded in a call detail record (CDR). The network generates and archives numerous such records for each call. Each CDR includes comprehensive information about the call, such as the phone numbers of both the caller and receiver, the date and time of the call, its duration, and the amount of SMS and data usage linked to each subscriber, including their billing information. CDRs are essential in the telecommunications industry as they provide valuable data for maintaining and enhancing network performance, monitoring call quality, analyzing usage trends, and managing billing processes.

3.4 Datasets and Their Descriptions

We've acquired a colossal dataset – two months' worth of CDRs from Ethio Telecom and public dataset. This rich data boasts 24 columns detailing call usage and 2 columns with cell tower information. Each column holds millions of rows, creating a data tsunami that overwhelms standard computers. While the information is invaluable for our research, its sheer size necessitates a robust approach for analysis. Given the substantial dataset from Ethio Telecom, containing millions of rows and numerous columns detailing call usage and cell tower information, we need a strategic approach for effective analysis. Adding to the complexity is the integration of public CDR data, which includes additional metrics such as previous network type, network type, call failure cause, packet loss, conversational MOS, and listening MOS.

Table 3. 1 CDR Data and Descriptions

No	Attribute	Description
1	DATA_DAY	Date and potential time of call or activity
2	CALLING_NUMBER	Number initiating the call
3	CALLED_NUMBER	Recipient of the call
4	OFF-PEAK_USG_MINUTE	Total call minutes during off-peak hours (evenings, weekends, holidays)
5	PEAK_USG_MINUTE	Highest resource usage (in minutes) during a call (typically during peak hours)
6	INTER_USAGE_MINUTES	Total call duration for calls made to different networks
7	CALL_DURATION	Total length of a call (in minutes)
8	CALL_FREQUENCY	Number of calls made by a subscriber within a specific period (daily, weekly, monthly)
9	PREVIOUS_NETWORK_TYPE	Network the subscriber was connected to before the current call
10	NETWORK_TYPE	Network type used for the call (e.g. cellular net.. wifi)
11	DATA_USAGE	Amount of data transferred during a data session (in megabytes or gigabytes)
12	CALL_FAILURE_CAUSE	Reason why a call failed to connect or dropped
13	PACKET_LOSS	Percentage of data packets not delivered during transmission

14	CONVERSATIONAL_MOS	Perceived speech quality during a call (conversation)
15	LISTENING_MOS	Perceived speech quality during a call (listening)
16	SUBSTR(T.SERV_NO)	A substring of the service number, likely used to anonymize or categorize the numbers.
17	SGMT_NAME	Segment name, possibly representing customer segmentation or market segment.
18	CUST_TYPE_NAME	Customer type name, indicating the type or category of the customer (e.g., residential, business).
19	ACTIVATE_DATE	The date when the customer's service was activated.
20	STATUS_NAME	The current status of the service (e.g., active, inactive, suspended).
21	LISTENING_MOS	Mean Opinion Score (MOS) for the quality of listening experience during calls.
22	DATA_REVENUE_ETB	Data revenue generated in Ethiopian Birr (ETB) from the customer's data usage.
23	ASL (AVERAGE SPEECH LEVEL)	Average speech level, indicating the average volume level of speech during calls.
24	RMS (ROOT MEAN SQUARE)	Root mean square value, a statistical measure of the magnitude of a varying quantity, used here for audio signal.
25	NOISE_PERCENT_VOICE	The percentage of noise present in the voice signal during calls.
26	CELL_ID	BTS id

3.5 Data Preparation and Analysis

3.5.1 Data preparation

The act of refining, converting, and structuring unprocessed data into a form that is suitable for analysis is known as data preparation, or data preprocessing. This stage holds essential significance in any data analysis or machine/deep learning algorithm, as the quality and suitability of the data directly influence the accuracy and effectiveness of the analysis. In the realm of intricate data analysis, data preparation plays a pivotal role in neural network modeling, exerting a substantial influence on the outcomes of a diverse array of complex data analysis endeavors (Koval, 2018).

Common data preparation tasks involve eliminating redundant or unimportant data, addressing missing values, converting data formats, standardizing or adjusting data ranges, and encoding categorical attributes. Additionally, it may encompass feature engineering, which involves creating new attributes from existing data to enhance the effectiveness of machine/deep learning models. Thus, data preparation plays a vital role in guaranteeing that the data utilized in analyses or machine/deep learning models is precise, comprehensive, and pertinent for model training.

3.6 Data Preprocessing and Feature Selection

3.6.1 Feature Correlation Analysis

Feature Correlation Analysis is a critical step in the process of feature selection for machine learning models. The goal is to identify a subset of features that significantly impact model performance, reduce unnecessary complexity, and avoid overfitting(Rehman et al., 2024). One widely used method for feature selection is the Pearson correlation, which measures the linear relationship between each feature and the target variable. The Pearson correlation coefficient, ranging from -1 to 1, quantifies the strength and direction of this relationship.

- **High absolute values** (closer to 1 or -1) indicate a strong relationship, meaning the feature is highly informative for predicting the target.
- **Low values** (closer to 0) suggest a weak or no relationship, implying that the feature may add little value or even noise to the model.

By focusing on features with strong correlations, we can improve model accuracy, minimize computational costs, and enhance the interpretability of results. Features with low or negative correlations are often discarded as they may be irrelevant or redundant. Pearson correlation, thus, helps prioritize key features during model training.

Feature selection is a dimensionality reduction technique aimed at improving model performance by eliminating irrelevant, redundant, or noisy features. This process involves selecting a subset of significant features from the original dataset, which often leads to enhanced learning accuracy, reduced processing costs, and better model interpretability Features that do not contribute to distinguishing between different classes or clusters are deemed unimportant and can be removed. The removal of such superfluous features typically does not impact the learning outcomes negatively; in fact, it can enhance model accuracy by reducing noise and computational inefficiencies caused by extraneous data (Jia et al., 2022)(F. Tesfaye, 2020)..

When dealing with a large number of input features, dimensionality reduction techniques become crucial. These techniques are generally categorized into two main types: feature extraction and feature selection. Feature extraction involves combining original features to create new, lower-

dimensional features, which may complicate the interpretation of the resulting features because they lack direct physical meaning(Jia et al., 2022). On the other hand, feature selection involves choosing a subset of the original features, preserving their original meanings and improving the readability and interpretability of the model (Kahsu, 2018). By retaining the most relevant features, feature selection helps in constructing more accurate and efficient models, reducing computational demands, and improving overall performance.

Extracting valuable insights from CDRs is crucial for combating OTT Bypass Fraud. However, not all attributes within a CDR hold equal weight in this fight. Let's delve into the key attributes that can expose fraudulent activity. First, the DATA_DAY attribute captures the date and potentially the time of the call or message, providing a crucial starting point for investigation (Ighneiwa & Mohamed, 2017). Subscriber identification comes next, with CALLING_NUMBER revealing the number initiating the call and CALLED_NUMBER identifying the recipient.

CDRs excel at revealing calling patterns. The OFF-PEAK_USG_MINUTE attribute exposes the total call minutes during off-peak hours (evenings, weekends, and holidays). Since OTT bypass services aim to dodge call charges, unusual calling patterns concentrated in these periods can be a red flag. Similarly, the PEAK_USG_MINUTE attribute reflects the highest resource usage during a call, which typically occurs during peak hours with congested networks and expensive rates. Analyzing these usage patterns can help identify potential bypass activity where fraudsters leverage OTT services to avoid peak charges(Sujata et al., 2015b; Veloso et al., 2020).

Beyond call volume, CDRs offer details about the calls themselves. The INTER_USAGE_MINUTES attribute highlights the total call duration for calls made to different networks. As OTT bypass calls often involve unfamiliar networks, analyzing this attribute can expose such anomalies. Additionally, the CALL_DURATION attribute indicates the total call length, with extremely short or long calls compared to a subscriber's usual patterns potentially signifying bypass fraud(Kouam et al., 2021).

Network information can also be revealing. The PREVIOUS_NETWORK_TYPE attribute identifies the network the subscriber was connected to before the current call. Frequent switching

could be a sign of bypass activity, especially for unknown operators (Veloso et al., 2020). The NETWORK_TYPE attribute specifies the network type used for the call. Analyzing call distribution across network types can help identify patterns associated with bypass fraud, as certain technologies might be favored for such calls(Sujata et al., 2015b; Veloso et al., 2020).

Data usage is another potential clue. The DATA_USAGE attribute reflects the amount of data transferred during a data session. Unusual spikes in data usage, particularly when not aligned with a subscriber's typical data consumption patterns, could indicate potential bypass activity where data is used for voice or video calls instead of traditional cellular services ((Ighneiwa & Mohamed, 2017).

Call quality metrics can also raise red flags. The CALL_FAILURE_CAUSE attribute identifies the reason why a call failed to connect or dropped. A high frequency of call failures, especially for calls to specific numbers or networks, might be a sign of bypass activity with less reliable call quality. The PACKET_LOSS attribute represents the percentage of data packets not delivered during transmission. Abnormally high packet loss can indicate network congestion or connection issues, which could be more prevalent in OTT bypass calls due to reliance on internet infrastructure. CONVERSATIONAL_MOS and LISTENING_MOS attributes measure the perceived speech quality during a call, for the conversation itself and listening, respectively. Lower scores in these metrics could suggest bypass activity with poor call quality often associated with OTT bypass services ((Ighneiwa & Mohamed, 2017; Sujata et al., 2015b; Veloso et al., 2020).

Table 3. 2 Selected Features

No	Attribute	Description	Significance for Fraud Detection
1	DATA_DAY	Date and potentially time of call or message	Provides a starting point for investigation.
2	CALLING_NUMBER	Number initiating the call	Identifies the subscriber suspected of fraud.
3	CALLED_NUMBER	Recipient of the call	Helps identify patterns in targeted numbers.
4	OFF-PEAK_USG_MINUTE	Total call minutes during off-peak hours	Unusual concentration in off-peak periods suggests a potential bypass to avoid call charges.

5	PEAK_USG_MINUTE	Highest resource usage during a call (typically during peak hours)	Analyzing usage patterns can help identify attempts to avoid peak charges through OTT services.
6	INTER_USAGE_MINUTES	Total call duration for calls made to different networks	Highlights calls to unfamiliar networks, a potential sign of OTT bypass.
7	CALL_DURATION	Total length of a call	Extremely short or long calls compared to usual patterns can be suspicious.
8	CALL_FREQUENCY	Number of calls made by a subscriber within a specific period (daily, weekly, monthly)	An unusual increase in call frequency, especially during off-peak hours or towards specific numbers, can suggest bypass activity.
9	PREVIOUS_NETWORK_TYPE	The network the subscriber was connected to before the call	Frequent switching, especially to unknown operators, could indicate bypass activity.
10	NETWORK_TYPE	Network type used for the call (e.g., cellular, WIFI)	Analyzing call distribution across network types can reveal patterns associated with bypass fraud (certain technologies might be favored).
11	DATA_USAGE	Amount of data Used during a data session	Unusual spikes, particularly when not aligned with typical data consumption, suggest the potential use of data for voice/video calls instead of cellular services.
12	CALL_FAILURE_CAUSE	The reason why a call failed to connect or dropped	High frequency of call failures, especially to specific numbers/networks, might indicate unreliable call quality associated with bypass services.
13	PACKET_LOSS	Percentage of data packets not delivered during transmission	Abnormally high packet loss can suggest network congestion or connection issues, more prevalent in

			OTT bypass calls due to reliance on internet infrastructure.
14	CONVERSATIONAL_MOS	Perceived speech quality during a call (conversation)	Lower scores could indicate poor call quality often associated with OTT bypass services.
15	LISTENING_MOS	Perceived speech quality during a call (listening)	Lower scores could suggest poor call quality often associated with OTT bypass services.

3.6.1 Preprocessing Data

Real-world databases, owing to their sheer size and complexity, often encounter a myriad of challenges that compromise data quality. Among these challenges are noise, which refers to irrelevant or erroneous data, missing values that can result from incomplete data entry or data corruption, and data inconsistency stemming from discrepancies or conflicts in the stored information. These databases typically draw from a multitude of sources, each with its structure, format, and level of reliability, rendering them heterogeneous. This heterogeneity further exacerbates the difficulties in ensuring data accuracy and consistency(Koval, 2018). To mitigate these challenges and enhance data quality, it is imperative to employ robust data processing techniques.

- ✓ **Data Cleaning:** Eliminates noise and rectifies discrepancies in the dataset to ensure accuracy and consistency.
- ✓ **Data Integration:** Merges data from multiple sources into a unified dataset, reducing heterogeneity and facilitating comprehensive analysis.
- ✓ **Data Reduction:** Minimizes dataset volume by aggregating, removing redundancy, or clustering, improving processing efficiency while preserving essential information.
- ✓ **Data Transformations:** Normalizes data to a standardized range, improving comparability and compatibility with analytical techniques.

These techniques work synergistically to enhance data quality and usability, enabling organizations to derive valuable insights and make informed decisions.

3.6.2 Data Cleaning

The process of data cleaning involves detecting and rectifying inaccuracies, inconsistencies, and errors in original data sets to ensure they are suitable for analysis, modeling, and decision-making purposes. It is an integral part of preparing data for use in machine or deep learning applications, involving tasks like filling in missing values, ensuring consistency, removing noisy data, and deduplication records to prevent bias. Successful data cleaning requires careful attention and effort to avoid introducing irrelevant information, especially considering the data's diverse origins and the need for consistency in attributes like size, data type, and format across all data sources (Babaei et al., 2019).

3.6.3 Data Scaling

We have selected standardization for preprocessing our dataset. Standardization rescales the data to have a mean of 0 and a standard deviation of 1, which is particularly beneficial for algorithms that assume a standard normal distribution, such as linear regression, logistic regression, and support vector machines. This method ensures that each feature contributes equally to the model by eliminating biases caused by differing scales. Additionally, standardization handles outliers more effectively than normalization. By using standardization, we aim to improve the model's learning efficiency and convergence, ultimately improving the overall performance of our machine learning algorithms.

3.6.4 Data Aggregation

The process of aggregating and condensing extensive data from various sources into a more manageable and meaningful format is referred to as data aggregation. This involves consolidating individual data points into summary statistics or metrics, such as averages, counts, sums, or percentages. By simplifying the complexity and volume of information, data becomes easier to analyze and comprehend. Data aggregation facilitates the swift and efficient extraction of insights that can be leveraged to make informed decisions or discern trends and patterns. When dealing with collected CDR data, data aggregation is a vital preprocessing task aimed at providing comprehensive insights into specific users (F. Tesfaye, 2020). While a raw CDR contains a single record reflecting the activities of a particular user, understanding user behavior based on a single CDR record is challenging. Consequently, aggregating multiple CDR records is essential to gain a complete understanding of user behavior.

During the process of data aggregation, caution must be exercised, as the proportion of legitimate calls to fraudulent calls in CDR usage is not uniform. Therefore, it is crucial to aggregate the important data features (attributes) that are utilized in detecting OTT Bypass fraud while considering this aspect.

3.7 Algorithm Training

Upon completing the crucial steps of data preprocessing, feature selection, and aggregation, the subsequent pivotal phase entails training and constructing a detection model utilizing the chosen algorithm, as illustrated in the earlier chapter on machine learning detection techniques. This process engages prominent machine learning algorithms such as LR, SVM, and Random Forest to formulate the models. Distinct training methodologies, including K-fold cross-validation and separate test data, are deliberately employed to train these models effectively.

Among these methodologies, K-fold cross-validation emerges as the prevalent choice. It partitions the dataset into k equal folds, allowing the detection algorithm to undergo training with k-1 folds and validation with the remaining fold. This iterative approach ensures robust model evaluation by systematically rotating the test fold across each fold in the dataset. Ultimately, the cumulative average error across all training and testing iterations furnishes comprehensive insights into model performance (F. Tesfaye, 2020).

Conversely, separate test data presents an alternative model training technique. In this approach, the dataset is split into distinct training and testing subsets, with no fixed ratio for labeled data allocation. The suitability of this partitioning hinges on the adequacy of data available for both training and testing purposes. While a common practice is to allocate 80% of the data for training and reserve the remaining 20% for testing, adjustments may be warranted based on dataset size and characteristics to ensure sufficient data for meaningful training and evaluation (F. Tesfaye, 2020).

In the context of this research endeavor, a balanced utilization of both K-fold cross-validation and separate test data methodologies is advocated. By leveraging these diverse training techniques, the research activities achieve robust and dependable insights into the performance of the selected

algorithms across varied datasets, thereby contributing to the advancement of machine learning applications within the targeted domain.

