Automated Feature Engineering Tool for Fraud Detection in Financial Transactions using Deep Learning

By

Stephen Onyango Buoro

051754

A Thesis Submitted in Total Fulfilment of the Requirements for the Degree of

Master of Science in Information Technology at Strathmore University

School of Computing & Engineering Sciences

Strathmore University

Nairobi, Kenya

June, 2025

Declaration and Approval

Declaration

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

Name of Candidate: **Stephen Onyango Buoro**

Sign: …… …… Date: …23rd May 2025……

Approval

The thesis of **Stephen Onyango Buoro** was reviewed and approved by the following:

Dr. Dickson Owuor,
Senior Lecturer, School of Computing & Engineering Sciences,
Strathmore University

Dr. Julius Butime,

Dean, School of Computing &Engineering Sciences,

Strathmore University

Prof. Bernard Shibwabo,

Director of Graduate Studies,

Strathmore University

Abstract

Financial fraud has become increasingly prevalent in modern corporate environments. It involves the deliberate use of deceptive tactics to achieve monetary benefits within various corporations and organizations. Conventional approaches such as manual verifications and inspections, although intended to detect fraudulent activities, frequently demonstrate shortcomings in terms of accuracy, cost-effectiveness, and efficiency. Existing automated fraud detection systems also have limitations. These includes inefficiencies, high costs of implementation, data imbalance, concept drift, false positives and negatives, limited generalizability, and difficulties with real-time processing. The quick and timely detection of fraudulent activities allows financial institutions to mitigate fraudulent conduct before it could lead to financial loss. This research developed an automated feature engineering tool for fraudulent detection. The developed solution involved utilizing deep learning (DL) techniques to analyse transactional data, thus revealing hidden trends that could indicate fraudulent activities. The developed MLP Classifier achieved an accuracy of 99.75%, surpassing the Logistic Regression and Decision Tree models. The model achieved perfect classification, with no errors in predicting fraud or non-fraudulent transactions. The significance of implementing effective fraud detection systems cannot be emphasized, as they serve as protectors of the security and integrity of the financial ecosystem. By providing protection to both financial institutions and cardholders against potential financial instability, these systems strengthen the fundamental basis of confidence on which transactions rely.

Key Words: Automated Feature Engineering, Deep Learning, Financial Fraud Detection, Machine Learning.

Table of Contents

List of Figures

List of Tables

List of Abbreviations and Acronyms

ANN	Artificial Neural Networks
API	Application Programming Interface
AUC	Area Under Curve
CART	Classification and Regression Trees
CCPA	California Consumer Privacy Act
FFD	Financial Fraud Detection
FN	False Negative
FP	False Positive
GDPR	General Data Protection Regulation
IG	Information Gain
LDA	Linear Discriminant Analysis
MLP	Multilayer Perceptron
PCA	Principal Component Analysis
ROC	Receiver Operating Characteristic
SPSS	Statistical Package for the Social Science
SVM	Support Vector Machine
TP	True positive
TN	True Negative

Definition of Terms

Artificial Neural Networks	An artificial neural network (ANN), or simply a neural network, is a computational model designed to process information and perform tasks like classification and regression. (Tian et al., 2021).
Automated Feature Engineering	Automated Feature Engineering is a technique that pulls out useful and meaningful features using a framework that can be applied to any problem (Xu et al., 2012).
Financial Fraud	Financial fraud is defined as acts that "intentionally and knowingly deceive the victim by misrepresenting, concealing, or omitting facts about promised goods, services, or other benefits and consequences that are non-existent, unnecessary, never intended to be provided, or deliberately distorted for the purpose of monetary gain." (Hilal et al., 2021)
Machine Learning	Machine learning encompasses a variety of algorithms that make intelligent predictions using a given dataset. (Nichols et al., 2018).
Deep Learning	Deep learning is a method in artificial intelligence (AI) that teaches computers to process data in a way that is inspired by the human brain (Janiesch et al., 2021).

Acknowledgments

First and foremost, I thank the Almighty God for granting me the strength and opportunity to pursue this master's Degree. I also wish to acknowledge and wholeheartedly thank my thesis supervisor, Dr. Dickson Owuor, for his unwavering guidance throughout this process. I could not have done it successfully without his help.

I also extend my warmest gratitude to Dr. Allan Omondi for his invaluable input and support throughout the whole journey. His insights and guidance during seminars were crucial to shaping my work

Finally, I would like to sincerely thank the entire Faculty for their guidance and the wealth of knowledge imparted throughout the course. Your collective wisdom has profoundly shaped my academic experience.

Dedication

This thesis is dedicated to the memory of my late father, Luke Buoro, whose unwavering dedication, sacrifices, and commitment laid the foundation for the person I have become.

I also dedicate this work to my wife and children, whose constant support and encouragement have been a source of strength and inspiration throughout this academic journey. To my mother, brothers, and sisters, I am profoundly grateful for your unconditional love, patience, and steadfast belief in my potential, which have sustained me through every challenge.

My sincere appreciation extends to my mentors and friends for their invaluable guidance and continued encouragement.

To all who have believed in me and contributed to my personal and academic growth, I offer my heartfelt thanks. This work stands as a testament to your enduring support and faith in my journey. Thank you.

Chapter 1: Introduction

1.1 Background of the Study

Financial fraud carries profound and widespread implications for both the financial sector and daily life. Its impact extends beyond immediate financial losses, eroding confidence in the industry, destabilizing economies, and influencing the cost of living for individuals (West & Bhattacharya, 2016). In recent years, the prevalence of fraud within Kenya's financial industry has reached alarming proportions. Fraud stands as the foremost challenge facing the business world, affecting virtually every facet of economic activity. There is now widespread concern that the rising tide of fraud in financial institutions over the past few years, if unchecked, could threaten the stability and survival of individual financial institutions as well as the performance of the banking industry. If fraud goes unchecked, it could cause the banking industry to collapse. The pervasive occurrence of frauds, thefts, and forgeries within the banking system poses a significant threat to the growth, development, and stability of financial institutions. This persistent challenge continues to exert a detrimental impact on the overall stability of the economic sector (Afjal et al., 2023).

Financial fraud encompasses a wide range of definitions; for the purposes of this research, it is defined as the intentional use of illegal methods or procedures to secure financial gain (West & Bhattacharya, 2016). According to the 2023 KPMG Barometer, Nigeria, Kenya, Zimbabwe, and South Africa account for 74% of all reported fraud cases in Africa, with Kenya leading the East African region, representing 7.75% of these incidents, followed by Uganda (2.98%) and Tanzania (2.88%). As is the case across much of Africa, government institutions and financial organizations are the primary targets of fraud in Kenya. The prevalence of fraud, misappropriation, extortion, and corruption is significant (KPMG, 2023). Over recent years, the frequency and volume of fraudulent activities have risen, making fraud one of the most persistent and formidable challenges faced by Kenyan institutions. In reality, con artists now view banks as their main means of survival. Only banks that are managed well—particularly in terms of preventing fraud—will be able to survive in the coming years.

Financial fraud extends beyond direct monetary losses, as it often funds illegal activities such as drug trafficking and organized crime. Merchants, in particular, bear the brunt of its financial impact, which encompasses not only the costs of shipping, disputes, and administrative overheads but also the erosion of consumer trust following fraudulent

transactions (Quah & Sriganesh, 2008; Sánchez et al., 2009). These consequences underscore the far-reaching implications of fraud and highlight the critical importance of implementing effective prevention measures.

Recent technological advancements, particularly the rise of the Internet and mobile computing, have significantly contributed to the escalation of financial fraud. Social factors, including the widespread use of credit cards, have not only fuelled increased consumer expenditure but also resulted in a corresponding surge in fraud cases. Given that fraudsters are constantly refining their methods, it is imperative that detection mechanisms evolve to keep pace with these developments (Vanini et al., 2023). In similar fields, such as credit card approval, bankruptcy prediction, and stock market analysis, machine learning approaches have proven to be highly effective, offering valuable insights and solutions. Fraud detection is viewed as a classification problem with a large disparity between fraudulent and legitimate transactions, as well as a substantial cost differential for misclassifying them.

This study developed an innovative approach to identify instances of financial fraud by harnessing the power of deep learning algorithms. The research employed automated feature engineering techniques applied to a comprehensive financial transaction's dataset. Automated feature engineering is a critical component of fraud detection systems. Automated feature engineering enabled the developed machine learning model to effectively identify fraudulent transactions by highlighting unusual patterns and anomalies. The continuous refinement of features and models ensures adaptability to evolving fraud tactics. This is expected to offer a robust tool for countering fraudulent activities within the banking industry in Kenya. This proactive approach aims to prevent potential financial losses that could otherwise disrupt the stability of the entire financial sector. Furthermore, the developed tool will play a pivotal role in curbing illicit activities, often funded by the proceeds of fraud, thereby contributing to the overall integrity of the financial system.

1.2 Problem Statement

Financial fraud represents a significant challenge within the financial sector, exhibiting a dynamic and evolving nature that defies clear trends (Hilal et al., 2021). Daily occurrences of fraudulent activities include identity theft, impersonation schemes (such as phishing attacks), debit and credit card fraud, foreclosure and loan scams, the use of counterfeit cheques, online fraud, as well as ransomware and malware-related crimes (Driel, 2018). These illicit practices can result in substantial financial losses, reputational harm, and a loss

of client trust. Fraudsters continually exploit advancements in technology to perpetrate their schemes.

In Kenya, bank fraud has risen sharply and is expected to continue growing as it becomes an entrenched part of everyday life. Data from the Banking Fraud Investigations Department (BFID) reveals that fraudsters have stolen at least Ksh1.5 billion ($17.64 million) from Kenyan banks over the past year, with many of these schemes orchestrated by technology-savvy bank employees (Nzomo, 2024). Despite efforts by investigators, only Ksh530 million has been recovered, with several cases still pending in court or under investigation. The BFID's data further indicates that a bank employee was involved in at least half of the reported crimes. The rise in fraud and cybercrime underscores the urgent need for financial institutions to invest in detection and preventive measures, as fraudsters become increasingly sophisticated with the growing reliance on technology (Nyakarimi, 2022).

This study focused on developing a fraud detection model specifically tailored to the challenges faced by Kenyan banks, utilising a Multi-Layer Perceptron (MLP) classifier. The study aimed to evaluate the effectiveness of deep learning in identifying and preventing fraud, focusing on data imbalances, classification accuracy, and the ability to generalise to new types of fraud. The research leveraged synthetic transaction data for model development and evaluation, recognizing the limitations of using synthetic data but aiming to demonstrate the model's potential for real-world applications.

1.3 Aim

The main aim of this study is to develop an automated feature engineering tool for fraud detection in financial transactions using deep learning.

1.4 Research Objectives

i). To investigate the causes of financial fraud in transactions in the banking industry in Kenya.

ii). To review the existing algorithms, models and frameworks used for fraud detection in financial transactions.

iii). To propose a deep learning-based feature engineering tool for fraud detection in financial transactions.

iv). To develop and test the proposed feature engineering tool.

1.5 Research Questions

i). What are the causes of financial fraud in transactions in the banking industry?

ii). What are the existing algorithms, models and frameworks used to detect fraud in financial transactions?

iii). How can an automated feature engineering tool for financial fraud detection be developed?

iv). How can the developed tool be tested?

1.6 Justification

Fraud and irregularities loom larger than ever in a time when banking and financial transactions are rapidly shifting into the digital sphere. As a proactive response, the industry has deployed fraud detection systems based on stringent rules meticulously crafted by experts to combat this escalating threat. Nonetheless, despite these efforts, the financial sector continues to face an alarming increase in fraudulent activity within its complex web of transactions, necessitating a concerted effort to mitigate their impact. This research focuses on identifying and mitigating fraudulent transactions within the banking industry, a sector that remains particularly susceptible to financial fraud. Utilizing the force of deep learning, the proposed method is able to identify anomalous behaviour patterns. By focusing on these anomalous deviations, the financial industry protections will be strengthened against fraudulent activities, heralding in a more secure and resilient era for financial transactions.

1.7 Scope and Limitations

This study explores the identification of financial fraud within Kenyan financial institutions, with a dual focus on traditional banking fraud and mobile banking fraud. The scope is expressly limited to banks, intentionally excluding other industries such as insurance and healthcare. Additionally, non-bank entities like savings and credit cooperatives (Saccos) are omitted. By concentrating on the banking sector, the study enables a more in-depth examination of the unique challenges and vulnerabilities associated with financial fraud, particularly in mobile banking. This focus facilitates the development of a specialized tool to detect and mitigate fraud in Kenyan banks, addressing both traditional and mobile banking fraud to enhance the integrity of financial transactions within these institutions.

Chapter 2: Literature Review

2.1 Introduction

The substantial losses incurred from financial fraud have consistently garnered the attention of scholars, businesses, and regulatory bodies. The onset of the coronavirus pandemic (COVID-19) delivered an unprecedented shock to the global financial system, catalysing the widespread adoption of digital financial services. This shift, while facilitating new avenues for financial transactions, also introduced novel challenges to the detection and prevention of fraud. This chapter offers a comprehensive review of the literature surrounding financial fraud detection, analysing the diverse techniques and models employed in this domain. By identifying key gaps, it seeks to inform the development of a conceptual framework. Additionally, the chapter examines the main factors contributing to financial fraud and provides an in-depth discussion of its various types.

2.2 Theoretical Literature

2.2.1 Financial Fraud

Financial fraud is the intentional act of deception involving financial transactions or data to gain an unfair or illegal advantage. This typically involves individuals or organizations falsifying information, misleading stakeholders, or committing acts that lead to financial losses. Financial fraud may involve manipulation of funds, forgery, embezzlement, and false reporting (Hilal et al., 2021).

Globally, financial fraud is a significant challenge affecting both developed and developing economies. According to the Association of Certified Fraud Examiners (ACFE), organizations lose approximately 5% of their annual revenues to fraud, equating to over $4 trillion globally (ACFE, 2021). Various forms of financial fraud have emerged in the digital age, including identity theft, credit card fraud, and cyber-fraud, exploiting the increasing reliance on digital financial systems. Countries like the United States and the UK have advanced systems in place to mitigate and monitor financial fraud, but it remains pervasive, especially in the wake of financial innovations.

In Africa, financial fraud is an ongoing issue that hinders economic development. The continent has seen significant strides in mobile banking and Fintech innovations, but these advancements have also created opportunities for fraud. According to the African Union Report (2024), Africa loses around $4 billion annually to cybercrime, with financial fraud being a significant contributor. Fraud schemes range from traditional embezzlement to

sophisticated digital scams. Regulatory bodies and financial institutions are working to strengthen anti-fraud mechanisms, but challenges like weak legal frameworks and low public awareness persist (Nsibirano et al., 2020).

financial fraud in Kenya is a major concern, especially with the increasing adoption of mobile money services like M-Pesa. According to the Central Bank of Kenya (2022), the country experiences significant financial fraud activities, with mobile money fraud being the most prevalent. In 2020, over 180 cases of mobile money-related fraud were reported, highlighting the vulnerabilities in the system. Financial institutions, in collaboration with the government, have introduced various anti-fraud initiatives, such as the Banking Fraud Investigation Unit (BFIU), but the rapid growth of financial technologies continues to expose new threats (CBK, 2022).

Financial fraud manifests in numerous forms, including identity theft, forgeries, mobile banking fraud, embezzlement, and phishing attacks. The consequences of these fraudulent activities extend far beyond financial losses, as they often lead to significant reputational damage and operational instability, potentially bringing an organization to a halt.

2.2.2 Types of Financial Fraud

2.2.2.1 Identity Theft

Identity theft is a type of fraud where an individual's personal information is stolen and used without their consent, often to commit financial crimes like applying for credit, loans, or making unauthorized purchases. Fraudsters typically obtain this information through various means, such as phishing, hacking, or social engineering (Hilal et al., 2021). Identity theft plays a significant role in financial fraud, with criminals exploiting stolen identities to open bank accounts, apply for credit cards, and execute fraudulent transactions. In the United States, the Federal Trade Commission (FTC) reported over 1.4 million cases of identity theft in 2020, marking a 100% increase compared to previous years (FTC, 2021). In Kenya, mobile money platforms have seen a surge in identity theft cases, as fraudsters use stolen credentials to access mobile wallets and siphon funds (CBK, 2021). The complexity of detecting identity theft lies in the seamless integration of legitimate financial transactions with fraudulent ones. Victims often realize their information has been compromised only after significant financial damage has been done. Strengthening cybersecurity measures and educating the public on secure data practices are critical steps in preventing identity theft.

However, the ever-evolving techniques used by cybercriminals continue to pose significant challenges (Olowookere & Adewale, 2020).

2.2.2.2 Credit Card Fraud

Credit card fraud happens when an individual unlawfully uses someone else's credit card information to obtain financial benefits. This can occur through physical theft or digital methods, such as hacking or skimming (RB & Kumar, 2021)
Globally, credit card fraud is one of the most common forms of financial fraud. In 2020, losses from credit card fraud worldwide reached $28.65 billion (Nilson Report, 2021). Fraudsters often engage in phishing or install skimming devices on ATMs or card readers to steal card details. In Kenya, credit card fraud is on the rise, especially with the increase in e-commerce platforms, where fraudulent card transactions are more prevalent (CBK, 2021). Financial institutions are implementing chip-based cards and multi-factor authentication to mitigate this risk.
Despite advancements in card security technologies, credit card fraud remains challenging to eliminate due to the increasing sophistication of cybercriminals. Additionally, the widespread use of online shopping and digital transactions has opened up new avenues for credit card fraudsters to exploit. While chip cards and tokenization help secure transactions, fraudulent techniques such as card-not-present fraud continue to be a significant challenge (Saad et al., 2024).