3.8 Evaluation Metrics for Detection Algorithm Performance.

The central aim of this study revolves around employing machine learning techniques for the improvement of the detection of OTT Bypass fraud. With data collection completed, the preprocessing phase is managed, paving the way for model training and testing. To assess the efficacy of the algorithms, it becomes imperative to subject the model to testing using dedicated test data. In evaluating a given deep-learning model, a spectrum of evaluation metrics comes into play. These include, but are not limited to, the confusion matrix, classification accuracy, F-measure, Recall, Root Mean Squared (RMS) error, and Receiver Operating Characteristic (ROC) curve. By utilizing these evaluation metrics, a comprehensive understanding of the model's performance can be gleaned, enabling informed decisions regarding its suitability for OTT Bypass fraud detection.

3.8.1 Confusion Matrix

The confusion matrix serves as a tool for evaluating the effectiveness of supervised machine/deep learning algorithms. It examines classification outcomes, reflecting the algorithms' performance in distinguishing between different classes. This 2X2 matrix includes True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN) values, offering insights into the algorithm's classification accuracy.

Table 3. 3 conceptual confusion matrix

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

True Positive (TP): This represents the number of instances correctly classified as normal (legitimate) telecommunications traffic. In the context of OTT bypass fraud detection, TP indicates the successful identification of instances where the telecommunications traffic is indeed legitimate and routed through authorized channels.

False Positive (FP): This denotes the number of instances that belong to OTT bypass fraudulent activity but are incorrectly classified as normal telecommunications traffic. FP instances represent false alarms where legitimate traffic is incorrectly flagged as fraudulent, potentially leading to unnecessary investigations or disruptions in service.

False Negative (FN): FN represents the number of instances that actually belong to legitimate telecommunications traffic but are incorrectly classified as OTT bypass fraudulent activity. These instances represent missed detections, where fraudulent activity goes undetected, allowing OTT bypass fraud to occur without detection.

True Negative (TN): TN indicates the number of instances correctly classified as OTT bypass fraudulent activity. In the context of OTT bypass fraud detection, TN signifies the successful identification of instances where telecommunications traffic is indeed fraudulent, routed through unauthorized OTT channels to circumvent regulations or costs.

In summary, True Positive (TP) and True Negative (TN) instances represent successful detections of legitimate and fraudulent telecommunications traffic, respectively. On the other hand, False Positive (FP) and False Negative (FN) instances indicate errors in detection, resulting in false alarms or missed detections. Striking a balance between minimizing false alarms (FP) and missed detections (FN) is essential for an effective OTT bypass fraud detection system. This balance ensures accurate identification of fraudulent activity while minimizing disruptions to legitimate telecommunications traffic, thereby improving the overall reliability and efficiency of the detection system.

3.8.2 Detection Accuracy

Detection accuracy evaluates the proportion of accurately detected instances, encompassing both the legitimate (class 'N') and fraudulent (class 'Y') categories within the entire dataset. The flowing mathematical expression illustrates the percentage of instances correctly detected.

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

Equation 3. 1 Accuracy formula

3.8.3 F-Measure

The F-measure, originating from the harmonic mean of precision and recall, is a well-rounded measure of a detection model's performance. Calculated using the provided mathematical equation, this metric considers both precision and recall. Precision evaluates the accuracy of positive predictions in comparison to all instances classified as positive, whereas recall assesses the completeness of positive instances correctly identified among all actual positives. By integrating these two metrics, the F-measure offers a comprehensive assessment of the model's ability to precisely detect positive instances while effectively managing false positives and false negatives.

$$\text{F-Measure} = \frac{2 * Precision * Recall}{Precision + Recall}$$

Equation 3. 2 F-Measure formula

3.8.4 Recall

Recall quantifies the proportion of accurately identified instances classified as Normal (class 'N'), determined by dividing the number of correctly identified instances of Normal (class 'N') by the total number of instances classified as Normal (class 'N') and those mistakenly classified as Fraudulent (class 'Y'). The same method applies to compute Recall for Fraudulent (class 'Y').

$$\text{Recall} = \frac{TP}{TP + FP}$$

Equation 3. 3 Recall formula

3.8.5 ROC Curve

The True Positive Rate (TPR) and False Positive Rate (FPR) are visualized in the Receiver Operating Characteristic (ROC) curve. The axes represent FPR on the X-axis and TPR on the Y-axis. An ideal algorithm is indicated by ROC curves positioned close to the upper-left corner of the plot. Conversely, if the ROC curves lie below the diagonal line (X=Y), the algorithm is considered a poor classifier(Koval, 2018).

3.8.6 Precision

Precision is a key metric in evaluating the performance of a binary classification model, particularly when the cost of false positives is high. It measures the accuracy of the positive predictions made by the model, answering the question: "Of all the instances that were predicted as positive, how many were actually positive?(Sujata et al., 2015a)"

$$pricison = \frac{(TP)}{(TP + FP)}$$

Equation 3. 4 Precision Formula

3.9 Hardware and Software Tools Used

The proposed solution leveraged a mix of hardware and software tools for its implementation, testing, and execution:

- **Hardware Tools:**

✓ Internal hard disk: 1TB
✓ Physical memory (RAM): 8GB
✓ Processor: Intel® Core™ i7-7500U CPU @ 2.70GHz 2.90Hz
✓ Operating system: Windows 10 Pro
✓ System architecture: 64-bit

- **Software Tools:**

Visual Studio Code (VS Code): Served as the primary coding editor, offering a lightweight yet robust environment for coding, debugging, and software development tasks.

Python: Emerged as the preferred programming language for machine learning development, owing to its simplicity, extensive library support, and powerful tools for data manipulation, analysis,

GitHub: GitHub is a widely used web-based platform for version control and collaboration, particularly in software development. We utilized GitHub for code backup, change tracking, and collaborative work across different locations and devices.

Word Processing Software: Microsoft Word and PowerPoint were employed for composing and formatting our thesis document and presentation materials.

Data Visualization Tools like draw.io, Excel, and Python libraries such as Matplotlib were utilized to construct visual representations of our data, comprising charts, graphs, and plots. and **EdrawMax** for drawing architecture.

Mendeley Desktop: A straightforward reference management program, that also serving as a social network for academics to discover pertinent papers.

Google Scholar: Literature search tools like Google Scholar were utilized to discover pertinent academic publications and research articles.

Software's Impact on Development: The choice of software tools, particularly Python and VS Code, streamlined the development process, offering features like syntax highlighting, code completion, and debugging capabilities, thereby increasing productivity and efficiency.

Google Colab and Google Classroom:

Google Colab served as an essential platform for running Python code in an interactive environment, facilitating the seamless execution of machine learning models, especially for large-scale data processing tasks. Google Classroom was employed for managing and organizing coursework, ensuring timely submission of assignments and easy access to course materials.

CHAPTER FOUR

4. PROPOSED SOLUTION AND ARCHTECTURE

4.1 Introduction

Up to this point, we've covered the literature review, research methodology, identified gaps in existing literature (i.e., the problem statement), and formulated research questions. In this section, we'll delve into the detailed architecture of our proposed solution aimed at addressing the identified gaps and answering the research questions. We'll also outline the procedures and steps taken, including the selection of algorithms. This chapter aims to provide an in-depth description of the architecture employed for OTT Bypass detection.

4.2 Proposed Model Architecture

Machine learning architectures play a pivotal role in improving the reliability of OTT bypass fraud detection by leveraging data-driven techniques. These architectures have the capability to learn from vast amounts of data, extracting meaningful patterns and features that aid in the identification of fraudulent activities. By analyzing historical transactional data and user behavior, machine learning algorithms can detect anomalies indicative of potential fraud, thus improving the overall reliability of fraud detection systems.

Furthermore, machine learning algorithms demonstrate adaptability to evolving patterns of fraudulent activity. They can dynamically adjust their models based on new data and emerging trends, ensuring that detection systems remain effective in the face of constantly evolving fraud tactics. This adaptability is crucial in staying ahead of fraudsters who continually seek new ways to circumvent detection measures.

However, creating an effective machine learning model for OTT bypass fraud detection comes with its own set of challenges. These difficulties include the need to overcome various forms of fraudulent activity, ensuring the model's accuracy, and minimize false positives. Overcoming these challenges requires careful data preprocessing, feature engineering, and model tuning to optimize performance and reliability.

In selecting machine learning models for fraud detection, we must give careful consideration to their appropriateness for the task. Models such as Random Forest, Support Vector Machines (SVM), and Logistic Regression are commonly chosen for their proven effectiveness in identifying fraudulent behavior. Each of these models offers unique advantages, ranging from the ability to handle nonlinear relationships (Random Forest) to the ability to classify data into distinct categories (Logistic Regression). By leveraging a combination of these models, we can enhance the accuracy and reliability of our fraud detection systems, thereby mitigating the risks associated with OTT bypass fraud.

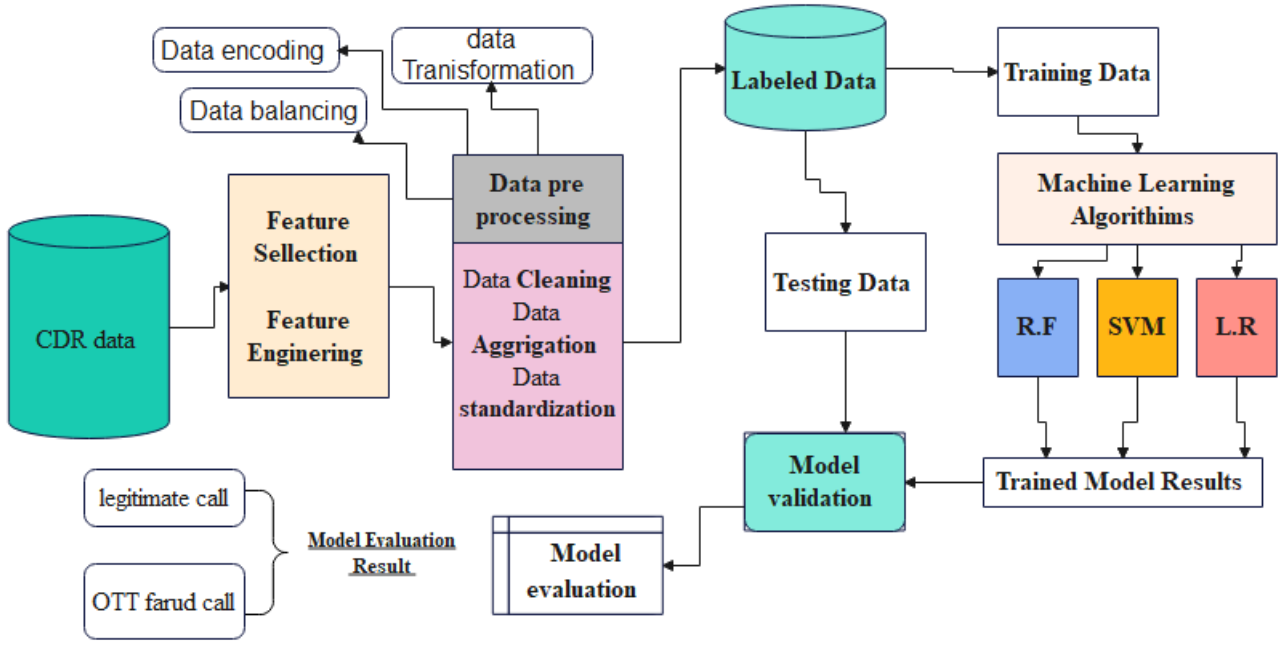
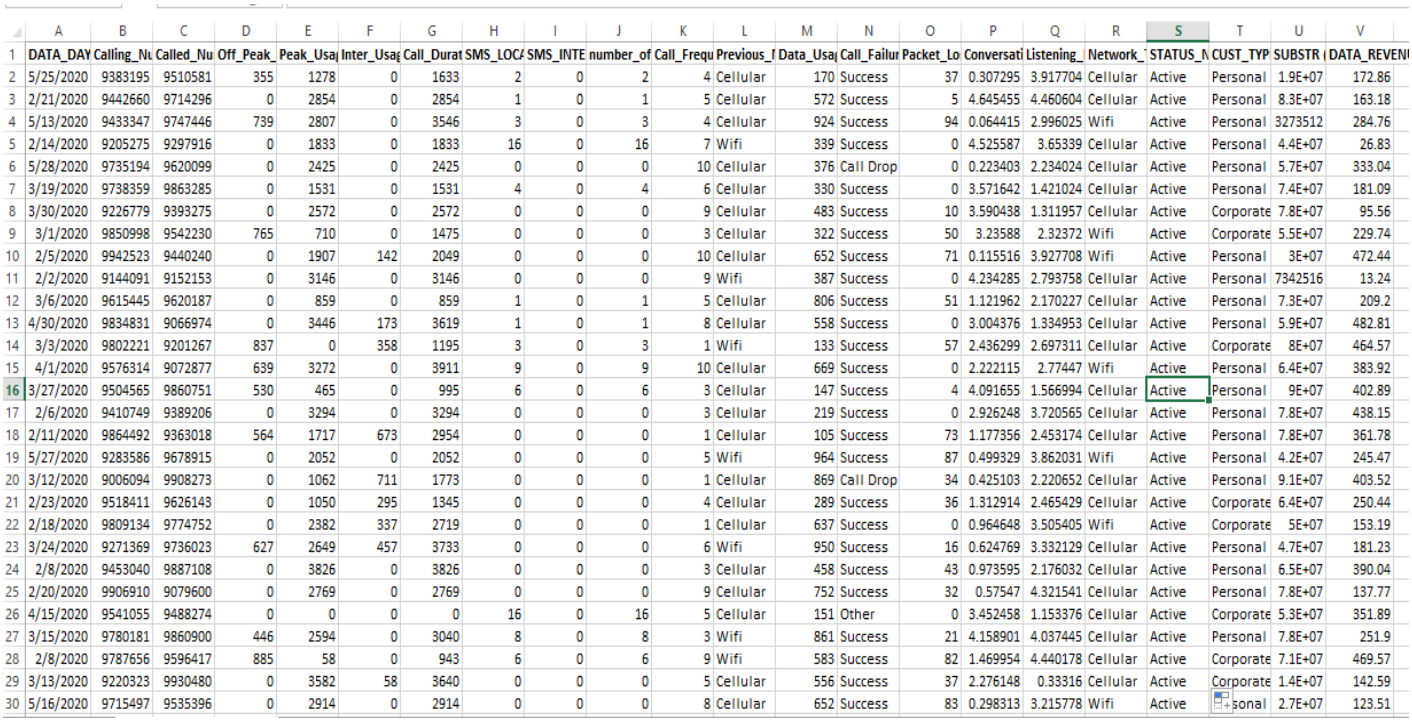


Figure 4. 1 proposed solution (architecture)

4.3 Proposed Data Preprocessing

Upon identifying key features, we precisely preprocessed and cleaned the data to prepare it for machine learning algorithms. Subsequently, we partitioned the data into training and testing sets, employing separate validation techniques for each. We utilized a 10-fold cross-validation method to evaluate the effectiveness of our classification algorithms. As previously mentioned, our chosen models for training included Random Forest, Logistic Regression, and Support Vector Machine