2.2.2.3 Embezzlement

Embezzlement is the act of dishonestly withholding or misappropriating funds entrusted to someone's care, typically in an employment or organizational setting. It involves manipulation of financial records to conceal the theft (Bhasin, 2016).
Embezzlement is a significant form of financial fraud, especially within corporate environments. A 2020 ACFE report revealed that embezzlement accounts for 30% of organizational fraud globally, leading to substantial financial losses. In Africa, weak internal controls and lack of oversight in public and private sectors create fertile ground for embezzlement. In Kenya, numerous high-profile cases involving government officials and large corporations have brought attention to the scale of embezzlement within the country (Transparency International,2020). Detecting embezzlement can be particularly difficult as the perpetrator often holds a trusted position within an organization. The complexity of financial transactions and the delay in discovery further complicate the process. Effective internal controls, regular audits, and whistleblower programs are necessary to prevent and

detect embezzlement, though these measures are not foolproof and can be circumvented by insiders (Bhasin, 2013).

2.2.2.4 Phishing Scams

Phishing scams represent a prevalent form of financial fraud in which attackers manipulate individuals into disclosing sensitive financial information, such as bank account details or login credentials. This form of fraud typically involves the use of deceptive emails, messages, or websites that impersonate trusted entities like banks, employers, or financial institutions. Fraudsters often send messages that appear legitimate, asking the recipient to provide personal information or click on links that direct them to fake websites, where their data is stolen. In financial phishing scams, attackers often aim to access bank accounts, initiate unauthorized transactions, or steal credit card information. Victims of phishing scams may face significant financial losses as a result. To prevent phishing attacks, individuals and organizations must remain cautious of unsolicited requests for financial details and employ cybersecurity measures such as two-factor authentication and email filtering ((Kulkarni, 2024)

2.2.2.5 Account Takeover

Account takeover is a type of financial fraud where a fraudster gains unauthorized access to an individual's online financial accounts, such as banking, credit card, or investment accounts. The attacker typically uses stolen login credentials, obtained through phishing, data breaches, or malware, to take control of the account. Once access is gained, the fraudster can make unauthorized transactions, withdraw funds, or change account information, making it difficult for the legitimate account holder to regain control. Account takeover is particularly damaging because it often goes unnoticed until significant financial losses occur. Victims may experience drained bank accounts, unauthorized purchases, or fraudulent loans in their name. To combat this type of fraud, financial institutions encourage the use of strong, unique passwords, two-factor authentication, and regular monitoring of account activity for suspicious behaviour (Aburbeian & Fernández-Veiga, 2024).

2.2.2.6 Debit Card Fraud

Debit card fraud transpires when unauthorized individuals illicitly obtain a person's debit card information and exploit it to make unauthorized transactions or withdraw funds from the victim's bank account. This type of fraud can happen through various methods, including card skimming, phishing attacks, or data breaches where personal banking information is

compromised (Adan, 2023). Fraudsters often use skimming devices to capture card details at ATMs or point-of-sale terminals, allowing them to clone cards and make transactions without the cardholder's knowledge. Additionally, social engineering tactics may involve tricking individuals into providing their card information directly. The implications of debit card fraud can be severe; victims may experience immediate financial loss, and recovery can be complex due to the direct link between debit cards and bank accounts (Alashwali et al., 2024).

In the United States, the Federal Reserve noted a substantial rise in debit card fraud, with losses reaching approximately $5.5 billion in a single year (Fed, 2021). Similarly, in Kenya, a rise in mobile banking has led to increased reports of debit card fraud, as criminals exploit vulnerabilities in mobile money platforms to drain victims' accounts (CBK, 2021). Detecting debit card fraud can be particularly challenging, as transactions may appear legitimate and often go unnoticed until account statements are reviewed. Consequently, proactive monitoring of bank statements and immediate reporting of suspicious activities to financial institutions are essential for mitigating the impact of such fraud. Implementing stronger security measures, such as chip technology and two-factor authentication, alongside consumer education on safeguarding card information, are crucial steps in combating debit card fraud. However, the rapid advancement of fraudulent techniques used by cybercriminals continues to present significant hurdles for both individuals and financial institutions (Olowookere & Adewale, 2020).

2.2.2.7 Mobile Banking Fraud

Mobile banking fraud encompasses unauthorized activities conducted via mobile platforms, leading to financial losses for individuals and institutions. Common methods include SIM swap fraud, where attackers deceive service providers to transfer a victim's phone number to a new SIM card, enabling interception of security codes and unauthorized account access. Phishing attacks involve deceptive messages that trick individuals into revealing sensitive information, such as login credentials, which fraudsters then exploit. Malware infections occur when malicious software is unknowingly installed on a user's device, allowing fraudsters to monitor keystrokes, capture login details, and execute unauthorized transactions. Social engineering tactics involve manipulating individuals into divulging confidential information by posing as legitimate entities, thereby gaining unauthorized access to mobile banking services (Phiri et al., 2024).

Globally, mobile banking fraud is a significant concern. In the United States, the Federal Bureau of Investigation has reported substantial financial losses due to SIM swap fraud, with individuals losing considerable amounts from their bank accounts. In the United Kingdom, online banking fraud occurs when criminals seize accounts and transfer funds from an individual's online bank account, posing challenges for machine learning due to extremely imbalanced data and the complexity of fraud (Vanini et al., 2023).

In Kenya, notable instances of mobile banking fraud have been documented. Fraudsters have exploited SIM swap techniques to gain control over victims' mobile numbers, facilitating unauthorized access to mobile banking accounts and resulting in significant financial losses (Ogara, 2023). Investigations have also revealed cases where bank employees colluded with external fraudsters, compromising customer accounts and siphoning funds through mobile banking platforms. Additionally, Kenyan mobile banking users have been targeted by phishing attacks, where deceptive messages prompt individuals to disclose personal information, subsequently used to access their banking accounts fraudulently.

2.2.3 Dimensionality Reduction Theory

Dimensionality reduction is an essential technique in machine learning and data analysis, which helps reduce the number of input variables (features) without losing critical information (Jolliffe, 2021). Dimensionality reduction plays a pivotal role in financial fraud detection, as financial datasets are often high-dimensional, and analysing them in their raw form can be computationally demanding and susceptible to overfitting. Two widely used techniques for dimensionality reduction are Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). PCA, introduced by Karl Pearson in 1901, is an unsupervised technique that focuses on identifying the principal components in the data—those directions where variance is maximized. The method transforms the original correlated variables into a set of uncorrelated variables (principal components), thereby capturing the maximum variance. Mathematically, PCA is derived by finding the eigenvectors and eigenvalues of the covariance matrix C of the dataset:

$$C = \frac{1}{N}\sum_{i=1}^{N}(X_i - \mu)\big((X_i - \mu)\big)^T$$

Equation 2.1 PCA Matrix

Where X_i represents the data points, and μ is the mean of the data. The eigenvectors corresponding to the largest eigenvalues represent the directions of maximum variance, allowing PCA to effectively reduce dimensionality by keeping only the principal components with the highest variance. PCA can help fraud detection models by identifying and keeping the most relevant features while ignoring noise and redundant information.

Linear Discriminant Analysis (LDA), first introduced by R.A. Fisher (1936), is a supervised technique designed to maximize the separability between classes. In the context of fraud detection, LDA helps distinguish between fraudulent and legitimate transactions by maximizing the ratio of between-class variance to within-class variance. LDA finds the projection vector www that solves:

$$arg\ max_w \frac{w^T S_B w}{w^T S_W w}$$

Equation 2.2 LDA Projection Vector

Where S_B is the between-class scatter matrix, and S_W is the within-class scatter matrix. LDA is particularly effective in cases where fraudulent activities have distinct patterns, improving the classifier's performance in distinguishing fraud from legitimate transactions. Both PCA and LDA are widely applied in fraud detection to reduce the complexity of datasets and enhance model performance. PCA is often used in exploratory data analysis to identify critical variables, while LDA is commonly applied in classification tasks to differentiate between fraudulent and non-fraudulent transactions (Jolliffe & Cadima, 2016). PCA assumes linear relationships between variables and focuses solely on variance, which may ignore important non-linear patterns relevant to fraud detection. LDA, while effective for classification, may struggle when the classes overlap significantly or are not linearly separable (Liu et al., 2019).

2.2.4 Information Theory

Information Theory, developed by Claude Shannon (1948), is a mathematical framework for quantifying information. It plays a pivotal role in many fields, including financial fraud detection, where the goal is to optimize decision-making by reducing uncertainty. Two key concepts from information theory—entropy and information gain—are commonly used in decision trees and other classification algorithms. Entropy measures the uncertainty or

disorder in a system. In financial fraud detection, entropy quantifies the unpredictability of a transaction being fraudulent. The entropy $H(X)$ of a random variable X is calculated as:

$$H(X) = \sum_{i=1}^{n} P(X_i) log_2 P(X_i)$$

Equation 2.3 Entropy

Where $P(X_i)$ is the probability of outcome X_i. In decision-making contexts, lower entropy indicates greater predictability. In fraud detection, a system with high entropy would imply significant uncertainty in distinguishing between fraudulent and legitimate transactions. Lowering the entropy means the model becomes better at predicting fraud.

Information Gain (IG) is used to determine the effectiveness of a particular feature in reducing uncertainty. Information gain is critical in decision tree algorithms, where it is used to select the feature that provides the most informative split. The formula for IG is:

$$IG(T, A) = H(T) - H(T|A)$$

Equation 2.4 Information Gain

Where $H(T)$ is the entropy of the system before the split, and $H(T|A)$ is the entropy after splitting on attribute A. In fraud detection, attributes such as transaction amount, geographical location, or transaction frequency can be assessed for their information gain to determine how well they distinguish between fraudulent and non-fraudulent activities (Zubair, 2023). In financial fraud detection, information gain helps identify the most significant attributes that differentiate fraudulent transactions from legitimate ones. Decision trees, which rely heavily on information gain, are commonly used to detect anomalies in large datasets by classifying transactions based on the most informative attributes (Rai et al., 2016).

While entropy and information gain are powerful tools in classification, they have some limitations. Entropy is sensitive to small changes in probability distribution, which can sometimes lead to overfitting, particularly in complex datasets. Additionally, decision trees based on information gain can become too complex if not pruned, leading to poor generalization on unseen data (Fan et al., 2011). Another limitation is that information gain favours attributes with more distinct values, which may not always correspond to meaningful splits for fraud detection.

2.3 Empirical Literature

2.3.1 Factors Influencing Financial Fraud

There exist many causes of fraud, which vary depending on the prevailing enabling environment. The focus of this study is directed towards the often-observed classes, namely social, technological, and legal, for the purpose of analysis and examination. Two other categories include personal and management.

2.3.1.1 Technological Causes of Fraud

The continuous progression of technology is a significant contributing aspect in the augmentation of fraudulent activities (Driel, 2018). As the level of ease increases, so does the susceptibility of fraudulent activities. The financial expenses associated with committing fraudulent activities utilizing readily accessible technology are relatively minimal. Technology enables the duplication of documents with a high degree of accuracy (Wu et al., 2022). The phenomenon has resulted in the worldwide integration of societies. The elimination of physical barriers has facilitated the perpetration of fraud across vast distances. Illicit gains can be acquired effortlessly through several means, such as computerized fund transfers. The majority of individuals engaging in technology fraud are young individuals who possess advanced cognitive abilities and are frequently inspired by accomplished peers. The detection and prevention of technological frauds pose significant challenges. There exist numerous global locations where fraudulent activities might be carried out by individuals. The advancement of technology is an ongoing and uninterrupted progression. As the identification and prevention of a specific fraudulent act are underway, concurrent efforts are being made to devise alternative methodologies.

2.3.3.2 Legal Causes of Fraud

The legal system may contribute to or facilitate instances of fraudulent behaviour through the following mechanisms: Most fraudulent cases are classified as "bailable offenses," resulting in perpetrators often evading legal consequences even upon apprehension (Fligstein & Roehrkasse, 2016). The prosecution of fraud necessitates adherence to the principles of "due process of the law." This necessitates a meticulous and extensive investigative procedure prior to apprehending the perpetrators (Driel, 2018). On certain occasions, individuals who are considered "suspects" or those with a known history of engaging in fraudulent activities may be apprehended, only to be released by the court due to insufficient evidence. Documents that appear to be compelling evidence to an individual

without specialized legal knowledge are deemed insufficient within the context of the law. The presence of corruption within the law enforcement sector of the legal system, including the police and judiciary, might inadvertently facilitate the activities of individuals engaging in fraudulent behaviour. Usually, individuals engaging in fraudulent activities opt to resolve their legal predicaments through settlement agreements.

2.3.3.3 Personal Causes of Fraud

Personal factors have been identified as key contributors to the occurrence, perpetuation, or encouragement of financial fraud. Among these, there are professional criminals whose primary focus is defrauding corporate entities and financial institutions. These individuals often recruit others with compromised moral character, capitalizing on their vulnerabilities. Research indicates that certain individuals possess an insatiable appetite for adventure, criminal or otherwise, and are inclined to steal when opportunities arise, regardless of their social status or material wealth—traits commonly observed in kleptomaniacs (Fligstein & Roehrkasse, 2016). Furthermore, the moral upbringing of individuals varies significantly; while some parents emphasize this crucial aspect of character development, others delegate the responsibility to teachers, religious leaders, or mentors. A poor choice of associates or mentors can expose individuals to fraudulent networks, where they may be lured by generous gifts or financial incentives before being drawn into illicit activities (Driel, 2018). Interestingly, some individuals from seemingly stable, well-educated, and religious backgrounds may still be susceptible to manipulation due to weak character. In some cases, fraudsters leverage influential connections, such as "crime fathers" or powerful friends, to thwart investigations. These backers, often well-connected figures, may intervene, using their influence to shield the perpetrators from justice, allowing entire syndicates to evade accountability.

2.3.3.4 Management Causes of Fraud

Management actions, or lack thereof, can create an environment conducive to fraud within financial institutions (Driel, 2018). Several factors contribute to this, including the recruitment of staff without conducting thorough background checks from reliable sources, such as educational institutions and previous employers. Additionally, placing undue emphasis on academic qualifications at the expense of actual performance can lead to issues such as certificate manipulation and other fraudulent behaviours. Furthermore, when an organization is publicly perceived, through statements from key officials, to be capable of offering higher remuneration but instead provides inadequate compensation, it can fuel

discontent and incentivize fraudulent activities. Weak internal controls, the absence of timely reconciliations of accounts, and delays or neglect in conducting regular internal and external audits further exacerbate the risk of fraud (Fligstein & Roehrkasse, 2016).

2.3.4 Financial Fraud Detection

In recent times, there has been a significant surge in the focus on fraud detection in credit card transactions, which has garnered considerable interest among scholars and researchers. In a recent study, Darwish (2019) introduced a novel approach for credit card fraud detection using a two-level model. The researcher model addressed the challenge of imbalanced datasets and leverages the semantic fusion of the k-means algorithm and the artificial bee colony algorithm (ABC). The suggested approach aims to improve the accuracy of classification and expedite the convergence of fraud detection. The ABC algorithm, when utilized as a secondary classification level, employs a hybrid approach that combines local neighbourhood search with global search. This approach is employed to address the limitation of the k-means classifier, which fails to accurately identify the true cluster when the input data is presented in a different order, potentially resulting in varying cluster outcomes. Moreover, the k-means classifier can be susceptible to local optima because of its sensitivity to the initial conditions.

The proposed solution utilized a rule engine integrated into it to evaluate the authenticity of transactions in a dataset. This evaluation is based on various consumer behaviour indicators, such as geographical areas, usage frequency, and book balance. The experimental findings demonstrated that the proposed model had the capability to improve the accuracy of categorization in detecting and mitigating the risks associated with suspicious transactions. Furthermore, the model exhibited superior accuracy when compared to conventional methods. While this hybrid approach improved classification accuracy and convergence speed, its reliance on heuristic optimization may pose challenges in scalability and interpretability when applied to large-scale, real-world datasets. In contrast, this study's use of automated feature engineering integrated within a deep learning framework offers enhanced scalability and adaptability by automatically extracting relevant features without extensive manual tuning, thus better supporting real-time fraud detection requirements outlined in the system architecture.

In their research, Olowookere and Adewale (2020) proposed a conceptual framework that merges the advantages of meta-learning ensemble methods with the cost-sensitive learning paradigm for fraud detection. The framework was designed to enable base classifiers to be trained using traditional methods while integrating cost-sensitive learning into the ensemble process to develop a cost-sensitive meta-classifier. This approach eliminates the need to apply cost-sensitive learning to each individual base classifier. The evaluation of both the meta-classifier and the base classifiers was conducted by measuring their predictive accuracy using the Area Under the Receiver Operating Characteristic (AUC) curve. The results from classifying previously unseen data revealed that the cost-sensitive ensemble classifier achieved a strong AUC score, indicating reliable performance across varying fraud rates within the dataset. The study's findings underscore the effectiveness of the cost-sensitive ensemble framework in generating classifiers that can accurately detect fraudulent transactions in diverse payment system databases, regardless of fraud prevalence. Additionally, the performance of the cost-sensitive ensemble classifiers was found to exceed that of traditional ensemble classifiers. However, the proposed model's complexity and the necessity for cost-sensitive parameter tuning may hinder deployment in dynamic fraud environments. Reflecting on these limitations, the proposed system in this study incorporates SMOTE for class balancing and an MLP classifier that inherently adapts to class imbalance, thereby streamlining model training and reducing the overhead of cost parameter optimization.