(SVM), renowned for their efficacy in detecting fraudulent activities. The primary objective of this study is to distinguish between fraudulent OTT bypass calls and genuine calls from service providers. To achieve this, we designated a target class labeled 'LABEL,' with '1' representing fraudulent call We've got two months of call detail records (CDR) from Ethio Telecom, but this data is massive and complicated to work with directly. To make it more manageable and suitable for training a model, we've applied feature selection and extraction techniques. Figure 4.2 shows what the raw CDR data looks like before this processing and '0' denoting legitimate ones.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	DATA_DAY	Calling_Nu	Called_Nu	Off_Peak_	Peak_Usa	Inter_Usa	Call_Durat	SMS_LOC/	SMS_INTE	number_of	Call_Frequ	Previous_I	Data_Usa	Call_Failur	Packet_Lo	Conversati	Listening_	Network_T	STATUS_N	CUST_TYP	SUBSTR (	DATA_REVEN
2	5/25/2020	9383195	9510581	355	1278	0	1633	2	0	2	4	Cellular	170	Success	37	0.307295	3.917704	Cellular	Active	Personal	1.9E+07	172.86
3	2/21/2020	9442660	9714296	0	2854	0	2854	1	0	1	5	Cellular	572	Success	5	4.645455	4.460604	Cellular	Active	Personal	8.3E+07	163.18
4	5/13/2020	9433347	9747446	739	2807	0	3546	3	0	3	4	Cellular	924	Success	94	0.064415	2.996025	Wifi	Active	Personal	3273512	284.76
5	2/14/2020	9205275	9297916	0	1833	0	1833	16	0	16	7	Wifi	339	Success	0	4.525587	3.65339	Cellular	Active	Personal	4.4E+07	26.83
6	5/28/2020	9735194	9620099	0	2425	0	2425	0	0	0	10	Cellular	376	Call Drop	0	0.223403	2.234024	Cellular	Active	Personal	5.7E+07	333.04
7	3/19/2020	9738359	9863285	0	1531	0	1531	4	0	4	6	Cellular	330	Success	0	3.571642	1.421024	Cellular	Active	Personal	7.4E+07	181.09
8	3/30/2020	9226779	9393275	0	2572	0	2572	0	0	0	9	Cellular	483	Success	10	3.590438	1.311957	Cellular	Active	Corporate	7.8E+07	95.56
9	3/1/2020	9850998	9542230	765	710	0	1475	0	0	0	3	Cellular	322	Success	50	3.23588	2.32372	Wifi	Active	Corporate	5.5E+07	229.74
10	2/5/2020	9942523	9440240	0	1907	142	2049	0	0	0	10	Cellular	652	Success	71	0.115516	3.927708	Wifi	Active	Personal	3E+07	472.44
11	2/2/2020	9144091	9152153	0	3146	0	3146	0	0	0	9	Wifi	387	Success	0	4.234285	2.793758	Cellular	Active	Personal	7342516	13.24
12	3/6/2020	9615445	9620187	0	859	0	859	1	0	1	5	Cellular	806	Success	51	1.121962	2.170227	Cellular	Active	Personal	7.3E+07	209.2
13	4/30/2020	9834831	9066974	0	3446	173	3619	1	0	1	8	Cellular	558	Success	0	3.004376	1.334953	Cellular	Active	Personal	5.9E+07	482.81
14	3/3/2020	9802221	9201267	837	0	358	1195	3	0	3	1	Wifi	133	Success	57	2.436299	2.697311	Cellular	Active	Corporate	8E+07	464.57
15	4/1/2020	9576314	9072877	639	3272	0	3911	9	0	9	10	Cellular	669	Success	0	2.222115	2.77447	Wifi	Active	Personal	6.4E+07	383.92
16	3/27/2020	9504565	9860751	530	465	0	995	6	0	6	3	Cellular	147	Success	4	4.091655	1.566994	Cellular	Active	Personal	9E+07	402.89
17	2/6/2020	9410749	9389206	0	3294	0	3294	0	0	0	3	Cellular	219	Success	0	2.926248	3.720565	Cellular	Active	Personal	7.8E+07	438.15
18	2/11/2020	9864492	9363018	564	1717	673	2954	0	0	0	1	Cellular	105	Success	73	1.177356	2.453174	Cellular	Active	Personal	7.8E+07	361.78
19	5/27/2020	9283586	9678915	0	2052	0	2052	0	0	0	5	Wifi	964	Success	87	0.499329	3.862031	Wifi	Active	Personal	4.2E+07	245.47
20	3/12/2020	9006094	9908273	0	1062	711	1773	0	0	0	1	Cellular	869	Call Drop	34	0.425103	2.220652	Cellular	Active	Personal	9.1E+07	403.52
21	2/23/2020	9518411	9626143	0	1050	295	1345	0	0	0	4	Cellular	289	Success	36	1.312914	2.465429	Cellular	Active	Corporate	6.4E+07	250.44
22	2/18/2020	9809134	9774752	0	2382	337	2719	0	0	0	1	Cellular	637	Success	0	0.964648	3.505405	Wifi	Active	Corporate	5E+07	153.19
23	3/24/2020	9271369	9736023	627	2649	457	3733	0	0	0	6	Wifi	950	Success	16	0.624769	3.332129	Cellular	Active	Personal	4.7E+07	181.23
24	2/8/2020	9453040	9887108	0	3826	0	3826	0	0	0	3	Cellular	458	Success	43	0.973595	2.176032	Cellular	Active	Personal	6.5E+07	390.04
25	2/20/2020	9906910	9079600	0	2769	0	2769	0	0	0	9	Cellular	752	Success	32	0.57547	4.321541	Cellular	Active	Personal	7.8E+07	137.77
26	4/15/2020	9541055	9488274	0	0	0	0	16	0	16	5	Cellular	151	Other	0	3.452458	1.153376	Cellular	Active	Corporate	5.3E+07	351.89
27	3/15/2020	9780181	9860900	446	2594	0	3040	8	0	8	3	Wifi	861	Success	21	4.158901	4.037445	Cellular	Active	Personal	7.8E+07	251.9
28	2/8/2020	9787656	9596417	885	58	0	943	6	0	6	9	Wifi	583	Success	82	1.469954	4.440178	Cellular	Active	Corporate	7.1E+07	469.57
29	3/13/2020	9220323	9930480	0	3582	58	3640	0	0	0	5	Cellular	556	Success	37	2.276148	0.33316	Cellular	Active	Corporate	1.4E+07	142.59
30	5/16/2020	9715497	9535396	0	2914	0	2914	0	0	0	8	Cellular	652	Success	83	0.298313	3.215778	Wifi	Active	sonal	2.7E+07	123.51

Figure 4. 2 Raw CDR file

The call detail records (CDR) we collected from Ethio Telecom to detect OTT Bypass Fraud is massive (see Figure 4.2). Analyzing all this data on a regular computer would be too slow. To address this, we can use a technique called feature ranking. Feature ranking helps us identify the most important pieces of information (features) within the CDR data for detecting this type of fraud. It's like finding the most relevant clues in a detective investigation.

Imagine we have a bunch of facts about phone calls: phone numbers involved, call duration, frequency, etc. Feature ranking assigns a score to each fact, indicating how helpful it is in spotting

fraud. We can then focus on the facts with the highest scores, streamlining our analysis. This approach makes data processing faster and helps us build a more efficient fraud detection model.

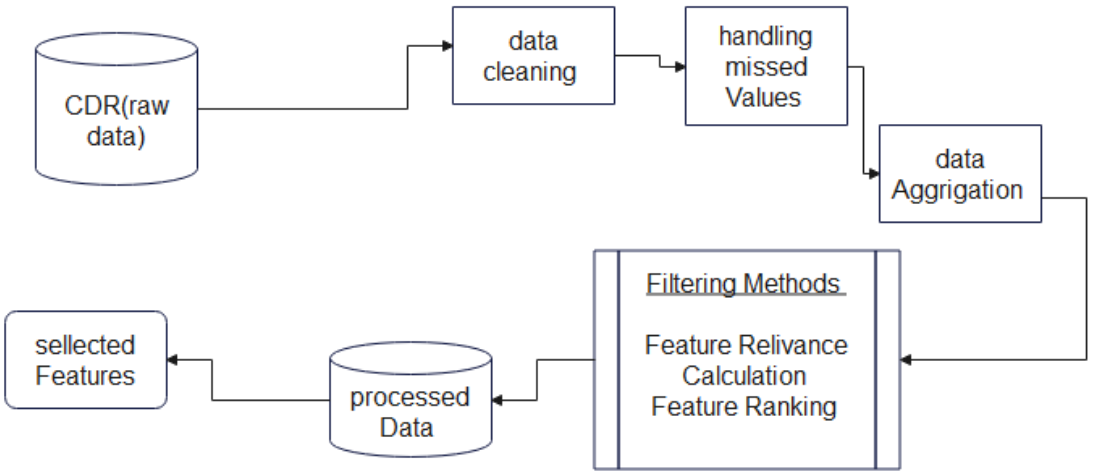


Figure 4. 3 Extracting the Most Relevant Call Details for Fraud Detection

Unveiling the key details is crucial for effectively detecting OTT Bypass Fraud. The raw call detail record (CDR) data we received from Ethio Telecom, while extensive with 17 columns of information, contains details not all equally important for our investigation. To streamline our analysis and focus on the most relevant aspects, we employed a feature selection technique called the filter method. This method acts like a filter, ranking each data feature based on its significance in identifying fraud. We then combined this ranking with our domain knowledge to select the most informative features. Our focus shifted towards customer usage data, the fingerprints that can potentially distinguish legitimate users from fraudsters. By analyzing this data, we can uncover patterns that deviate from typical user behavior. This approach helps us see through any attempts to mimic a legitimate customer, a common tactic in OTT Bypass Fraud. Through this process, we narrowed down the original 26 features to a core set of 8. Additionally, we incorporated 6 features from a public dataset, bringing the total to 15. This refined set encompasses crucial usage data collected over two months, including calling and called phone numbers, voice usage details (both international and local call minutes), call frequency, packet loss information, conversational MOS, listening MOS and data usage (megabytes) and more. By analyzing these key details, we can establish baselines for typical customer behavior and identify anomalies that might indicate

fraudulent activity. This approach allows us to build a more robust and efficient model for detecting OTT Bypass Fraud.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	DATA_DAY	Calling_N	Called_Nı	Off_Peak_	Peak_Usa	Inter_Usa	Call_Dura	Call_Freq	Previous_	Data_Usa	Call_Failı	Packet_Lc	Conversat	Listening	Network_
2	5/15/2020	9397243	9139727	141	1932	0	2073	10	Cellular	569	Success	0	2.412449	4.817088	Cellular
3	5/23/2020	9057686	9831744	236	412	799	1447	8	Cellular	815	Success	42	1.527827	3.655084	Cellular
4	5/29/2020	9504603	9760585	1149	2097	0	3246	7	Cellular	818	Equipmer	0	1.795879	1.183147	Cellular
5	5/9/2020	9958127	9439627	833	2071	0	2904	4	Cellular	508	Success	12	2.285157	4.010072	Cellular
6	5/18/2020	9217970	9757432	1061	2272	460	3793	9	Cellular	118	Success	97	3.996988	0.104873	Cellular
7	5/21/2020	9604415	9217368	1226	2493	0	3719	1	Cellular	313	Success	0	0.218349	4.197468	Cellular
8	5/19/2020	9439988	9133590	0	500	0	500	10	Cellular	162	Success	0	4.875632	0.733649	Cellular
9	5/30/2020	9196180	9362214	0	1423	0	1423	9	Cellular	243	Success	0	3.230427	0.402559	Cellular
10	5/12/2020	9180850	9461986	0	1241	0	1241	10	Cellular	668	Success	4	3.767983	3.678156	Cellular
11	5/4/2020	9990037	9576841	888	107	767	1762	7	Cellular	648	Success	57	0.86117	0.242234	Cellular
12	5/26/2020	9940635	9128684	0	827	0	827	3	Cellular	307	Success	25	2.916845	3.702095	Wifi
13	5/28/2020	9679712	9093842	0	1939	0	1939	8	Cellular	742	Success	8	3.678093	3.154905	Cellular
14	5/9/2020	9698918	9314672	0	3005	0	3005	8	Cellular	171	Success	76	2.029894	4.857429	Cellular
15	5/22/2020	9854717	9903962	0	1125	0	1125	6	Wifi	946	Success	12	2.088133	0.346022	Cellular
16	5/22/2020	9482347	9288169	0	440	0	440	9	Cellular	648	Success	0	2.013715	3.316902	Cellular
17	5/7/2020	9235203	9354206	0	1444	0	1444	5	Wifi	540	Success	9	4.814007	1.445648	Cellular
18	5/8/2020	9306267	9905673	0	119	0	119	9	Cellular	758	Success	66	1.85825	2.254911	Cellular
19	5/25/2020	9780203	9977923	0	1421	225	1646	9	Cellular	158	Equipmer	36	3.314722	0.155925	Cellular
20	5/9/2020	9106427	9458442	0	2470	0	2470	5	Cellular	397	Success	0	1.32336	4.100768	Cellular
21	5/18/2020	9935997	9245404	61	3882	0	3943	6	Cellular	447	Success	0	4.833838	1.185444	Cellular
22	5/3/2020	9991499	9769903	0	532	0	532	6	Cellular	289	Success	66	0.582671	2.658755	Cellular
23	5/15/2020	9111079	9098729	0	2069	523	2592	5	Cellular	720	Success	0	2.950831	0.348347	Cellular
24	5/24/2020	9061685	9724543	0	2223	0	2223	2	Cellular	212	Success	0	0.975015	4.273647	Cellular
25	5/18/2020	9491335	9189658	240	313	0	553	8	Cellular	344	Equipmer	77	0.796037	0.71871	Cellular
26	5/14/2020	9105491	9213837	0	1013	0	1013	6	Cellular	535	Success	71	2.652845	0.845205	Cellular
27	5/5/2020	9582253	9445747	0	2653	0	2653	1	Wifi	245	Success	43	2.871255	4.493535	Cellular
28	5/14/2020	9561009	9171510	0	2484	0	2484	6	Cellular	728	Success	38	1.131635	1.991848	Cellular
29	5/26/2020	9362964	9790108	0	2561	0	2561	1	Cellular	865	Success	0	2.83449	3.962883	Cellular
30	5/27/2020	9245393	9354803	0	3835	0	3835	9	Cellular	946	Success	0	4.840788	0.50781	Cellular

Figure 4. 4 Selected features from CDR

4.4 Proposed Model

Our machine learning toolbox equips us to tackle OTT bypass fraud by analyzing CDRs using advanced algorithms. **Random Forest** acts like a team of skilled investigators, constructing multiple decision trees to analyze data from various perspectives and uncover hidden patterns that may indicate fraud. Each tree votes on whether a call is fraudulent, and the majority decision determines the final outcome, making Random Forest highly accurate and robust in detecting subtle fraud indicators. **Support Vector Machines** SVMs, on the other hand, serve as sharp-eyed sentinels, meticulously examining features to draw a clear boundary between legitimate and fraudulent calls. By maximizing the margin between classes, SVMs excel at distinguishing normal from suspicious behavior. Lastly, **Logistic Regression** functions as our data analyst, building a mathematical model that estimates the probability of a call being fraudulent based on its features. While simpler than Random Forest and SVM, logistic regression provides a valuable baseline,

helping us understand the key factors contributing to fraud and allowing us to compare and refine the more complex models. This multi-model approach, or even an ensemble combining their strengths, enables us to adapt to evolving fraud tactics and effectively safeguard our network.

While the suggested model offers improvements in fraud detection, all machine learning models require careful configuration before training. Here's a breakdown of key parameters we'll define for our chosen models:

Optimizer: This algorithm guides the model's weight updates based on the calculated gradients. Choosing the right optimizer is essential for efficient learning and convergence. We'll explore different optimizers to find the one that best suits our model and data.

Loss Function: This function measures the discrepancy between the model's predicted and actual outputs, indicating how well the model is performing. We'll select a loss function appropriate for our task (e.g., classification or regression) to evaluate the model's performance during training.

Metrics: These are quantifiable measures that reflect a model's effectiveness. We'll use various metrics (accuracy, precision, recall, etc.) to assess our models' performance and potentially fine-tune hyperparameters for optimal results.

Why Not Other Machine Learning Models?

Neural Networks and Deep Learning: While deep learning models have demonstrated impressive performance in various applications, they require extensive computational resources and large volumes of data to train effectively. Given the scope of our study and the available data, the computational demands of deep learning models might outweigh their benefits. Additionally, deep learning models often act as "black boxes," making them less interpretable compared to LR, SVM, and RF.

K-Nearest Neighbors (KNN): KNN is a simple and intuitive algorithm but can be computationally expensive, especially with large datasets. It also struggles with high-dimensional data and might not be as effective for complex fraud detection as LR, SVM, and RF.

Naive Bayes: While Naive Bayes is efficient and easy to implement, it relies on strong independence assumptions between features, which may not hold true in real-world data. This can limit its effectiveness in detecting sophisticated fraud patterns.

Gradient Boosting Machines (GBM): GBM is a powerful ensemble method but can be computationally intensive and require careful tuning of hyperparameters. RF offers similar advantages with less complexity, making it a more practical choice for our objectives.

Why not Deep Learning?

While deep learning methods have shown great promise in various domains, they require extensive computational resources and large amounts of data to train effectively. In our case, we opted for LR, SVM, and RF because they are less resource-intensive and can provide good performance with the available CDR data from Ethio-Telecom. Additionally, these methods offer interpretability and ease of implementation, which are crucial for practical applications in fraud detection. By focusing on these models, we aim to balance performance, computational efficiency, and interpretability, making them well-suited for our specific fraud detection needs.

4.5 Implementation of the Proposed Solution

To combat OTT Bypass Fraud, we investigated into both the how and the where of our machine learning model implementation. We explored the specifics of the model's design and training process, along with the configuration of our Python and Jupyter Notebook environment. This environment provides the tools to interact with the model, analyze data, and visualize results – essential elements for building and wielding a powerful fraud detection system.

4.5.1 Environment Setup

We chose Python 3.9.0 as the programming language for this project. To maximize its capabilities, we used Jupyter Notebook, a widely-used IDE. This environment, running version 7.1.2 alongside Python, provides seamless integration for coding, data visualization, model training, collaboration with Google Colab, and progress tracking—essential components in developing our fraud detection system.

Under the hood, we employed a powerful set of Python libraries:

Pandas: This library empowers us to manipulate and analyze the call detail records (CDRs) that fuel our fraud detection model.

NumPy: As the foundation for numerical computing in Python, NumPy provides the essential tools for working with the data.

Matplotlib: This library serves as our visualization toolkit, enabling us to create static, animated, and interactive visualizations to understand the data and model behavior.

SciPy: Building upon NumPy, SciPy offers additional functionalities for scientific and technical computing, potentially used for advanced data analysis tasks.

Scikit-learn: This comprehensive machine learning library provides the core algorithms and tools we need for data mining, classification (including fraud detection), regression, and other essential tasks. It allows us to implement and analyze various machine learning algorithms with a consistent interface.

Finally, we might consider incorporating L2 Regularization (Ridge) as a technique to improve model performance. This approach modifies the loss function during training by penalizing models with large weights, potentially leading to better generalization and reduced overfitting.

This combined approach – Python, Jupyter Notebook, and powerful libraries – equips us with the necessary tools to build and refine our machine learning model for effective OTT Bypass Fraud detection.

4.6 Implementation of the Proposed Models

This section dives into the implementation of our chosen machine learning models – RF, SVM and LR for tackling OTT Bypass Fraud. Before we delve into the specifics of each model, we'll establish the foundation. This involves importing essential Python libraries for model training and loading the relevant CSV file containing our CDRs into the Jupyter Notebook environment. Figure 4.5 provides a visual representation of this initial setup.

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        from scipy import stats
        from sklearn.model_selection import train_test_split
        from sklearn.preprocessing import StandardScaler
        from sklearn.linear_model import LogisticRegression
        from sklearn.ensemble import RandomForestClassifier
        from sklearn.svm import SVC
        from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

        # Load the dataset
        data = pd.read_csv('cdr_data_136k.csv')

        # Display the first few rows of the dataset
        print("First few rows of the dataset:")
        print(data.head())
```

Figure 4. 5 Import Libraries and Loading CSV File

4.6.1 Implementation of RF

Our fraud detection system for OTT Bypass utilizes a Random Forest, a robust ensemble learning algorithm. Random Forest enhances detection accuracy by constructing multiple decision trees during training and aggregating their predictions. Our implementation begins with thorough data preprocessing, addressing missing values, encoding categorical variables, and normalizing features. The algorithm generates numerous decision trees, each trained on a random subset of the data with replacement (bootstrap sampling) and a random subset of features at each split, promoting diversity among the trees.

We assess the model's performance using metrics such as accuracy, precision, recall, F1 score, and cross-validation to ensure effective fraud detection. Feature importance scores help us understand each feature's contribution to detection, improving model interpretability. Hyperparameter tuning, including adjustments to the number of trees, maximum depth, and minimum samples per split, is conducted via grid search or random search to optimize performance. Regularization techniques and appropriate feature scaling are applied to improve generalization and reduce overfitting. By combining the predictions of multiple trees, our Random Forest model offers a reliable and accurate solution for detecting OTT Bypass fraud.

```
In [1]: import pandas as pd
        from sklearn.ensemble import RandomForestClassifier
        from sklearn.model_selection import train_test_split

In [3]: # Load the balanced dataset from the CSV file
        balanced_df = pd.read_csv('balanced_dataset.csv')

In [4]: # Split the dataset into features (X) and target variable (y)
        X = balanced_df.drop(columns=['Fraudulent'])
        y = balanced_df['Fraudulent']

In [5]: # Split the data into training and testing sets
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

In [6]: # Initialize and train your machine learning model
        rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)
        rf_classifier.fit(X_train, y_train)

Out[6]: RandomForestClassifier(random_state=42)
```

Figure 4. 6 Implementation of the RF Model

4.6.2 Implementation of the SVM Model

Utilizing a Support Vector Machine (SVM), our fraud detection system for Over-the-Top (OTT) Bypass relies on this powerful supervised learning algorithm to find the optimal hyperplane that effectively separates legitimate actions from fraudulent ones. Our implementation begins with thorough data preprocessing, addressing missing values, encoding categorical variables, and normalizing features. We carefully select the appropriate kernel (linear, polynomial, RBF, or sigmoid) to transform data into a higher-dimensional space when necessary. During training, the SVM identifies support vectors and optimizes the hinge loss function with a regularization parameter (C) to balance bias and variance. We evaluate performance using metrics such as accuracy, precision, recall, F1 score, and cross-validation to ensure effective fraud detection. Hyperparameter tuning through grid search or random search further refines our model. Additionally, regularization and feature scaling are applied to enhance generalization and minimize overfitting, ensuring our system reliably detects OTT Bypass fraud.

Once we've equipped ourselves with the necessary libraries and loaded the preprocessed data into Jupyter Notebook, we can define the parameters and train the Support Vector Machine (SVM) model.

```
In [2]: # Load your dataset
        data = pd.read_csv('CDR balanced_dataset.csv')

In [3]: # Assuming the last column is the target and the rest are features
        X = data.iloc[:, :-1].values
        y = data.iloc[:, -1].values

In [4]: # Split the data into training and testing sets
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

In [5]: # Standardize the feature variables
        scaler = StandardScaler()
        X_train = scaler.fit_transform(X_train)
        X_test = scaler.transform(X_test)

In [6]: # Initialize the SVM model with a linear kernel
        svm_model = SVC(kernel='linear', C=1, random_state=42)

In [*]: # Train the model
        svm_model.fit(X_train, y_train)
```

Figure 4. 7 Implementation of the SVM Model

4.6.3 Implementation of LR

Our approach to detecting OTT Bypass fraud utilizes Logistic Regression, a statistical method designed for binary classification tasks by modeling the relationship between a dependent variable and one or more independent variables. Logistic Regression is particularly effective in distinguishing between legitimate and fraudulent activities due to its probabilistic foundation.

The implementation starts with thorough data preprocessing, including handling missing values, encoding categorical variables, and normalizing features. Logistic Regression employs the logistic function to model the probability of a binary outcome, translating input features into a predicted probability of fraud. Training the model involves estimating the weights for each feature by maximizing the likelihood of the observed data, optimizing the logistic loss function to achieve the best-fitting model parameters. Regularization techniques, such as L1 (lasso) or L2 (ridge) regularization, are incorporated to prevent overfitting and improve generalization.

Model performance is evaluated using metrics such as accuracy, precision, recall, F1 score, and the area under the ROC curve (AUC-ROC). Cross-validation ensures robustness and assesses performance on unseen data. Hyperparameter tuning, through methods like grid search or random search, is employed to fine-tune parameters such as regularization strength.

Logistic Regression's interpretability is a key advantage, with model coefficients revealing the impact of each feature on fraud probability. This transparency aids in understanding the detection mechanism and making data-driven decisions. By leveraging Logistic Regression, our system offers a reliable, interpretable solution for detecting OTT Bypass fraud, striking a balance between simplicity and effective performance. The implementation of the RL model with its defined parameters is shown in Figure **4.8**.

```
In [2]: # Load your dataset
        data = pd.read_csv('minimized_balanced_dataset.csv')

In [3]: # Assuming the last column is the target and the rest are features
        X = data.iloc[:, :-1].values
        y = data.iloc[:, -1].values

In [5]: # Split the data into training and testing sets
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

In [6]: # Standardize the feature variables
        scaler = StandardScaler()
        X_train = scaler.fit_transform(X_train)
        X_test = scaler.transform(X_test)

In [7]: # Initialize the Logistic Regression model
        logistic_model = LogisticRegression(random_state=42, max_iter=1000)

In [8]: # Train the model
        logistic_model.fit(X_train, y_train)