Alwadain et al. (2023) proposed a novel approach for predicting financial fraud using machine learning. They used transaction-level features from 6,362,620 transactions in a paysim synthetic dataset and fed them into various machine-learning classifiers. The correlation between different features was also analysed. Additionally, approximately 5,000 more data samples were generated using a Conditional Generative Adversarial Network for Tabular Data (CTGAN). The evaluation of the proposed predictor demonstrated higher accuracies, outperforming previously existing machine-learning-based approaches. Among the 27 classifiers tested, XGBoost achieved the highest accuracy score of 0.999. When evaluated through exhaustive repeated 10-fold cross-validation, XGBoost still maintained a high average accuracy score of 0.998. These findings were particularly relevant for financial institutions and provided valuable insights for regulators and policymakers aiming to develop effective policies for mitigating financial fraud risk. While their use of synthetic data generation addresses data scarcity and imbalance, synthetic samples may not fully

capture complex temporal and behavioural fraud patterns observed in real transaction logs. This study builds upon such approaches by incorporating sequence modelling and temporal feature synthesis within automated feature engineering, capturing richer transaction context. Moreover, the dataset's limitations in regional specificity noted in Alwadain et al.'s work are mitigated in the current research by focusing on Kenyan financial transaction patterns, enhancing contextual relevance and compliance with local regulatory requirements such as the Kenya Data Protection Act (2019).

Hajek et al. (2022) proposed an XGBoost-based fraud detection framework that accounted for the financial consequences of fraud detection systems. The framework was empirically validated using a large dataset of more than 6 million mobile transactions from the Paysim dataset. To demonstrate the framework's effectiveness, a comparative evaluation was conducted on existing machine learning methods designed for modelling imbalanced data and outlier detection. The results indicated that, in terms of standard classification measures, the proposed semi-supervised ensemble model—integrating multiple unsupervised outlier detection algorithms and an XGBoost classifier—achieved the best results. Additionally, the highest cost savings were achieved by combining random under-sampling with XGBoost methods. This study provided financial implications for organizations, enabling them to make informed decisions regarding the implementation of effective fraud detection systems. However, semi-supervised models demand extensive unlabelled data and complex tuning, which may conflict with the low-latency requirements of real-time fraud detection systems. The architecture developed in this study prioritizes efficient feature extraction and rapid inference through an MLP model deployed via a Flask API, aligning better with practical implementation constraints in Kenyan financial institutions.

In their study, Ashfaq et al. (2022) introduced a machine learning method that utilizes blockchain technology for the purpose of enhancing the security of digital transactions. The model under consideration has the capability to make predictions regarding the fraudulent nature of incoming transactions within the blockchain. The machine learning methods under consideration were trained and evaluated using a dataset derived from bitcoin transactions. The objective was to forecast the patterns and characteristics of future transactions. The provided dataset consisted of 30,047 entities, with a relatively lesser proportion of entities classified as fraudulent. The researchers employed the Synthetic Minority Over-sampling Technique (SMOTE) to generate artificial instances of malicious data points, with the aim of improving the outcomes of their study. The researchers employed XGboost and random

forest algorithms for model classification and afterwards computed the confusion matrix. The implementation of this classification scheme facilitated the model's ability to discern between false and authentic data. The simulation results demonstrated that the algorithm proposed in this study effectively detects instances of transaction fraud. Additionally, the system was tested against vulnerabilities and assaults using two attacker models to assess its effectiveness. The system under consideration demonstrated resilience against both double-spending and Sybil assaults. Nonetheless, blockchain-based approaches can suffer from high computational costs and limited scalability for high-frequency transaction environments typical of banking systems. Given the transactional volumes encountered in Kenyan banking, this study focuses on scalable deep learning models optimized for transactional data rather than blockchain integration, though future work may explore hybrid architectures.

Ileberi et al. (2022) introduced a credit card fraud detection engine that leverages machine learning (ML) techniques in conjunction with the genetic algorithm (GA) for feature selection. The proposed engine utilizes a variety of ML classifiers, including Decision Tree (DT), Random Forest (RF), Logistic Regression (LR), Artificial Neural Network (ANN), and Naive Bayes (NB), following the optimization of relevant features. To assess the performance of the detection engine, a dataset consisting of European cardholders was analysed. The results indicated that the proposed approach outperformed existing fraud detection systems available at the time of the study. While the inclusion of GA for feature selection highlights the importance of optimizing feature relevance, this method introduces additional computational overhead that may limit scalability in high-frequency transaction environments. Unlike their approach, this current research emphasizes automated feature engineering integrated within a deep learning MLP classifier, which reduces reliance on manual or heuristic feature selection techniques. This enables more adaptive and scalable fraud detection suited to the dynamic nature of Kenyan financial transactions, consistent with the system architecture’s goal of operational efficiency.

Lucas et al. (2020) explored the use of machine learning and data mining techniques for detecting credit card fraud. They observed that most existing studies treated credit card transactions as isolated events, neglecting the sequential nature of these transactions. To address this gap, the authors developed a novel approach that emphasized feature engineering and sequence modelling to enhance fraud detection. The data preparation process was structured around three key perspectives. First, the sequences were categorized

based on whether they contained fraudulent transactions. Second, the sequences were analysed by fixing either the cardholder or the payment terminal, allowing the model to focus on patterns specific to individuals or locations. Third, the sequences were examined in terms of temporal information, such as the amounts spent, or the time elapsed between consecutive transactions. By combining these three perspectives, the authors created eight distinct sets of transaction sequences, ensuring a comprehensive and diverse representation of the data.

Feature engineering played a critical role in the study. The authors employed Hidden Markov Models (HMMs) to model the sequential nature of transactions. Each HMM assigned a likelihood to a transaction based on the preceding transactions within the sequence. These likelihoods, representing temporal and contextual patterns, were added as new features to the dataset. This approach enabled the automated generation of features that captured temporal correlations, complementing traditional expert-engineered features. The enhanced dataset, containing both HMM-derived features and expert features, was then used to train a Random Forest classifier for fraud detection. The combination of these features allowed the model to leverage both the temporal dependencies in transaction sequences and domain-specific knowledge. This integration proved effective in improving the model's ability to detect fraudulent transactions. The study demonstrated that incorporating sequence modelling and automated feature engineering could significantly enhance the performance of fraud detection systems. By addressing the limitations of treating transactions as isolated events, Lucas et al. (2020) provided a framework that not only captured temporal relationships but also improved the overall accuracy and robustness of fraud detection methods. This research strongly influenced the system design in this study, where automated feature engineering was prioritized to capture temporal and behavioural nuances, allowing the MLP classifier to perform effectively without the need for manual feature crafting. The approach aligns with the system's objective to reduce domain expert dependency and increase adaptability to emerging fraud trends.

Wedge et al. (n.d.) proposed an automated feature engineering approach aimed at significantly reducing false positives in fraud prediction, a persistent challenge in the fraud detection industry. False positives are a critical issue, with estimates suggesting that only 1 in 5 transactions flagged as fraud are genuinely fraudulent, and approximately 1 in 6 customers experiencing declined valid transactions within the past year. To address this challenge, the authors employed the Deep Feature Synthesis (DFS) algorithm to

automatically generate behavioural features from the historical transaction data associated with each card. In total, 237 features were derived, capturing over 100 distinct behavioural patterns for each transaction. These features were subsequently utilized to train a Random Forest classifier aimed at predicting fraudulent activities. The proposed solution was tested on a dataset comprising 1.852 million transactions from a large multinational bank and compared against the bank's existing fraud detection system. The results demonstrated a 54% reduction in false positives and yielded cost savings of 190,000 euros. Additionally, the authors explored the practicality of deploying this solution and whether it required real-time streaming computation for scoring. Their findings revealed that the benefits of the model could be maintained even when historical features were updated only once every seven days. This study highlighted the effectiveness of automated feature engineering in improving fraud detection systems by reducing false positives, enhancing accuracy, and optimizing resource utilization. Wedge et al. (n.d.) provided a practical and impactful framework for advancing fraud prediction in the financial industry. The current study builds on this research by employing a similar automated feature engineering strategy but adapts it to the Kenyan financial context using the Paysim dataset and additional feature sets. Furthermore, by deploying the MLP classifier via a Flask API, this study addresses the practical aspects of model deployment highlighted as critical by Wedge et al., ensuring that the fraud detection system can be effectively integrated into real-world banking infrastructure.

Yang et al. (2021) proposed an optimized Deep Feature Synthesis (DFS) method as part of a comprehensive framework designed to improve car loan fraud detection. Their approach sought to address several challenges typically associated with feature engineering, including feature dimensionality explosion, poor interpretability, extended training times, and suboptimal detection accuracy. To mitigate these issues, the authors compressed abstract and uninterpretable features by limiting the depth of the DFS algorithm, effectively reducing the complexity of the feature space while preserving essential information. The method was tested on a real-world car loan credit database to assess its performance. When compared to conventional automatic feature engineering techniques, the optimized DFS method demonstrated substantial improvements, reducing the number of features by 92.5%, cutting training time by 54.3%, and enhancing detection accuracy by 23%. These findings underscored the method's ability to streamline the feature engineering process while significantly improving the efficiency and effectiveness of car loan fraud detection. The

findings highlighted the potential of the optimized DFS approach to address key limitations in existing feature engineering methods. By improving interpretability, reducing computational demands, and boosting accuracy, Yang et al. (2021) provided a robust framework for advancing fraud detection in car loan credit systems. The optimization used by the researchers in this research parallels the challenges faced in the study, where managing feature complexity without sacrificing accuracy was a priority. The integration of automated feature synthesis in this study similarly seeks to balance the richness of behavioural and temporal features with computational efficiency, supporting real-time fraud detection needs in the Kenyan banking sector. Moreover, the interpretability improvements advocated by Yang et al. resonate with ongoing efforts in this study to ensure that model outputs are explainable enough to build trust among stakeholders, a factor critical for regulatory compliance and user acceptance.

2.3.5 Feature Engineering

Feature engineering is a pivotal process in developing machine learning models for financial fraud detection, involving the transformation of raw data into meaningful features (Verdonck et al., 2021). Effective feature engineering can significantly improve the performance of fraud detection systems by capturing complex patterns and behaviours indicative of fraud. Feature creation involves generating new variables that provide additional insights into the data. Two primary methods are commonly employed: feature aggregation and feature transformation. Feature aggregation combines multiple features to summarize information, often capturing user behaviour over time. For instance, aggregating transaction amounts over a specific period can reveal spending patterns that deviate from the norm, indicating potential fraud. Ikeda (2023) proposed a framework that utilizes feature aggregation based on customer behaviour to create effective feature candidates for fraud detection models. Feature transformation applies mathematical functions to existing features to uncover hidden patterns. Transformations such as logarithmic scaling or polynomial combinations can highlight nonlinear relationships within the data. In fraud detection, feature transformation can help in normalizing skewed data distributions, making it easier for models to learn from the data.

Manual feature engineering requires domain expertise and can be time-consuming. Automated feature engineering aims to streamline this process. Lucas et al. (2019)

introduced a method using multi-perspective Hidden Markov Models (HMMs) to model sequences of credit card transactions. This approach captures temporal correlations, providing automated feature generation that enhances the effectiveness of fraud detection models. Once new features are created, selecting the most relevant ones is crucial to prevent model overfitting and reduce computational complexity. Feature selection techniques evaluate the importance of each feature and retain those that contribute most to the model's performance. Ikeda (2023) emphasized the importance of feature selection in her framework, which evaluates all features and removes irrelevant or highly correlated ones to optimize the dataset for machine learning algorithms.

In fraud detection, behavioural features that capture user habits and transaction patterns are particularly valuable. For example, analysing the frequency and timing of transactions can help identify anomalies. A study by Lucas et al. (2019) demonstrated that modelling sequences of transactions from different perspectives, such as cardholder behaviour and payment terminal usage, can improve fraud detection by capturing temporal correlations.

Despite its benefits, feature engineering in financial fraud detection faces several challenges. Fraudsters continually adapt their methods, rendering static feature sets obsolete over time. Continuous monitoring and updating of features are necessary to maintain model effectiveness. Fraudulent transactions are typically rare compared to legitimate ones, leading to imbalanced datasets. This imbalance can bias models towards predicting non-fraudulent behaviour. Resampling strategies and the creation of features that highlight minority class characteristics are essential to address this issue. Access to detailed transactional data can be restricted due to privacy regulations, limiting the availability of information necessary for comprehensive feature engineering. Techniques that utilize aggregated or anonymized data can help mitigate privacy concerns while still providing valuable insights for fraud detection.

Recent research has focused on developing frameworks that integrate both feature creation and selection processes. Ikeda (2023) proposed a comprehensive feature engineering framework that combines feature aggregation and transformation to create effective feature sets for machine learning and deep learning algorithms in fraud detection. This approach emphasizes capturing complex patterns in customer behaviour to improve detection accuracy. Additionally, studies have explored the use of network-based features for fraud

detection. Azarm et al. (2024) investigated the potential of network-based features, such as those derived from the personalized PageRank algorithm, to capture the social dynamics of fraud by analysing relationships between financial accounts. Their findings suggest that integrating such features can enhance the predictive power of fraud detection models.

In conclusion, feature engineering remains a cornerstone of effective financial fraud detection. By transforming raw data into insightful features, it enables models to detect subtle patterns indicative of fraudulent behaviour. This research aim at utilizing automated feature engineering frameworks continue to enhance the efficiency and effectiveness of detecting fraud in financial transactions. However, challenges such as dynamic fraud patterns, data imbalance, and privacy concerns necessitate continuous innovation and adaptation of feature engineering techniques to maintain robust fraud detection systems.

Table 2.1 Summary of Financial Fraud Literature.

Author	**Methods**	**Feature Engineering**	**Limitations**
Olowookere and Adewale (2020)	Ensemble Techniques	Manual	-The study was limited to only credit card fraud detection hence the findings of the research cannot be generalized to other types of financial fraud.
Ileberi et al. (2022)	Decision Tree (DT), Random Forest (RF), Logical Regression (LR), Artificial Neural Network (ANN), and Naive Bayes (NB)	Manual	- The study validated the proposed framework primarily on the European credit card fraud dataset and a synthetic dataset, indicating a need for validation on more diverse and real-world datasets to ensure generalizability. - The research did not explore the computational complexity or efficiency of the proposed GA-based feature selection method, which could impact its scalability and practical application in large datasets.
Darwish (2019)	K-Means	Manual	-The reliance on the K-means classifier, which is sensitive to the initial conditions and can produce different clusters if the same data is inputted in a different order, introduces variability and potential inconsistency in the fraud detection process. This

			sensitivity may undermine the reliability of the model's performance in practical applications.
Ashfaq et al. (2022)	XGboost, Random Forest	Manual	-The models developed in this study are susceptible to adversarial attacks. -The accuracy of the models in this research can be improved.
Ileberi et al. (2022)	Decision Tree, Random Forest, Logistic Regression, ANN, Naïve Bayes) and genetic algorithm (GA) for feature selection.	Manual	-Did not explore advanced models beyond traditional machine learning and focused on optimizing existing features.
Yang et al. (2021)	Optimized Deep Feature Synthesis (DFS) method for car loan fraud detection	Automated	-Focused on car loan fraud and did not explore broader applications of DFS in other fraud detection contexts.
Lucas et al. (2020)	Hidden Markov Models (HMMs).	Manual and Automated	-The authors still relied on manual feature engineering and did not explore fully automated feature extraction methods.

			-Despite achieving commendable accuracy, the research still did not achieve optimal accuracy level required for such sensitive models.
Wedge et al. (n.d.)	Automated feature engineering using Deep Feature Synthesis (DFS) to generate behavioural features for fraud detection.	Automated	- Limited to a specific dataset and may not be scalable to all types of fraud detection scenarios. -The researchers only explored one algorithm and did not compare its performance with other advanced models.

2.4 Algorithms

The majority of academics and researchers' attention has recently been drawn to the enormous growth in fraud detection. The numerous methods for spotting fraudulent transactions are examined in this section.

2.4.1 Machine Learning

Machine learning (ML) is a comprehensive concept including a diverse array of algorithms designed to make intelligent predictions by analysing a given dataset (Nichols et al., 2018). The datasets in question are frequently of substantial size, perhaps encompassing millions of distinct data points. In recent times, significant advancements have been made in the field of machine learning, resulting in the achievement of a level of semantic comprehension and information extraction that closely resembles that of humans. In certain cases, machine learning algorithms have even demonstrated superior accuracy in detecting abstract patterns compared to human specialists. Building upon traditional statistical modelling methods, contemporary machine learning has arisen as a formidable tool owing to the substantial increase in data quantities, exponential advancements in processing capabilities, and progress in algorithmic design, propelled by the demands of web-based companies.

There is currently a diverse range of machine learning algorithms, commonly referred to as models, that are being utilized. The selection of an appropriate model for a given problem depends on both the characteristics of the data and the desired outcome. A critical factor to consider is the volume of distinct data points available. When dealing with large datasets, typically consisting of around one million unique data points, more advanced deep learning techniques are often warranted (Nichols et al., 2018). Conversely, a reduction in the number of data points suggests that traditional methods, such as linear regression or decision-tree algorithms—which partition datasets into regions based on predefined criteria—are likely to deliver superior performance. It is essential to exercise careful consideration in tailoring the methodology to suit the specific nature of the data, whether it involves images, time-series signals, or more general descriptive data.

2.4.1.1 Supervised Learning

Supervised learning is a fundamental approach in machine learning, where the primary objective is to train a model to map input data to corresponding output values through a set of example input-output pairings (Sarker, 2021). This process relies on annotated training data and a collection of instances to derive a mathematical function that captures the

underlying relationships. Supervised learning is employed when specific objectives are to be achieved from a given set of inputs (Sarker, 2021), signifying a method directed by task requirements. The two most common supervised learning tasks are "classification," which involves categorizing data into distinct groups, and "regression," which focuses on fitting data to a mathematical model. An illustrative example of supervised learning is text categorization, where the goal is to predict the class label or sentiment of a given piece of text, such as a tweet or product review.