Out[8]: LogisticRegression(max_iter=1000, random_state=42)
```

Figure 4. 8 Implementation of the LR Model

4.7 Implementation Evaluation Methods

In Chapter 3, we evaluated the model's performance using various metrics. These included accuracy, precision, recall, F1 score, confusion matrix, and ROC Curve. These measures were employed to assess the effectiveness of our machine learning model in detecting OTT bypass. Fraud. Precision gauges the model's accuracy in identifying fraudulent calls, while accuracy represents the overall correctness of predictions. The F1 score combines precision and recall to determine the model's ability to detect fraudulent behavior. The confusion matrix provides insight into the model's performance by indicating the number of true positives, true negatives, false positives, and false negatives. Additionally, the ROC curve illustrates the trade-off between sensitivity and specificity at different classification thresholds.

CHAPTER FIVE

5. RESULT AND DISCUSSION

This chapter provides a summary of the research experiment outcomes, which involved employing 10-fold cross-validation and separate test data validation techniques. In the detection of OTT Bypass fraud, three machine learning algorithms, namely RF, CVM, and LR, were utilized.

5.1 Model Performance Evaluation

The main objective of this study is to employ machine learning algorithms for the detection of OTT Bypass fraud and to compare their performance. As discussed in previous chapters, the use of machine learning techniques is essential in combating fraudulent activities, and this study aims to yield significant outcomes in this regard. The experiments involved the application of specified algorithms for the detection of OTT Bypass fraud using supervised machine learning techniques and 10-fold cross-validation, as well as a separate test approach with an 80% training and 20% testing data split. The final experimental results of the model are documented, evaluated, and compared.

We compared three machine learning models: Random Forest (RF), Support Vector Machine (SVM), and Logistic Regression (LR). We evaluated their performance using two techniques: training on a portion of the data (80%) and testing on the remaining portion (20%), and 10-fold cross-validation. Overall, RF achieved the best accuracy across both techniques. With the training and testing approach, RF reached 99.79% accuracy, while SVM and LR came in at 99.46% and 97.85% respectively.

We utilized a confusion matrix to assess the performance of our classification algorithms. The confusion matrix illustrates our model's effectiveness by evaluating the following: True Positives (TP), which are correctly classified positive instances; True Negatives (TN), which are correctly classified negative instances; False Positives (FP), which are negative instances incorrectly classified as positive; and False Negatives (FN), which are positive instances incorrectly classified as negative. The figures below display the confusion matrices of our model using both 10-fold cross-validation and train-test validation techniques.

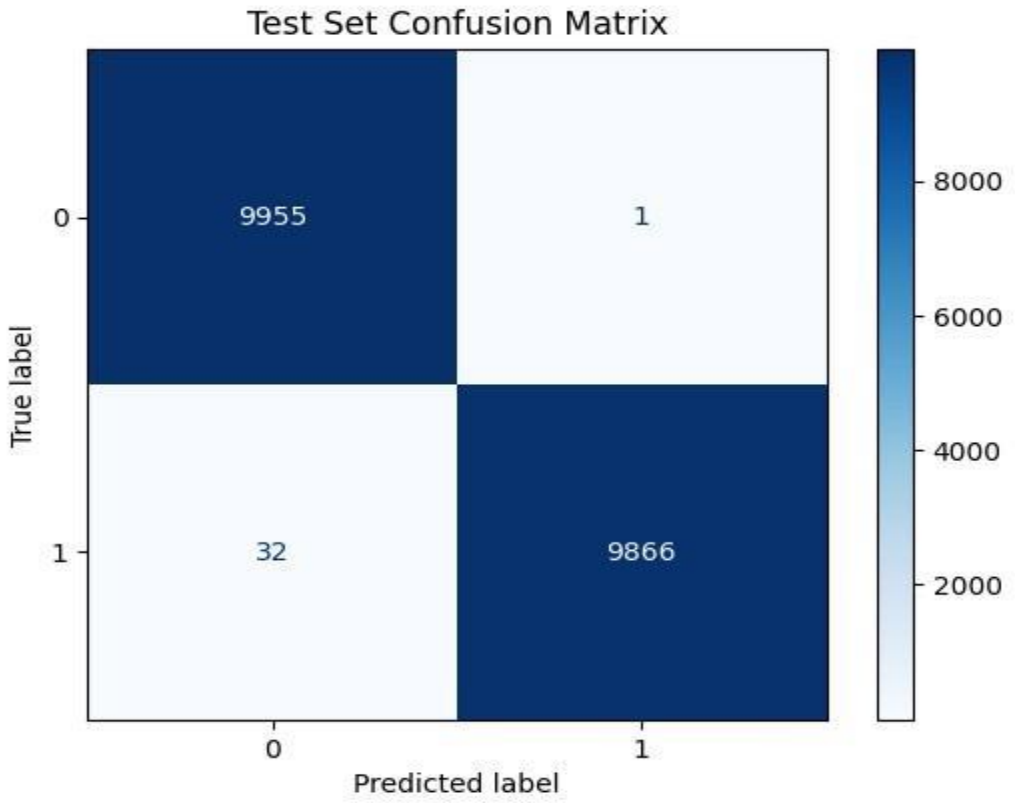


Figure 5. 1 Confusion matrix of RF model using 10 fold-cross validation

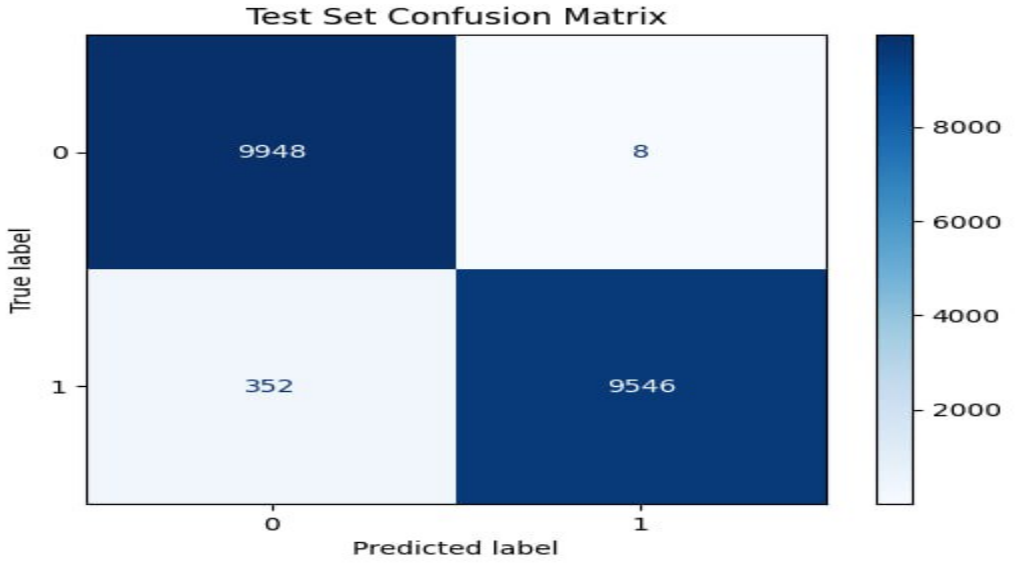


figure 5.2 confusion matrix of SVM model using 10 fold-cross Validation

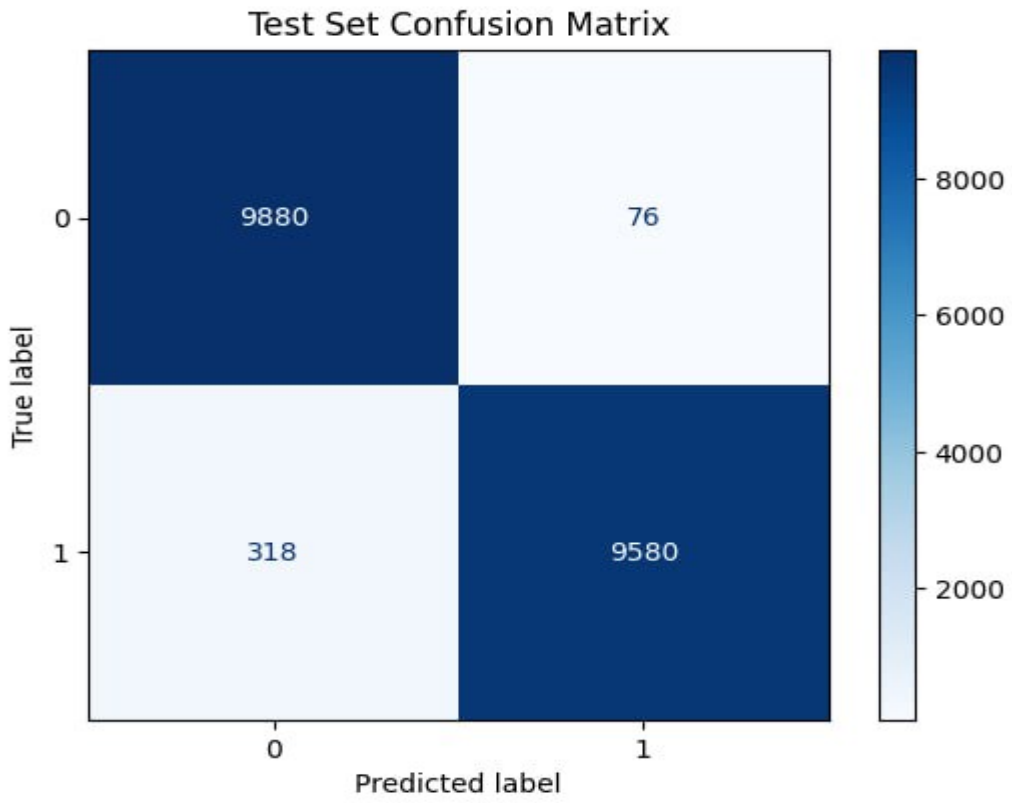


figure 5.3 confusion matrix of LR model using 10 fold-cross validation

In the context of 10-fold cross-validation, the model demonstrates higher accuracy and a low number of misclassified instances relative to correctly classified ones. This indicates that the model is highly effective in accurately identifying both positive and negative cases. The confusion matrices shown in Figures **5.1**, **5.2**, and **5.3** offer valuable insights into the performance of our classification model.

The matrix reveals a balanced performance in terms of precision, reflecting the model's capability to correctly identify positive cases, and recall, indicating its ability to recognize all positive occurrences accurately. This equilibrium suggests that the model strikes a solid balance between minimizing false positives and false negatives.

Summarizing the findings, the confusion matrix derived from the 10-fold cross-validation technique, as outlined in Table **5.2** below, provides a comprehensive overview of the model's performance, reaffirming its effectiveness in distinguishing between positive and negative instances with a high degree of accuracy and reliability.

Table 5. 1 Confusion Matrix of 10-fold Cross-Validation

Confusion matrix			Number of Instances	Correctly Classified	Incorrectly classified	Accuracy	Model
X	Y	Classified as					
9955	1	X=legitimate	19854	19821	33	99.79%	RF
32	9866	Y=fraudulent					
9948	8		19841	19481	400	99.46%	SVM
352	9546						
9880	76		19854	19460	394	97.85%	LR
318	9580						

The confusion matrix, as detailed in Table 5.1, highlights the performance of our classification models in detecting and classifying OTT/Bypass fraud. The models show strong performance, evidenced by the low counts of false positives (FP) and false negatives (FN). A low FP rate suggests that the models are rarely misclassifying negative cases as positive, while a low FN rate indicates that positive cases are accurately identified. When we assess the misclassified instances against the correctly classified ones (TP and TN), the differences are minor. Among the models evaluated using 10-fold cross-validation, the Random Forest (RF) model shows the highest performance, with SVM and LR models also performing well.

To further bolster our assessment of the model's performance in detecting OTT Bypass fraud, we applied the confusion matrix performance measure to the train-test split data, as depicted in Figures **5.4, 5.5,** and **5.6.** This approach allows us to scrutinize the model's behavior on unseen data, providing a more comprehensive evaluation of its generalization capability and robustness.

By examining the confusion matrices generated from the train-test split data, we can glean detailed insights into the model's classification accuracy and error tendencies. Similar to the observations made with the 10-fold cross-validation technique, we expect to see a low incidence of false positives and false negatives, indicative of the model's ability to effectively distinguish between fraudulent and legitimate transactions.

By integrating the insights obtained from both 10-fold cross-validation and train-test split evaluations, we can establish a more holistic understanding of the classification model's

performance in detecting OTT Bypass fraud. This comprehensive approach ensures that our assessment accounts for various facets of model behavior and provides a robust basis for evaluating its real-world efficacy and reliability.

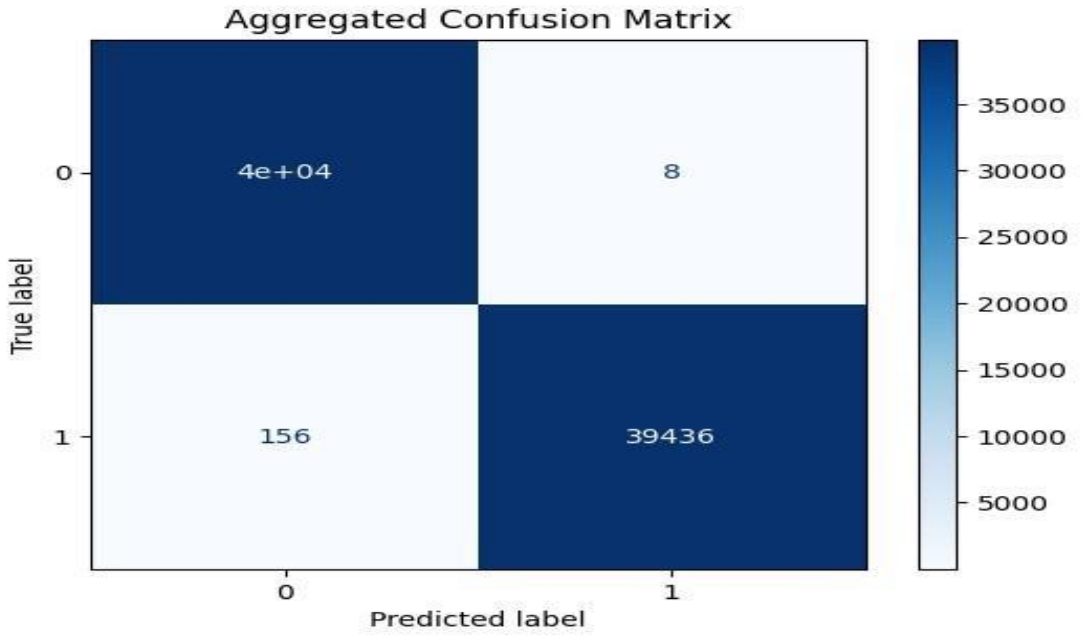


Figure 5. 4 confusion matrix of RF model using train-test split

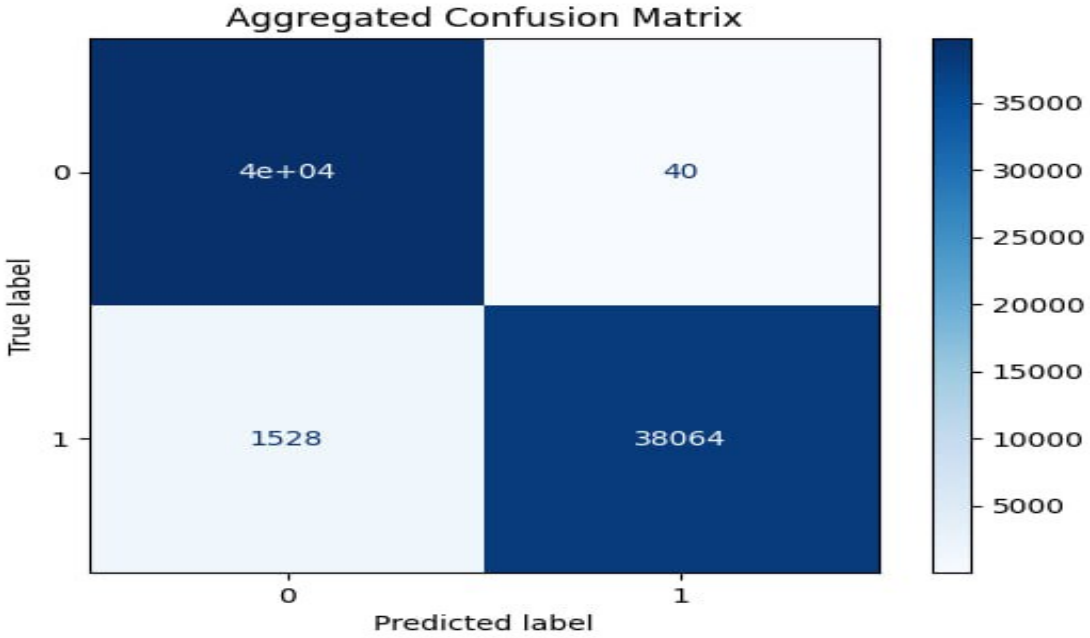


Figure 5. 5 confusion matrix of **SVM** model using train-test split

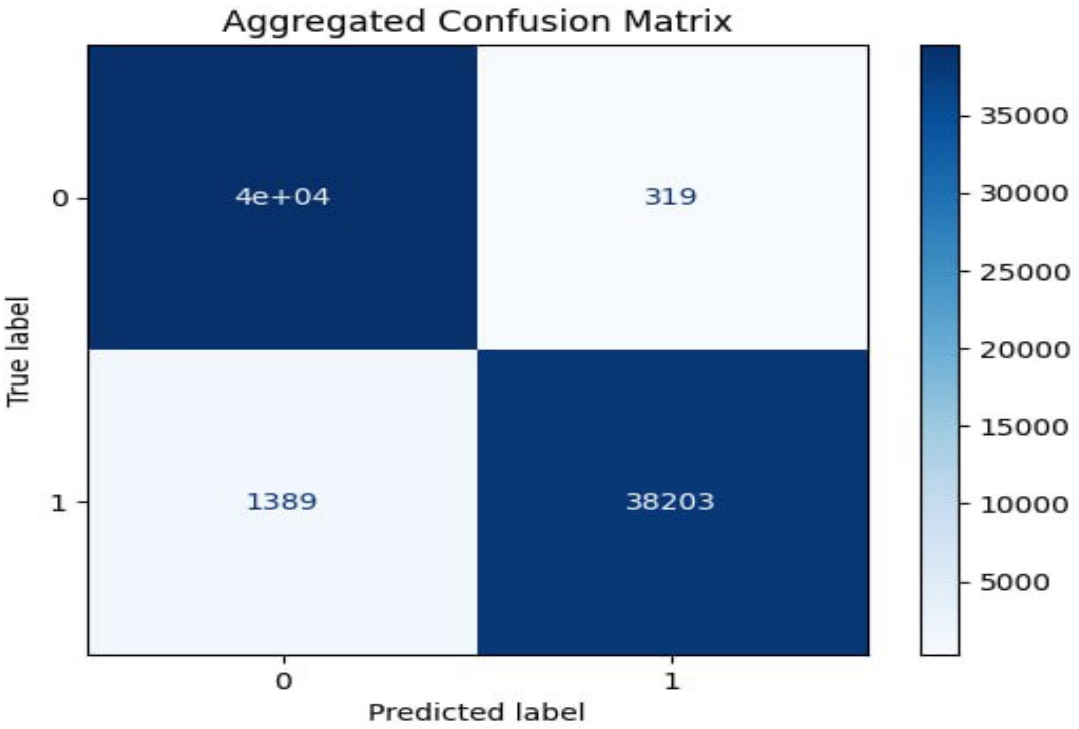


Figure 5. 6 Confusion matrix of **LR** model using train-test split

Figures **5.4, 5.5,** and **5.6** demonstrate how three different classification algorithms are used to detect and classify OTT Bypass fraud using train-test split data validation techniques. These techniques offer a unique perspective on the models' performance with unseen data, providing supplementary insights to those gained from other validation methods.

For a comprehensive summary of the algorithms' performance, Table **5.2** presents detailed results from the confusion matrices. These matrices categorize predictions into true positives, true negatives, false positives, and false negatives, allowing us to evaluate the accuracy, precision, recall, and overall effectiveness of each algorithm in detecting OTT Bypass fraud.

This thorough evaluation approach captures various aspects of each algorithm's performance across different validation techniques, facilitating well-informed decisions about their effectiveness for real-world fraud detection applications.

Table 5. 2 Confusion Matrix of Train Test Split Data

Confusion matrix			Number of Instances	Correctly Classified	Incorrectly classified	Accuracy	Model
X	Y	Classified as					
40,000	8	X=legitimate	79,600	79,436	164	99.83%	RF
156	39,436	Y=fraudulent					
40,000	40		79,632	78,264	1568	99.44%	SVM
1528	38,064						
40,000	319		79,911	78,203	1708	98.02%	LR
1,389	38,203						

The table provided offers a comprehensive breakdown of the confusion matrices, highlighting various performance measures such as true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Each of these metrics provides valuable insights into the model's ability to accurately classify instances of OTT Bypass fraud and legitimate calls from Ethio Telecom.

By analyzing the confusion matrices, we can observe how effectively the models classify calls into their respective categories while minimizing misclassifications. True positives represent legitimate calls correctly identified as such, while true negatives signify fraudulent calls correctly labeled as such. False positives occur when fraudulent calls are incorrectly classified as legitimate, and false negatives occur when legitimate calls are misclassified as fraudulent.

Through the employment of three classification techniques and the comparison of their performance, we can discern the strengths and weaknesses of each approach. It is evident from the analysis that the Random Forest (RF) algorithm outperforms both Logistic Regression (LR) and SVM techniques in terms of accuracy and efficacy in classifying calls into their proper categories.

This comprehensive evaluation underscores the importance of employing robust classification techniques, such as RF, in detecting and combating instances of OTT Bypass fraud effectively. By leveraging advanced algorithms and carefully analyzing performance metrics, organizations can enhance their fraud detection capabilities and mitigate potential risks associated with fraudulent activities.

When assessing the performance of our classification models using the confusion matrix, it's crucial to pinpoint which values need to be minimized based on the specific application. In our case, where the objective is to identify instances of fraud, our primary focus should be on reducing the number of negative instances misclassified as positive instances, which corresponds to false positives (FP). FP represents the count of fraudulent calls erroneously classified as legitimate.

Generally, the following Table **5.3** summarizes the performance of the classification algorithms by different performance measures metrics such as Accuracy, Precision, recall, and F1 score.

Table 5. 3 Performance Measures of Classification Algorithms

Algorithm	**Performance measure metrics**							
	10-fold cross-validation				**Train test split data (80 training, 20 test)**			
	accuracy	**Precision**	**recall**	**F1 score**	**Accuracy**	**precision**	**recall**	**F1 score**
RF	99.79%	99.98%	99.61%	99.79%	99.83%	99.99%	99.68%	99.83%
SVM	99.46%	99.99%	98.92%	99.45%	99.44%	99.95%	98.92%	99.43%
LR	97.85%	99.17%	96.49%	97.81%	98.02%	99.21%	96.79%	97.99%

The table suggests most models outperform others in all metrics (accuracy, precision, recall, F1 score). However, for OTT Bypass fraud detection, the focus seems to be on accuracy. Here's the breakdown based on accuracy:

- Random Forest (RF): Performed best using both validation methods (10-fold cross-validation and training/testing split).
- SVM: Ranked second in both validation techniques.
- LR: Came in third using both validation methods.

We used multiple metrics (accuracy, precision, recall, and F1 score) to assess how well our classification algorithm performs in detecting OTT Bypass fraud. These metrics are important

because they tell us how good the model is at identifying both real fraud cases (positive instances) and legitimate calls (negative instances).

The good news is that all the metrics performed similarly. This suggests the model is:

- Precise: It can accurately differentiate between fraud and legitimate calls.
- Good at recall: It catches most of the actual fraud cases.
- Overall accurate: It minimizes misclassifications (both false positives and negatives).

In simpler terms, the model seems to be very effective in distinguishing between fraudulent and legitimate calls.

Figures 5.1 illustrate the variations in accuracy, precision, recall, and F1 score for different classification algorithms, including RF, SVM, and LR, in detecting and classifying OTT/Bypass fraud during model training and testing with 10-fold cross-validation.

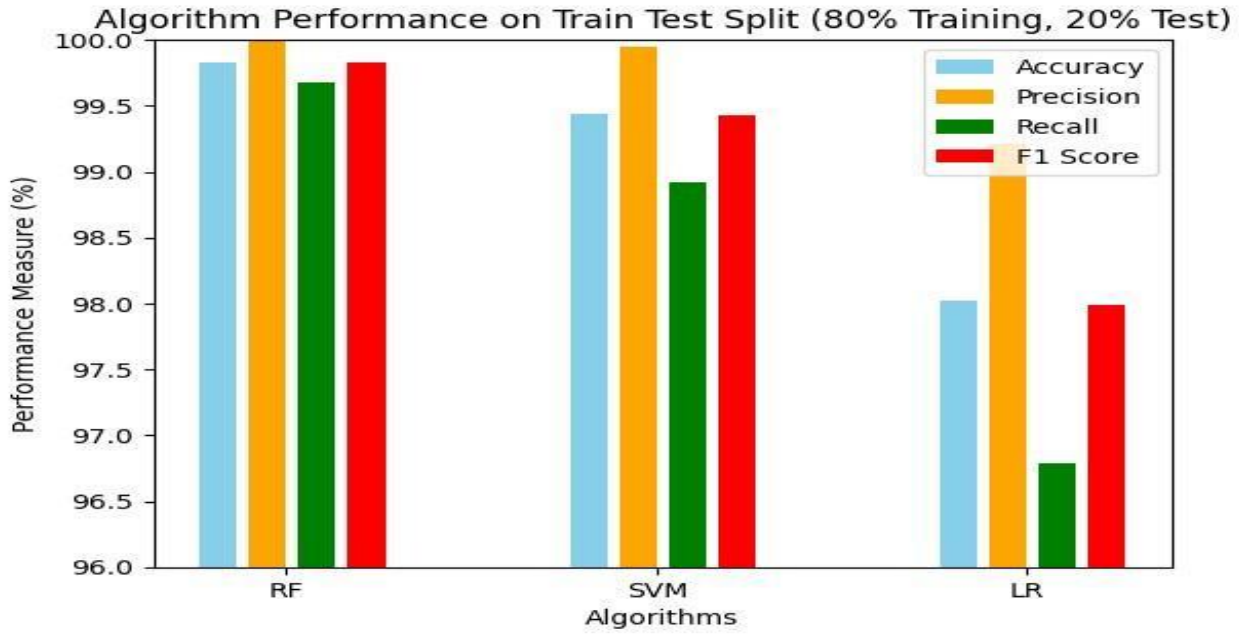


Figure 5. 7 Performance of Machine Learning Models on 10-Fold Cross-Validation

As shown in Figure 5.1, three classification algorithms were trained to detect and classify OTT/Bypass fraud using the 10-fold cross-validation technique. Random Forest (RF) achieved the highest accuracy and is ranked first, while Support Vector Machine (SVM) and Logistic Regression (LR) are ranked second and third, respectively, based on their performance. Additionally, we applied both 10-fold cross-validation and separate train-test split techniques to

train and evaluate our classification algorithms to determine which method provides better results. Figure 5.2 displays the performance of these algorithms during training and testing, measured by accuracy, precision, recall, and F1 score.

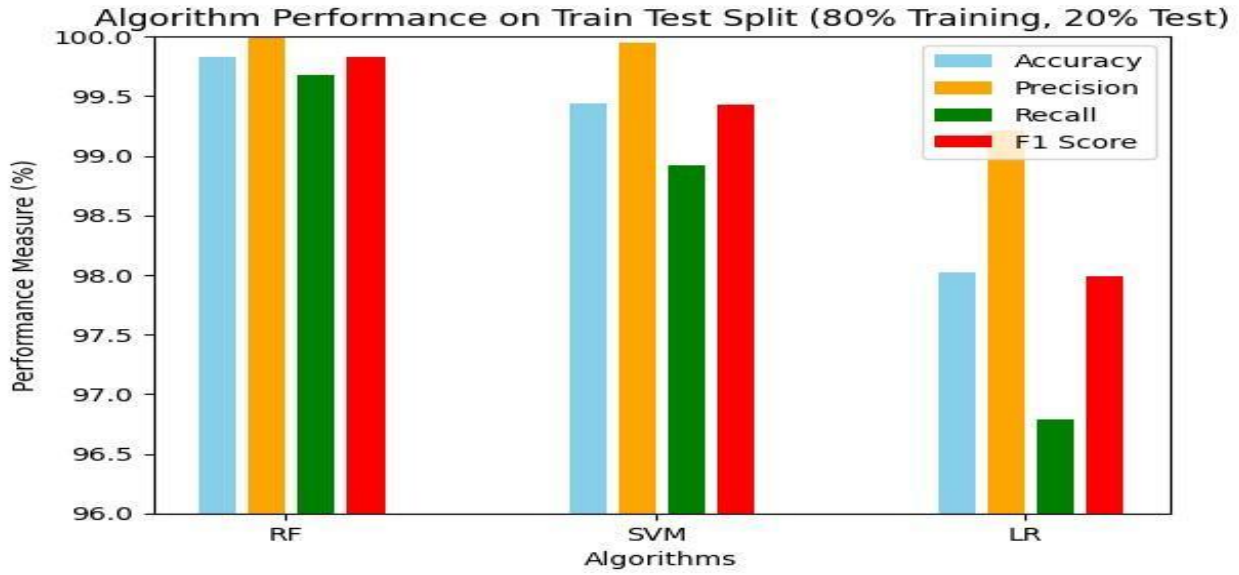


Figure 5. 8 Performance of Machine Learning Models on Train Test Split Data

A distinct train-test data validation method was employed to train three classification algorithms for detecting OTT Bypass fraud, as shown in Figures 5.1 and 5.2 above (with 80% of the data used for training and 20% for testing). Based on their performance, the RF classifier achieved the highest accuracy and is ranked first, while the LR and SVM algorithms are ranked second and third, respectively.

Both the 10-fold cross-validation technique and the train-test split data validation technique demonstrate that the number of misclassified instances is minimal compared to the correctly classified legitimate and fraudulent calls. This indicates that our models perform well in distinguishing between the two classes by minimizing misclassifications.

However, in scenarios where the number of FP instances is substantial, it signifies that a significant portion of fraudulent instances is being misidentified as legitimate calls. This can have far-reaching implications for telecom firms, including financial losses and potential damage to reputation.

Given the importance of reducing false positives in fraud detection applications, it's essential to consider the FP rate when evaluating a fraud detection system. A high FP rate means that a

considerable number of fraudulent cases are erroneously classified as non-fraudulent (legitimate) by the model, potentially leading to missed opportunities for further investigation and mitigation of fraudulent activities.

In summary, the low incidence of false positives observed in our classification models, as indicated by the confusion matrix performance metrics, underscores their effectiveness in recognizing both fraudulent and legitimate calls accurately. By minimizing misclassifications, our models can reliably identify fraudulent instances, thereby aiding in the prevention and mitigation of fraudulent activities within telecom networks.

In addition to conventional performance metrics like accuracy, precision, recall, F1 score, and confusion matrix, the Receiver Operating Characteristic (ROC) curve provides another valuable tool for evaluating model performance. This curve plots the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings, offering insights into the trade-offs between sensitivity and specificity(Li et al., 2018).

The ROC curve, depicted in the following figure using the test split validation technique, allows us to visualize how well the classification algorithms discriminate between positive and negative instances across different threshold levels. By examining the curve's shape and the area under it (AUC), we can measure the model's overall performance and its ability to distinguish between classes effectively.

Selecting the appropriate evaluation metric is crucial and often depends on the specific characteristics of the problem at hand. While accuracy is a commonly used metric, it may not always provide a comprehensive understanding of a model's performance, especially in scenarios where class distributions are imbalanced. In telecommunication fraud detection, ROC and AUC are particularly valuable metrics as they offer important insights into the model's ability to differentiate between fraudulent and legitimate transactions.

The advantages of using ROC analysis include its conceptual simplicity and its effectiveness in handling skewed data distributions. Additionally, the AUC provides a single, interpretable

measure of classification performance that accounts for various threshold settings, offering a more nuanced evaluation compared to accuracy alone.

In summary, leveraging metrics like ROC and AUC alongside traditional performance measures can provide a more comprehensive understanding of a model's effectiveness in telecommunication fraud detection. By considering multiple evaluation criteria, researchers and practitioners can make more informed decisions regarding model selection and deployment in real-world applications (Djomadji et al., 2023).

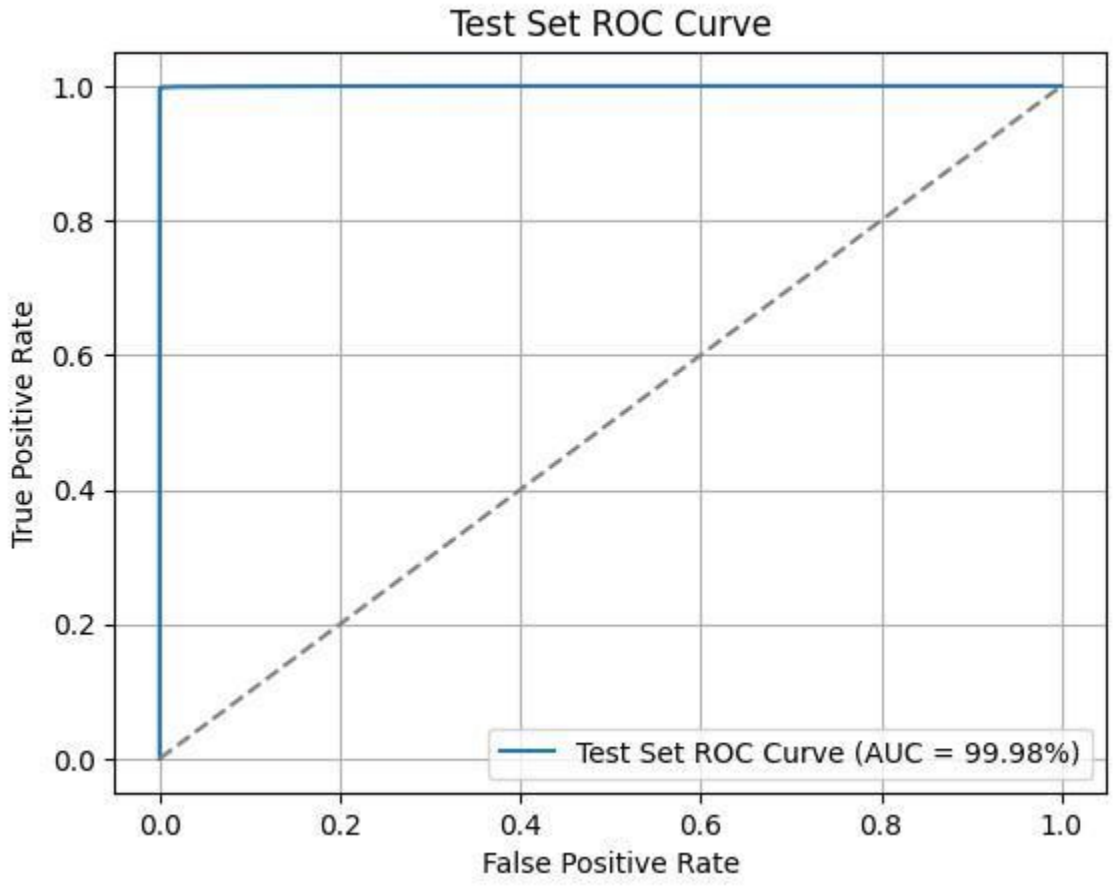


Figure 5. 9 ROC Curve of RF

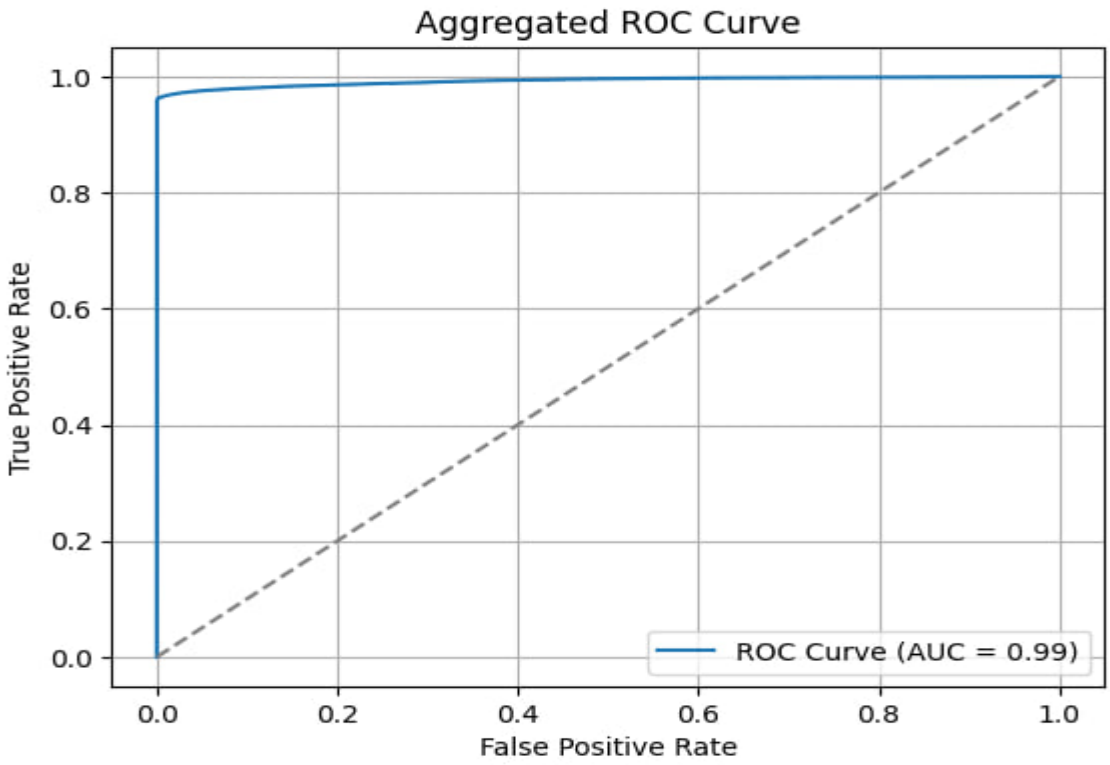


Figure 5. 10 ROC Curve of SVM

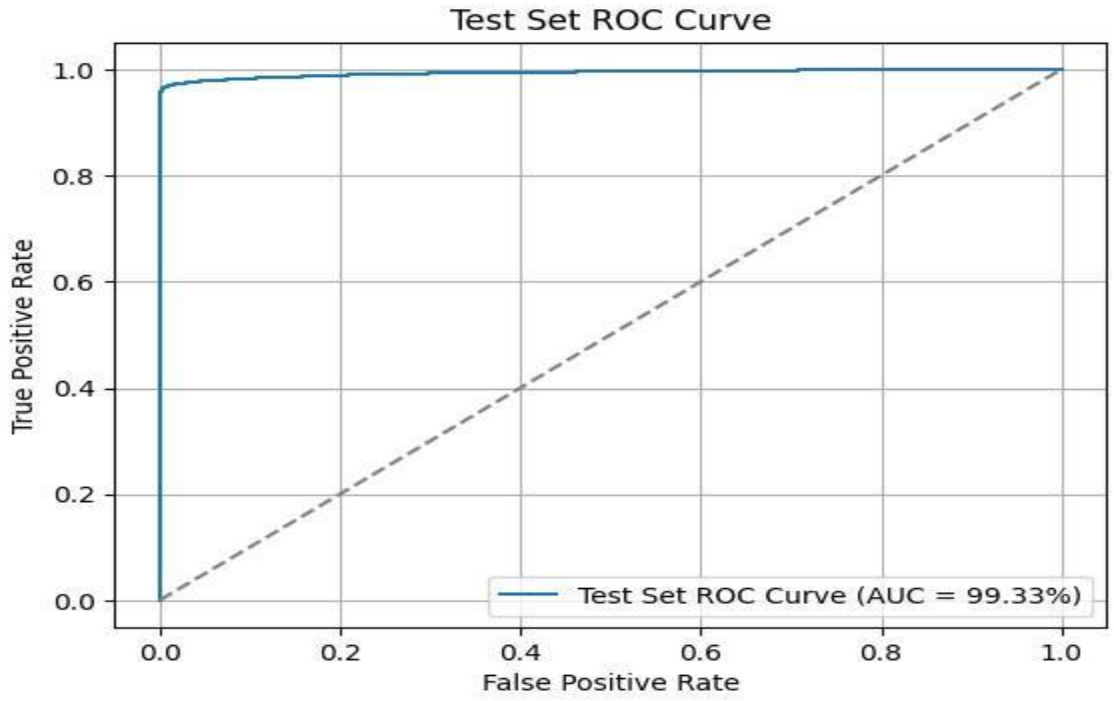


Figure 5. 11 ROC Curve of LR

The figures referenced, namely Figures **5.9, 5.10**, and **5.11**, depict the Receiver Operating Characteristic (ROC) curves generated for the classification algorithms utilized in detecting OTT Bypass fraud. In these curves, the X-axis represents the false positive rate (FPR), denoting the proportion of negative instances incorrectly classified as positive, while the Y-axis represents the true positive rate (TPR), indicating the proportion of positive instances correctly classified as positive. The ROC curve is calculated by varying the threshold settings for classification.

An ideal classifier would have an ROC curve passing through the top left corner of the plot, indicating a high true positive rate and a low false positive rate. The area under the ROC curve (AUC) serves as a summary metric of classifier performance, with a larger AUC suggesting superior performance. As observed in Figures **5.9, 5.10**, and **5.11**, the ROC curves for all classification algorithms achieve near-perfect discrimination, with AUC scores of 99.98%, 99.41%, and 99.33% for RF, SVM, and LR models, respectively.

In conclusion, the high AUC-ROC values and the ROC curves passing through the top-left corner signify excellent discrimination and overall performance of the classification models in correctly classifying positive instances while maintaining a low false positive rate. This capability is particularly valuable in real-world applications such as fraud detection, where accurately identifying positive instances while minimizing false positives is crucial.

Overall, the comprehensive evaluation of the models' performance, utilizing various performance metrics including accuracy, precision, recall, F1 score, confusion matrix, and ROC curve, confirms their effectiveness in detecting OTT Bypass fraud. Among the models, RF achieves the highest accuracy of 99.79%, followed by LR with 99.46%, and SVM with 97.85% accuracy, as determined through the train-test split data validation technique. Figure 5.12 illustrates the performance of our classification model, highlighting its accuracy in comparison to other metrics.

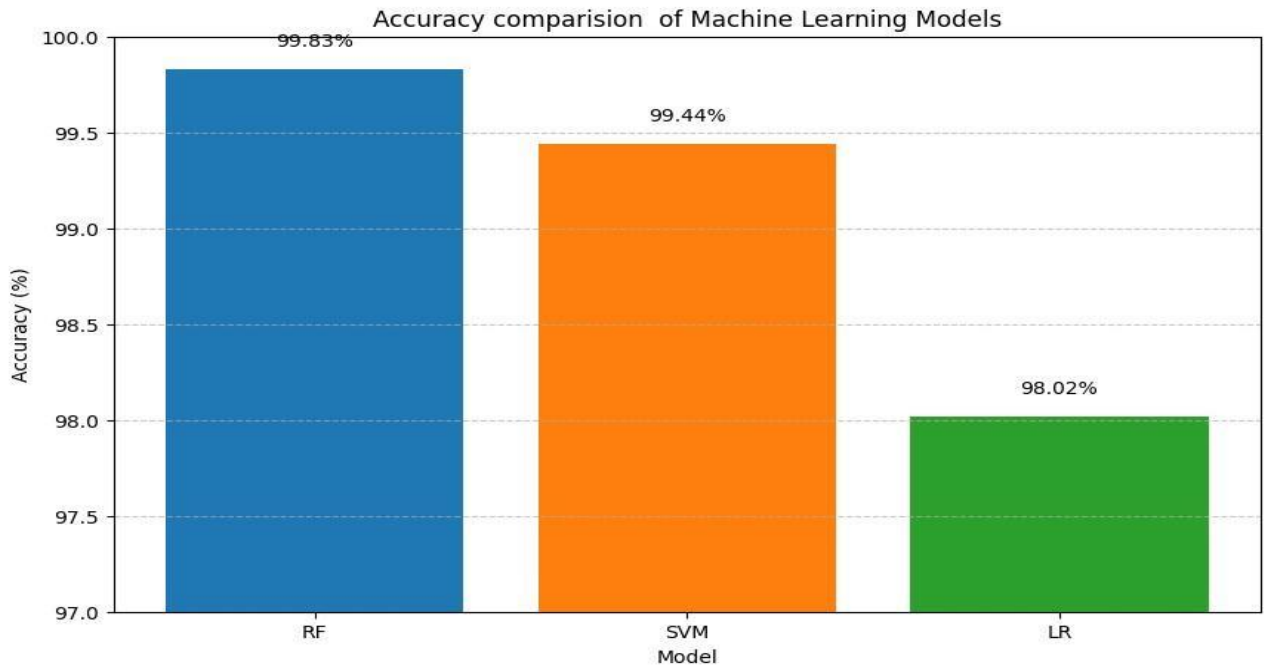


Figure 5. 12 Accuracy Comparison of Machine Learning Models

5.2 Integration of the Model into the Telecom Network

To effectively combat fraud in telecom networks, implementing a comprehensive monitoring system is crucial. We utilized machine learning models and analyzed Call Detail Records (CDRs) from Ethio-Telecom to devise an advanced fraud detection strategy. This approach involved scrutinizing various key indicators to identify abnormal calling patterns that may signal fraudulent activity.

Our analysis focused on several factors: Call Frequency to detect unusual increases in calls, particularly during off-peak hours or to specific numbers, which could suggest bypass fraud; Previous Network Type to monitor frequent switching to unfamiliar operators; and Network Type to observe call distribution across different network types. We also evaluated Data Usage for unexpected spikes that might indicate misuse for voice or video calls, and Call Failure Cause for high frequencies of failures that could point to unreliable services associated with fraud. Additional metrics included Packet Loss to identify network congestion issues, and Conversational MOS and Listening MOS to assess the perceived quality of calls, with lower scores potentially reflecting bypass services.

Our machine learning model demonstrated strong performance in distinguishing between legitimate and fraudulent calls, addressing the dynamic nature of fraud that traditional FMS and

TCG often struggle to detect. Unlike static rule-based systems, our model adapts to evolving fraud tactics through reactive analysis of CDRs and usage data. We recommend integrating this Intelligent Approach model into existing fraud detection systems to enhance their ability to manage sophisticated fraudulent activities and improve overall fraud prevention efforts.

5.3 Research Question Discussions

In over-all, our study addresses the research questions aimed at understanding and improving the detection of OTT bypass fraud using machine learning.

RQ1. What are the essential attributes to consider when implementing machine learning algorithms for detecting and classifying OTT bypass fraud?

Identifying key features for detecting OTT bypass fraud is crucial to designing effective machine learning models. Features like Call Frequency, Previous Network Type, and Call Failure Cause provide valuable insights into unusual behavior patterns indicative of fraud. For example, spikes in call frequency or frequent network switching may signal bypass attempts. Data Usage and Packet Loss help detect anomalies in service usage, particularly when OTT fraud exploits voice or video services. Careful selection of these features ensures the model focuses on the most relevant data points for fraud detection.

RQ2. How should the key attributes be processed to improve the performance of machine learning algorithms in identifying OTT bypass fraud?

Proper preprocessing is essential to optimize machine learning models for classifying OTT bypass fraud. Techniques like data cleaning (handling missing values, duplicates) and normalization ensure consistency across features such as call duration, data usage, and network types. Feature scaling helps balance attributes with different ranges, improving algorithm performance. Additionally, removing outliers and applying techniques like correlation analysis helps isolate attributes strongly associated with fraudulent activity, such as call failure rates and network switching patterns, ensuring the model can accurately distinguish between legitimate and fraudulent behaviors.

RQ3. How can we accurately evaluate the performance RF, SVM and LR models to ensure their effectiveness?

To accurately evaluate the performance of the selected machine learning models for detecting and classifying OTT bypass fraud, we first focus on choosing appropriate performance metrics and implementing rigorous validation techniques, including data splitting and cross-validation.