2.4.1.2 Unsupervised Learning

Unsupervised learning is a computational technique designed to analyze datasets that lack labels or annotations, thereby eliminating the need for human intervention. This approach is distinguished by its reliance on data-driven processes (Sarker, 2021). It is widely used for extracting generative features, identifying significant patterns and structures, clustering data, and conducting exploratory analyses. Common tasks in unsupervised learning include clustering, density estimation, feature learning, dimensionality reduction, association rule discovery, and anomaly detection. These methods enable the identification of inherent structures within the data without prior knowledge or labeling.

2.4.2 Classification Algorithms

The field of financial fraud detection has experienced an increase in the variety of classification approaches utilized. Several strategies have emerged as major means to enhance the detection of fraudulent financial transactions. These techniques include Random Forests (RF), Artificial Neural Networks (ANN), Support Vector Machines (SVM), and k-Nearest Neighbours, each possessing unique strengths and capabilities. Moreover, there has been an increasing tendency towards the adoption of novel methodologies that involve a combination of different approaches, resulting in the emergence of hybrid strategies. These novel frameworks not only improve the precision of detection but also prioritize the utmost importance of data privacy, demonstrating a dedication to protecting confidential financial data.

2.4.2.1 Random Forest

The random forest method is a machine learning technique derived from the decision tree algorithm, commonly applied to a variety of regression and classification tasks. This approach enables the precise prediction of outcomes in datasets of considerable size. By aggregating multiple classifiers, the random forest methodology effectively addresses

complex problems, leveraging the collective strength of various decision trees to enhance overall predictive accuracy and robustness. The random forest algorithm is utilized to forecast the average mean of output based on the predictions made by multiple individual trees. The augmentation of tree population often leads to an enhancement in the accuracy of the result.

The random forest methodology is particularly effective in mitigating several limitations inherent in the decision tree algorithm (Darwish, 2019). Furthermore, it diminishes the necessity for manual dataset manipulation, thereby improving overall accuracy. A random forest is composed of multiple decision trees, each serving as a weak learner. However, when these trees are aggregated, they collectively form a robust and highly accurate learner, capitalizing on the strength of the ensemble approach. The RF approach is characterized by its ability to process big datasets and handle imbalanced data in an efficient and expedient manner. Nevertheless, the random forest algorithm exhibits several limits when it comes to effectively training diverse datasets, particularly in the context of regression issues.

While the random forest algorithms have demonstrated efficacy in identifying regression problem classes, they present some drawbacks in real-time CCFD scenarios. The model has strong performance on laboratory-based datasets characterized by a scarcity of available data. The performance of random forest algorithms is comparatively slower in real-time applications. The training process exhibits a reduced pace, necessitating a greater duration for prediction generation. Hence, to achieve efficient Cross-Correlation Fire Detection (CCFD) in practical datasets, a substantial amount of data is required. However, it should be noted that random forest algorithms exhibit limitations in efficiently training such datasets and generating accurate predictions.

2.4.2.2 Artificial Neural Network (ANN)

Artificial Neural Networks (ANNs) are machine learning algorithms designed to mimic the functioning of the human brain. Typically, ANNs operate through two primary approaches: supervised and unsupervised learning methods. The unsupervised neural network is widely applied in fraud detection due to its high accuracy rate of 95% (Bin Sulaiman et al., 2022). This approach identifies patterns by comparing the behaviours of current credit cardholders with those observed in past transactions. When details from a current transaction align with patterns of previous fraudulent activities, a potential fraud case is flagged. Notably, ANNs demonstrate remarkable fault tolerance, as they can continue generating outputs even when

one or more components are compromised. With their rapid processing capabilities and robust performance, ANNs present a highly suitable solution for detecting financial fraud efficiently and effectively.

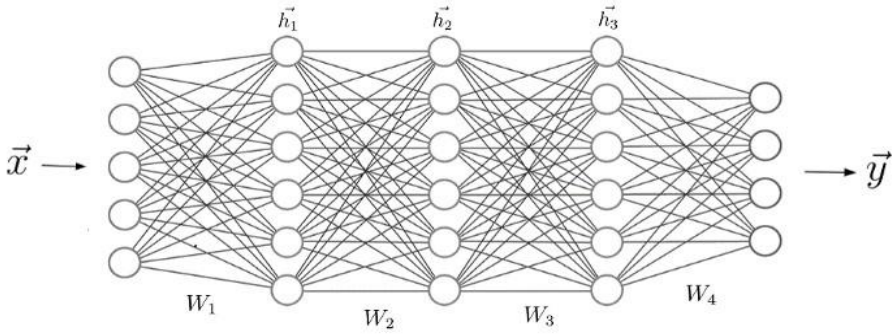


Figure 2.1: A Neural Network (Oppermann, 2019).

2.4.2.3 K Nearest Neighbour

The K-Nearest Neighbours (KNN) algorithm is a supervised machine learning technique widely utilized for both classification and regression tasks. As an effective supervised learning approach, KNN enhances detection capabilities and reduces the occurrence of false alarms. In the context of financial fraud detection, KNN plays a crucial role in identifying fraudulent behaviour in financial transactions by assessing two key factors: the correlation between transactions and the distance between transaction occurrences in the dataset. This method is particularly well-suited for detecting fraudulent activity over transactional periods.

KNN's application, coupled with over-sampling techniques and data segregation, facilitates the identification of anomalies in target variables, making it a valuable tool for financial fraud detection. It is especially useful in addressing memory constraints, as it leverages limited memory and computational resources, thus offering an efficient solution for processing datasets of varying sizes. Compared to traditional anomaly-based methods, KNN demonstrates superior accuracy and efficiency in detecting fraudulent activities.

In practice, KNN is often employed to identify patterns in historical transactions, particularly in credit card fraud detection. When evaluated, KNN has shown a remarkable accuracy rate of 97.69% in identifying fraudulent transactions. The method's effectiveness is further confirmed by its ability to minimize false positives, making it a robust choice for fraud detection systems. A subsequent study using KNN yielded a 72% accuracy rate in

classifying financial fraud detection (FFD), highlighting its strong performance across multiple evaluation criteria.

Table 2.2 Summary of Classification Techniques

Technique	Short Description	Strengths	Drawbacks
Random Forest (RF)	Combines multiple decision trees for regression and classification tasks, improving prediction accuracy (Darwish, 2019).	Handles large datasets efficiently, reduces overfitting, and improves accuracy by averaging multiple trees.	Slower training and prediction time in real-time applications; requires large data for effectiveness.
Artificial Neural Network (ANN)	Mimics the human brain to detect patterns, often used for unsupervised fraud detection with high accuracy (Bin Sulaiman et al., 2022).	High fault tolerance, fast processing capabilities, and effective for detecting patterns in complex data.	Requires large datasets and significant computational resources; can be difficult to interpret.
K-Nearest Neighbours (KNN)	Uses proximity and correlation to classify data, suitable for detecting fraudulent transactions (KNN accuracy ~97.69%).	High accuracy and efficient for datasets of varying sizes with minimal false positives.	Memory-intensive and may struggle with large datasets or overlapping data points.

2.5 Models and Frameworks

2.5.1 Frameworks

2.5.1.1 TensorFlow

TensorFlow, an open-source software framework for numerical computation, employs data flow graphs to facilitate machine learning (ML) and deep learning (DL) tasks. Developed and managed by the Google Brain team within Google's Machine Intelligence research

division, TensorFlow is distributed under the Apache 2.0 open-source license (TensorFlow, 2018). The framework is designed to support large-scale training and inference, particularly within distributed computing environments. In the context of a data flow graph, mathematical operations are represented as nodes, while the communication of multidimensional data arrays, known as tensors, occurs through the graph's edges.

TensorFlow's distributed architecture consists of master and worker services, which incorporate kernel implementations for enhanced computational efficiency. The framework provides a broad range of operations, totalling over 200 standard functionalities, including mathematical computations, array manipulation, control flow, and state management, all implemented in C++. This extensive set of features makes TensorFlow highly versatile, capable of supporting a wide array of applications across research, development, and production systems.

Notably, TensorFlow operates seamlessly across diverse computing environments, ranging from single CPU systems to mobile devices, GPUs, and large distributed systems with hundreds of nodes. TensorFlow Lite, a streamlined version of the framework, is specifically optimized for mobile and embedded devices, offering improved performance and reduced binary size while maintaining core functionalities (TensorFlow Lite, 2018). While it supports fewer operators compared to the full TensorFlow framework, it ensures efficient machine learning inference on devices with minimal delay, further enhanced by hardware acceleration via the Android Neural Networks API.

The TensorFlow programming interfaces include APIs for Python and C++, with ongoing efforts to extend support for Java, Go, R, and Haskell. Additionally, TensorFlow is compatible with cloud environments such as those provided by Google and Amazon, further expanding its accessibility and application across various platforms.