Performance Metrics: We need to select performance metrics that best reflect the model's effectiveness, especially given the imbalance often present in fraud detection datasets. For this purpose, metrics such as Precision, Recall, and the F1-Score are crucial. Precision measures how many of the predicted fraudulent cases are indeed frauds, which helps in reducing false positives. Recall, or the True Positive Rate, shows how well the model identifies actual fraudulent cases, which is essential for minimizing false negatives. The F1-Score, being the harmonic mean of Precision and Recall, provides a balanced measure, particularly valuable when dealing with class imbalance. Additionally, using a confusion matrix allows us to visualize the model's performance across different classes, and the ROC-AUC curve helps assess the model's ability to distinguish between fraud and non-fraud.

Data Splitting and Cross-Validation: To ensure that our models generalize well to unseen data, we first split the dataset into training and test sets, using an 80% training and 20% test split. This approach allows us to train the model on a substantial portion of the data while reserving a portion for final evaluation. To further validate the model's performance, we implement 10-fold cross-validation. This involves dividing the training data into 10 subsets (folds). We train the model on 9 of these folds and validate it on the remaining fold, repeating this process 10 times, each time with a different fold as the validation set. This method provides a robust estimate of the model's performance by ensuring that each data point is used for both training and validation. Additionally, in cases where class imbalance is significant, we use stratified 10-fold cross-validation to maintain the proportion of fraudulent and non-fraudulent cases in each fold, ensuring that each subset is representative of the overall dataset. By employing these techniques, we can obtain a comprehensive and reliable evaluation of our models' effectiveness in real-world scenarios.

RQ4. How do RF, SVM and LR algorithms compare in terms of their effectiveness in detecting and classifying OTT bypass fraud?

To compare the effectiveness of different machine learning algorithms in detecting and classifying OTT bypass fraud, we evaluate their performance using a range of models, including Random Forest (RF), Support Vector Machines (SVM), and Logistic Regression (LR). Each algorithm has its strengths and limitations, which influence its effectiveness in our context.

We find that **Random Forest** performs exceptionally well, achieving an accuracy score of 99.98%. Its ensemble approach, which aggregates multiple decision trees, allows it to handle complex interactions between features effectively. This high accuracy underscores its ability to capture subtle patterns associated with fraud and make reliable predictions.

Support Vector Machines also demonstrate strong performance with an accuracy of 99.41%. SVMs are adept at handling high-dimensional spaces and complex decision boundaries, making them suitable for distinguishing between legitimate and fraudulent activities in our dataset. However, they may require extensive computational resources and parameter tuning.

Logistic Regression, with an accuracy of 99.33%, provides a valuable baseline for comparison. Although it is simpler and less capable of capturing complex relationships compared to RF and SVM, it offers efficiency and interpretability. Logistic Regression is useful for initial detection efforts and helps set a benchmark for the performance of more complex models.

By comparing these algorithms, we gain insights into their respective strengths and trade-offs. This analysis allows us to identify the most effective machine learning model for detecting and classifying OTT bypass fraud, ensuring that we select the best tool for our specific application based on accuracy and performance metrics.

CHAPTER SIX

6. CONCLUSION AND FUTURE WORK

6.1 Conclusion

Telecom fraud, defined as the illegal or fraudulent use of telecommunications services or equipment for monetary gain or other harmful purposes, poses a significant threat to telecom companies. With the advancement of technology, fraudsters have found new ways to exploit vulnerabilities within telecom networks, causing substantial financial losses and security issues. One particularly damaging form of telecom fraud is OTT/Bypass fraud, which affects telecom operators by leading to consumer dissatisfaction and revenue loss. Ethio-Telecom, like many other providers, has not been immune to these challenges. Fraudsters often target regions with high international call termination fees but low local call costs, making it imperative for telecom companies to adopt more sophisticated detection methods. Despite the deployment of various detection technologies, telecom fraud remains a persistent issue.

The primary goal of this research was to detect and classify OTT/Bypass fraud using machine learning methods, focusing on user Call Detail Records (CDR) data. To achieve this, we employed three different machine-learning techniques: Random Forest (RF), Support Vector Machines (SVM), and Logistic Regression (LR). After collecting and preprocessing the CDR data from Ethio Telecom and public sources, we identified the most relevant features for detecting and classifying OTT/Bypass fraud. Given the evolving nature of fraudulent behavior, it was crucial to utilize machine learning methods capable of operating in such a complex environment. The RF, SVM, and LR models were chosen for their effectiveness in handling complex data, their proven success in cybersecurity applications, and their complementary strengths, all of which are critical for the task of detecting and classifying OTT/Bypass fraud in telecom networks. We trained our models using a 10-fold cross-validation technique as well as separate training and test datasets (80% for training and 20% for testing).

Our findings confirmed that all three models performed well in detecting and classifying OTT/Bypass fraud, as evidenced by several performance metrics. The RF model demonstrated the highest overall accuracy at 99.83%, followed by SVM at 99.44%, and LR at 98.02%. These results indicate that the RF model outperforms the other two models in classification accuracy, making it the most effective tool for this specific fraud detection task. The SVM and LR models, while

slightly less accurate, still offer robust performance and contribute valuable insights into the detection process. These findings highlight the potential of machine learning methods to enhance fraud detection in the telecom industry, offering a more adaptive and accurate approach to combating the ever-evolving tactics of fraudsters.

6.2 Major Contribution of the Research

This research significantly advances the detection and classification of OTT/Bypass fraud within telecom companies by developing advanced detection techniques and highly accurate classification algorithms. These innovations enhance both the accuracy and efficiency of fraud detection systems, ultimately reducing financial losses for telecom carriers. The study also highlights critical features necessary for effective fraud detection and advocates for the fusion and integration of diverse data sources. Furthermore, it offers decision support tools that facilitate the seamless integration of detection systems into telecom networks.

The impact of this research extends beyond immediate financial benefits; it also contributes to increasing revenue assurance, boosting operational efficiency, improving customer satisfaction, ensuring compliance with regulatory standards, and establishing industry leadership. Moreover, this work addresses the challenges of handling large-scale Call Detail Records (CDR) data in the context of OTT/Bypass fraud detection and classification. The research emphasizes the importance of preprocessing tasks such as data cleaning, normalization, and feature selection, which are essential when dealing with vast CDR Datasets. The development and implementation of efficient preprocessing techniques have significantly improved data quality, thereby improving the readiness of the data for model training.

This contribution is crucial as it lays the groundwork for accurate and reliable fraud detection and classification, enabling telecom companies to minimize revenue losses and strengthen network security. Overall, this study provides valuable insights and tools that empower telecom companies to more effectively detect and classify OTT/Bypass fraud, ensuring their continued success and security in a rapidly evolving technological landscape.

6.3 Future Work

Future work could focus on improving the generalizability of detection systems by testing models with diverse datasets from various telecom operators and regions. Additionally, incorporating real-time fraud mitigation strategies would enhance the models' ability to respond quickly to evolving fraud tactics and prevent potential losses. Researchers could also explore broader fraud detection by investigating other types of telecom fraud and incorporating more attributes, leading to more comprehensive detection methods that include proactive mitigation efforts to safeguard networks.

References

Ahmad, R., Alsmadi, I., Alhamdani, W., & Tawalbeh, L. (2022). A comprehensive deep learning benchmark for IoT IDS. In *Computers and Security* (Vol. 114). https://doi.org/10.1016/j.cose.2021.102588

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

CFCA 2017 Global Fraud Loss Survey – CFCA. (n.d.). Retrieved November 15, 2023, from https://cfca.org/document/cfca-2017-fraud-loss-survey-pdf/

Commission, F. T. (2013). National Do Not Call Registry. *Policing: An International Journal of Police Strategies & Management*, *36*(2), 158–163. https://doi.org/10.1108/pijpsm.2013.18136baa.002

Gudimalla, T. K. M., P, V., & Kannan, S. (2019). Survey Analysis of Cloned SIM Card. *SSRN Electronic Journal, January 2019*, 0–11. https://doi.org/10.2139/ssrn.3431642

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

Kinfmichael, W. (2021). *Semi-Supervised Algorithm to Detect Over-The-Top Bypass Fraud: in the case of ethio telecom*.

OTT app use undermining SMS revenue - Telecoms.com. (n.d.). Retrieved November 8, 2023, from https://telecoms.com/197721/ott-app-use-undermining-sms-revenue/

Pepper, R. (2013). *Cisco Visual Networking Index (VNI) Global Mobile Data Traffic Forecast Update Cisco Visual Networking Index (VNI) Expanding the Scope of Cisco ' s IP Thought Leadership*. *February*, 1–34.

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

Rehman, Z., Tariq, N., Moqurrab, S. A., & Yoo, J. (2024). *Machine learning and internet of things applications in enterprise architectures : Solutions , challenges , and open issues*. *September 2023*, 1–34. https://doi.org/10.1111/exsy.13467

Sahin, M., & Francillon, A. (2016). Over-The-Top bypass: Study of a recent telephony fraud.

Proceedings of the ACM Conference on Computer and Communications Security, *24-28-Octo*(June), 1106–1117. https://doi.org/10.1145/2976749.2978334

Sahin, M., Francillon, A., Gupta, P., & Ahamad, M. (2017). SoK: Fraud in Telephony Networks. *Proceedings - 2nd IEEE European Symposium on Security and Privacy, EuroS and P 2017*, 235–240. https://doi.org/10.1109/EuroSP.2017.40

Sawe, T. K. (2016). Emergence of OTT Communication Services and Sustenance of Revenue among Kenya Telcos . *International Journal of Innovative Science, Engineering & Technology*, *3*(8), 377–381. http://ijiset.com/vol3/v3s8/IJISET_V3_I8_52.pdf

Shalev-Shwartz, S., & Ben-David, S. (2013). Understanding machine learning: From theory to algorithms. In *Understanding Machine Learning: From Theory to Algorithms* (Vol. 9781107057). https://doi.org/10.1017/CBO9781107298019

Shanthamallu, U. S., Spanias, A., Tepedelenlioglu, C., & Stanley, M. (2017). A brief survey of machine learning methods for classification.pdf. *Ieee*. https://ieeexplore.ieee.org/document/8316459

Sujata, J., Sohag, S., Tanu, D., Chintan, D., Shubham, P., & Sumit, G. (2015). Impact of Over the Top (OTT) Services on Telecom Service Providers. *Indian Journal of Science and Technology*, *8*(S4), 145. https://doi.org/10.17485/ijst/2015/v8is4/62238

Tewodros, H. (2018). *Network Traffic Classification Using Machine Learning: A Step Towards Over-the-Top Bypass Fraud Detection*.

Wang, X. (2022). The Impact of IoT on News Media in the Smart Age. *Mobile Information Systems, 2022*. https://doi.org/10.1155/2022/2238233

Appendex
Feature Correlation Analysis Code

```
# Step 1: Upload the dataset to Google Colab

from google.colab import files
uploaded = files.upload()  # This will prompt you to upload your dataset

# Step 2: Import necessary libraries
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Step 3: Load the dataset
# Replace 'your_dataset.csv' with the name of the uploaded file
df = pd.read_csv('/content/reduced_dataset20per.csv')

# Step 4: Display the first few rows to inspect the data
print(df.head())

# Step 5: Compute Pearson Correlation
correlation_matrix = df.corr(method='pearson')

# Step 6: Display the correlation matrix
print(correlation_matrix)

# Step 7: Visualize the Pearson correlation matrix using a heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title("Pearson Correlation Matrix")
plt.show()

# Step 8: Identify highly correlated features
# Set a threshold to identify features strongly correlated with the target
threshold = 0.5  # Example: Features with correlation > 0.5 (or -0.5) are considered strong

target_variable = 'Fraudulent'  # Replace with your target column name
high_correlation_features = correlation_matrix[abs(correlation_matrix[target_variable]) >
threshold][target_variable]
print("Highly correlated features with the target variable:")
print(high_correlation_features)
```

Implementation code of Random Forest Model

```
Ll
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import StratifiedShuffleSplit, StratifiedKFold
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import (accuracy_score, precision_score, recall_score,
                             f1_score, confusion_matrix, ConfusionMatrixDisplay,
                             roc_curve, roc_auc_score)

# Load your dataset
data = pd.read_csv('/content/reduced_dataset20per (2).csv')

# Assuming the last column is the target and the rest are features
X = data.iloc[:, :-1].values
y = data.iloc[:, -1].values

# Stratified 80/20 train-test split
splitter = StratifiedShuffleSplit(n_splits=1, test_size=0.2, random_state=42)
for train_index, test_index in splitter.split(X, y):
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]

# Standardize the feature variables
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Initialize the Random Forest model
rf_model = RandomForestClassifier(random_state=42, n_estimators=100)

# 10-Fold Cross-Validation with Random Forest
kfold = StratifiedKFold(n_splits=10, shuffle=True, random_state=42)

# Initialize arrays to hold all the predictions and true labels
```

```
y_true_all = []
y_pred_all = []
y_proba_all = []

# Perform 10-Fold Cross Validation
for train_index, val_index in kfold.split(X_train, y_train):
    X_fold_train, X_fold_val = X_train[train_index], X_train[val_index]
    y_fold_train, y_fold_val = y_train[train_index], y_train[val_index]

    # Train the model
    rf_model.fit(X_fold_train, y_fold_train)

    # Predict on the validation fold
    y_fold_pred = rf_model.predict(X_fold_val)
    y_fold_pred_proba = rf_model.predict_proba(X_fold_val)[:, 1]

    # Collect the predictions and true labels
    y_true_all.extend(y_fold_val)
    y_pred_all.extend(y_fold_pred)
    y_proba_all.extend(y_fold_pred_proba)

# Convert to numpy arrays
y_true_all = np.array(y_true_all)
y_pred_all = np.array(y_pred_all)
y_proba_all = np.array(y_proba_all)

# Calculate the aggregated confusion matrix
cm_aggregated = confusion_matrix(y_true_all, y_pred_all)
disp_aggregated = ConfusionMatrixDisplay(confusion_matrix=cm_aggregated)
disp_aggregated.plot(cmap=plt.cm.Blues)
plt.title('Aggregated Confusion Matrix')
plt.show()

# Calculate the aggregated ROC curve
fpr, tpr, _ = roc_curve(y_true_all, y_proba_all)
roc_auc = roc_auc_score(y_true_all, y_proba_all)

plt.figure()
plt.plot(fpr, tpr, label=f'ROC Curve (AUC = {roc_auc*100:.2f}%)')
plt.plot([0, 1], [0, 1], linestyle='--', color='gray')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Aggregated ROC Curve')
plt.legend(loc='lower right')
plt.grid(True)
```

```
plt.show()

# Calculate and print aggregated metrics
mean_accuracy = accuracy_score(y_true_all, y_pred_all)
mean_precision = precision_score(y_true_all, y_pred_all)
mean_recall = recall_score(y_true_all, y_pred_all)
mean_f1 = f1_score(y_true_all, y_pred_all)

print("Aggregated 10-Fold Cross-Validation Metrics:")
print(f" Accuracy: {mean_accuracy*100:.2f}%")
print(f" Precision: {mean_precision*100:.2f}%")
print(f" Recall: {mean_recall*100:.2f}%")
print(f" F1 Score: {mean_f1*100:.2f}%")
print(f" ROC AUC: {roc_auc*100:.2f}%")

# Train on full training data and evaluate on test data
rf_model.fit(X_train, y_train)
y_test_pred = rf_model.predict(X_test)
y_test_pred_proba = rf_model.predict_proba(X_test)[:, 1]

# Calculate metrics on the test set
test_accuracy = accuracy_score(y_test, y_test_pred)
test_precision = precision_score(y_test, y_test_pred)
test_recall = recall_score(y_test, y_test_pred)
test_f1 = f1_score(y_test, y_test_pred)
test_roc_auc = roc_auc_score(y_test, y_test_pred_proba)

print("\nTest Set Metrics:")
print(f" Accuracy: {test_accuracy*100:.2f}%")
print(f" Precision: {test_precision*100:.2f}%")
print(f" Recall: {test_recall*100:.2f}%")
print(f" F1 Score: {test_f1*100:.2f}%")
print(f" ROC AUC: {test_roc_auc*100:.2f}%")

# Confusion Matrix for Test Set
cm_test = confusion_matrix(y_test, y_test_pred)
disp_test = ConfusionMatrixDisplay(confusion_matrix=cm_test)
disp_test.plot(cmap=plt.cm.Blues)
plt.title('Test Set Confusion Matrix')
plt.show()

# ROC Curve for Test Set
fpr_test, tpr_test, _ = roc_curve(y_test, y_test_pred_proba)

plt.figure()
```

```
plt.plot(fpr_test, tpr_test, label=f'Test Set ROC Curve (AUC = {test_roc_auc*100:.2f}%)')
plt.plot([0, 1], [0, 1], linestyle='--', color='gray')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Test Set ROC Curve')
plt.legend(loc='lower right')
plt.grid(True)
plt.show()
```

Implementation code of SVM

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import StratifiedShuffleSplit, StratifiedKFold
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import (accuracy_score, precision_score, recall_score,
                             f1_score, confusion_matrix, ConfusionMatrixDisplay,
                             roc_curve, roc_auc_score)

# Load your dataset
data = pd.read_csv('/content/reduced_dataset20per.csv')

# Assuming the last column is the target and the rest are features
X = data.iloc[:, :-1].values
y = data.iloc[:, -1].values

# Stratified 80/20 train-test split
splitter = StratifiedShuffleSplit(n_splits=1, test_size=0.2, random_state=42)
for train_index, test_index in splitter.split(X, y):
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]

# Standardize the feature variables
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Initialize the SVM model with probability enabled
svm_model = SVC(random_state=42, probability=True)

# 10-Fold Cross-Validation with SVM
```

```
kfold = StratifiedKFold(n_splits=10, shuffle=True, random_state=42)

# Initialize arrays to hold all the predictions and true labels
y_true_all = []
y_pred_all = []
y_proba_all = []

# Perform 10-Fold Cross Validation
for train_index, val_index in kfold.split(X_train, y_train):
    X_fold_train, X_fold_val = X_train[train_index], X_train[val_index]
    y_fold_train, y_fold_val = y_train[train_index], y_train[val_index]

    # Train the model
    svm_model.fit(X_fold_train, y_fold_train)

    # Predict on the validation fold
    y_fold_pred = svm_model.predict(X_fold_val)
    y_fold_pred_proba = svm_model.predict_proba(X_fold_val)[:, 1]

    # Collect the predictions and true labels
    y_true_all.extend(y_fold_val)
    y_pred_all.extend(y_fold_pred)
    y_proba_all.extend(y_fold_pred_proba)

# Convert to numpy arrays
y_true_all = np.array(y_true_all)
y_pred_all = np.array(y_pred_all)
y_proba_all = np.array(y_proba_all)

# Calculate the aggregated confusion matrix
cm_aggregated = confusion_matrix(y_true_all, y_pred_all)
disp_aggregated = ConfusionMatrixDisplay(confusion_matrix=cm_aggregated)
disp_aggregated.plot(cmap=plt.cm.Blues)
plt.title('Aggregated confusion matrix of SVM model using 10 fold-cross Validation')
plt.show()

# Calculate the aggregated ROC curve
fpr, tpr, _ = roc_curve(y_true_all, y_proba_all)
roc_auc = roc_auc_score(y_true_all, y_proba_all)

plt.figure()
plt.plot(fpr, tpr, label=f'ROC Curve (AUC = {roc_auc*100:.2f}%)')
plt.plot([0, 1], [0, 1], linestyle='--', color='gray')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
```

```
plt.title('Aggregated ROC Curve')
plt.legend(loc='lower right')
plt.grid(True)
plt.show()

# Calculate and print aggregated metrics
mean_accuracy = accuracy_score(y_true_all, y_pred_all)
mean_precision = precision_score(y_true_all, y_pred_all)
mean_recall = recall_score(y_true_all, y_pred_all)
mean_f1 = f1_score(y_true_all, y_pred_all)

print("Aggregated 10-Fold Cross-Validation Metrics:")
print(f" Accuracy: {mean_accuracy*100:.2f}%")
print(f" Precision: {mean_precision*100:.2f}%")
print(f" Recall: {mean_recall*100:.2f}%")
print(f" F1 Score: {mean_f1*100:.2f}%")
print(f" ROC AUC: {roc_auc*100:.2f}%")

# Train on full training data and evaluate on test data
svm_model.fit(X_train, y_train)
y_test_pred = svm_model.predict(X_test)
y_test_pred_proba = svm_model.predict_proba(X_test)[:, 1]

# Calculate metrics on the test set
test_accuracy = accuracy_score(y_test, y_test_pred)
test_precision = precision_score(y_test, y_test_pred)
test_recall = recall_score(y_test, y_test_pred)
test_f1 = f1_score(y_test, y_test_pred)
test_roc_auc = roc_auc_score(y_test, y_test_pred_proba)

print("\nTest Set Metrics:")
print(f" Accuracy: {test_accuracy*100:.2f}%")
print(f" Precision: {test_precision*100:.2f}%")
print(f" Recall: {test_recall*100:.2f}%")
print(f" F1 Score: {test_f1*100:.2f}%")
print(f" ROC AUC: {test_roc_auc*100:.2f}%")

# Confusion Matrix for Test Set
cm_test = confusion_matrix(y_test, y_test_pred)
disp_test = ConfusionMatrixDisplay(confusion_matrix=cm_test)
disp_test.plot(cmap=plt.cm.Blues)
plt.title('Test Set Confusion Matrix')
plt.show()

# ROC Curve for Test Set
```

```
fpr_test, tpr_test, _ = roc_curve(y_test, y_test_pred_proba)

plt.figure()
plt.plot(fpr_test, tpr_test, label=f'Test Set ROC Curve (AUC = {test_roc_auc*100:.2f}%)')
plt.plot([0, 1], [0, 1], linestyle='--', color='gray')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Test Set ROC Curve')
plt.legend(loc='lower right')
plt.grid(True)
plt.show()
```

Implementation of LR

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import StratifiedShuffleSplit, StratifiedKFold
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import (accuracy_score, precision_score, recall_score,
                             f1_score, confusion_matrix, ConfusionMatrixDisplay,
                             roc_curve, roc_auc_score)

# Load your dataset
data = pd.read_csv('/content/reduced_dataset20per.csv')

# Assuming the last column is the target and the rest are features
X = data.iloc[:, :-1].values
y = data.iloc[:, -1].values

# Stratified 80/20 train-test split
splitter = StratifiedShuffleSplit(n_splits=1, test_size=0.2, random_state=42)
for train_index, test_index in splitter.split(X, y):
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]

# Standardize the feature variables
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
```

```
X_test = scaler.transform(X_test)

# Initialize the Logistic Regression model
lr_model = LogisticRegression(random_state=42)

# 10-Fold Cross-Validation with Logistic Regression
kfold = StratifiedKFold(n_splits=10, shuffle=True, random_state=42)

# Initialize arrays to hold all the predictions and true labels
y_true_all = []
y_pred_all = []
y_proba_all = []

# Perform 10-Fold Cross Validation
for train_index, val_index in kfold.split(X_train, y_train):
    X_fold_train, X_fold_val = X_train[train_index], X_train[val_index]
    y_fold_train, y_fold_val = y_train[train_index], y_train[val_index]

    # Train the model
    lr_model.fit(X_fold_train, y_fold_train)

    # Predict on the validation fold
    y_fold_pred = lr_model.predict(X_fold_val)
    y_fold_pred_proba = lr_model.predict_proba(X_fold_val)[:, 1]

    # Collect the predictions and true labels
    y_true_all.extend(y_fold_val)
    y_pred_all.extend(y_fold_pred)
    y_proba_all.extend(y_fold_pred_proba)

# Convert to numpy arrays
y_true_all = np.array(y_true_all)
y_pred_all = np.array(y_pred_all)
y_proba_all = np.array(y_proba_all)

# Calculate the aggregated confusion matrix
cm_aggregated = confusion_matrix(y_true_all, y_pred_all)
disp_aggregated = ConfusionMatrixDisplay(confusion_matrix=cm_aggregated)
disp_aggregated.plot(cmap=plt.cm.Blues)
plt.title('Aggregated Confusion Matrix')
```

```
plt.show()

# Calculate the aggregated ROC curve
fpr, tpr, _ = roc_curve(y_true_all, y_proba_all)
roc_auc = roc_auc_score(y_true_all, y_proba_all)

plt.figure()
plt.plot(fpr, tpr, label=f'ROC Curve (AUC = {roc_auc*100:.2f}%)')
plt.plot([0, 1], [0, 1], linestyle='--', color='gray')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Aggregated ROC Curve')
plt.legend(loc='lower right')
plt.grid(True)
plt.show()

# Calculate and print aggregated metrics
mean_accuracy = accuracy_score(y_true_all, y_pred_all)
mean_precision = precision_score(y_true_all, y_pred_all)
mean_recall = recall_score(y_true_all, y_pred_all)
mean_f1 = f1_score(y_true_all, y_pred_all)

print("Aggregated 10-Fold Cross-Validation Metrics:")
print(f"  Accuracy: {mean_accuracy*100:.2f}%")
print(f"  Precision: {mean_precision*100:.2f}%")
print(f"  Recall: {mean_recall*100:.2f}%")
print(f"  F1 Score: {mean_f1*100:.2f}%")
print(f"  ROC AUC: {roc_auc*100:.2f}%")

# Train on full training data and evaluate on test data
lr_model.fit(X_train, y_train)
y_test_pred = lr_model.predict(X_test)
y_test_pred_proba = lr_model.predict_proba(X_test)[:, 1]

# Calculate metrics on the test set
test_accuracy = accuracy_score(y_test, y_test_pred)
test_precision = precision_score(y_test, y_test_pred)
test_recall = recall_score(y_test, y_test_pred)
test_f1 = f1_score(y_test, y_test_pred)
test_roc_auc = roc_auc_score(y_test, y_test_pred_proba)
```

```
print("\nTest Set Metrics:")
print(f"  Accuracy: {test_accuracy*100:.2f}%")
print(f"  Precision: {test_precision*100:.2f}%")
print(f"  Recall: {test_recall*100:.2f}%")
print(f"  F1 Score: {test_f1*100:.2f}%")
print(f"  ROC AUC: {test_roc_auc*100:.2f}%")

# Confusion Matrix  for Test Set
cm_test = confusion_matrix(y_test,  y_test_pred)
disp_test = ConfusionMatrixDisplay(confusion_matrix=cm_test)
disp_test.plot(cmap=plt.cm.Blues)
plt.title('Test  Set Confusion Matrix')
plt.show()

# ROC Curve for Test Set
fpr_test, tpr_test, _ = roc_curve(y_test, y_test_pred_proba)

plt.figure()
plt.plot(fpr_test, tpr_test, label=f'Test  Set ROC Curve (AUC =
{test_roc_auc*100:.2f}%)')
plt.plot([0,  1], [0,  1], linestyle='--',  color='gray')
plt.xlabel('False  Positive Rate')
plt.ylabel('True  Positive Rate')
plt.title('Test  Set ROC Curve')
plt.legend(loc='lower  right')
plt.grid(True)
plt.show()
```