2.5.1.2 Torch

The Torch framework is a scientific computing tool that offers extensive support for machine learning methods. It is primarily built on the Lua programming language (Torch 2018). The project has been continuously developed since its inception in 2002, as documented by Collobert et al. (2002). Torch is a widely adopted framework that has garnered backing and utilization from prominent organizations such as Facebook, Google, DeepMind, Twitter, among others. It is noteworthy because Torch is openly accessible under the BSD license. The programming language has an object-oriented approach and is coded

in C++. In the present day, the API of the system is further implemented using the Lua programming language, serving as a layer that encapsulates very efficient C/C++ and CUDA code.

At the heart of the system lies the Tensor library, which is compatible with both CPU and GPU backends, providing a robust foundation for computational tasks. The library encompasses a comprehensive suite of traditional operations, particularly those pertinent to linear algebra, facilitating efficient processing across diverse hardware configurations. These operations are implemented in an efficient manner using the C programming language, taking advantage of SSE instructions on Intel platforms. Furthermore, the library offers the flexibility to integrate linear algebra operations with established, high-performance BLAS/Lapack implementations, such as Intel MKL (Collobert et al., 2011). In addition, the framework is optimized to harness parallel processing capabilities, leveraging OpenMP for multi-core CPUs and CUDA for GPUs, thereby enhancing computational efficiency across diverse hardware architectures. This technology is mostly employed in the context of extensive learning, encompassing voice, image, and video applications. It is utilized for several types of learning, including supervised learning, unsupervised learning, reinforcement learning, as well as neural networks, optimization techniques, graphical models, and image processing.

2.5.1.3 Keras

Keras is a Python-based framework designed as a wrapper to seamlessly integrate various deep learning tools, including TensorFlow, CNTK, Theano, the beta version with MXNet, and the recently introduced Deeplearning4j (Keras, 2018). Created with a focus on enabling rapid experimentation, Keras is distributed under the MIT license. The framework supports Python versions 2.7 through 3.6 and is capable of running on both GPUs and CPUs, contingent on the underlying frameworks. Developed and actively maintained by François Chollet, Keras is guided by four core principles that shape its ongoing development and refinement.

The concepts of user friendliness and minimalism are important considerations in the design and development of various products and systems. Keras is an application programming interface (API) that has been specifically built to prioritize user experience. Keras adheres to established principles for minimizing cognitive burden by providing APIs that are both consistent and straightforward. The concept of modularity refers to the degree to which a

system or a process is composed of separate and independent components that can be A model is defined as a series or a network of independent, highly adaptable modules that can be interconnected with few limitations. The various components involved in the creation of new models include neural layers, cost functions, optimizers, initialization schemes, activation functions, and regularization methods. These components can be combined as separate modules. One notable advantage of this system is its ease of extension. The addition of new modules is straightforward, and the presence of already existing modules offers enough examples, which in turn aids in minimizing the need for excessive expressiveness.

2.5.2 Models

2.5.2.1 Random Forest

Random forests represent an ensemble learning approach that integrates multiple decision tree predictors to enhance predictive accuracy. In this methodology, each tree within the forest is constructed using values derived from a randomly sampled vector. This vector is sampled independently for each tree, maintaining a consistent distribution across all trees in the ensemble. Liu et al. (2015) conducted a study wherein they described the use of Random Forest (RF) for the detection of financial fraud techniques. The researchers also discussed the process of selecting features, measuring the importance of variables, doing partial correlation analysis, and performing Multidimensional analysis. The findings of their study demonstrated that the utilization of a composite of eight variables yielded the maximum level of accuracy. The variable of utmost significance in the model was the debt-to-equity ratio (DEQUTY). Furthermore, the researchers employed four statistical approaches, encompassing both parametric and non-parametric models, in order to develop detection models. Their findings indicated that Random Forest exhibited the best level of accuracy, while the non-parametric models shown superior accuracy compared to the parametric models. Nevertheless, the utilization of Random Forest has the potential to substantially enhance the efficiency of detection, hence carrying considerable practical implications.

2.5.2.2 Artificial Neural Network (ANN)

Artificial Neural Networks (ANNs) are computational models designed to perform specific tasks by mimicking the functioning of the human brain. These networks consist of hundreds or even thousands of artificial neurons, or processing units, that work in tandem. The implementation of an ANN involves creating a learning algorithm that allows the system to autonomously derive its own rules of behaviour, thereby obviating the need for manual

programming of each individual rule. This process, often referred to as "experience," allows the network to adapt and refine its responses. The practical feasibility of neural networks is rooted in their architecture as massively parallel computing systems, comprising numerous interconnected processing units (neurons) that learn from their environment. The synaptic weights, which represent the strength of these interconnections, are adjusted through a sequential, supervised learning algorithm to meet predefined objectives (Montesinos López et al., 2022). Research indicates that, through collaborative operation, neurons can effectively learn both complex linear and nonlinear input-output relationships via sequential training procedures. While the foundational inspiration for these models differs from that of traditional statistical models, the core components of neural networks share notable similarities with those of statistical approaches.

In their study, RB and Kumar (2021) introduced a method for detecting fraud in credit card transactions using Artificial Neural Networks (ANN). The researchers initially compared this approach with other machine learning algorithms, such as k-Nearest Neighbors and Support Vector Machines. Despite the challenges in training the model to achieve optimal performance for fraud detection, they ultimately concluded that the ANN approach, which demonstrated an accuracy rate approaching 100%, was the most suitable for credit card fraud detection. The performance of the ANN model surpassed that of unsupervised learning algorithms. To address the issues associated with imbalanced datasets, the researchers implemented data pre-processing, normalization, and under-sampling techniques.

Table 2.3 Summary of Models

Model	**Author**	**Limitations**
Random Forest (RF): Combines multiple decision trees using random vectors for financial fraud detection. Developed detection models using parametric and non-parametric methods, with RF showing the best accuracy.	Liu et al. (2015)	Requires selecting significant variables; computationally expensive for large datasets; slower for real-time applications.
Artificial Neural Network (ANN): Mimics the human brain to detect fraud, achieving nearly 100% accuracy	RB and Kumar (2021)	Difficult to train; computationally intensive; requires extensive data pre-

in credit card fraud detection. Utilized data pre-processing, normalization, and under-sampling to handle imbalanced datasets.		processing and tuning for optimal performance.

2.6 Gaps in the Existing Systems

Despite advancements in machine learning and the use of diverse data sources, financial fraud detection faces several challenges, including issues with cost, effectiveness, bias, and robustness. A major problem is model bias, which often results from class imbalance, where fraudulent activities are much less common than legitimate transactions. This imbalance can lead to poor model performance in detecting fraud, as well as other biases related to under-representation, neglect of sensitive features, and social feedback loops. Also, traditional deep learning models are vulnerable to adversarial attacks, making it crucial to develop more robust and resilient detection systems.

In the context of the Kenyan financial sector, there is a notable gap in the research landscape pertaining to financial fraud detection. Surprisingly, there has been a scarcity of studies that delve into this crucial area, particularly studies that harness advanced models. In a rapidly evolving and increasingly sophisticated landscape of financial fraud, it's imperative that players in the financial sector embrace cutting-edge approaches. However, most of the existing studies have leaned on traditional machine learning models, perhaps unaware of the immense potential that advanced methodologies hold. This situation highlights the need for a paradigm shift in the approach to fraud detection within the Kenyan financial sector, one that harnesses the power of advanced models to safeguard against the ever-growing threats posed by financial fraud.

Many existing studies on financial fraud detection have relied on manual feature engineering, which often leads to consistent patterns across different models. This approach limits the ability to uncover new anomalies, particularly in complex datasets, and causes models to become overly dependent on historical features, struggling to adapt to evolving fraud tactics. Additionally, manual feature engineering is time-consuming, requires substantial human expertise, and is prone to biases. To address these inefficiencies, automated feature engineering methods, such as automated feature selection, can uncover hidden patterns without human intervention, improving adaptability and reducing biases.

A key contribution of this research is the use of the Multi-Layer Perceptron (MLP) Classifier, a deep learning architecture designed to enhance model performance. The MLP consists of multiple layers, each serving a specific purpose. The input layer captures raw features from the dataset, while hidden layers, using activation functions like ReLU, help the model learn non-linear relationships. The addition of dropout layers mitigates overfitting by randomly "dropping" neurons during training, thus improving generalization. The final output layer provides a binary classification decision, determining whether a transaction is fraudulent or not.

This MLP architecture is advantageous because it automatically refines features through its layers, enhancing the model's ability to detect complex fraud patterns. By automating feature refinement and enhancing model robustness, the MLP classifier becomes more resilient to adversarial attacks and evolving fraud tactics, leading to improved fraud detection accuracy and efficiency. This contribution fills a significant gap in existing fraud detection models by offering a more adaptable, efficient, and less biased approach to feature engineering and model learning.

2.7 Conceptual Model

The financial transactions dataset will undergo a rigorous pre-processing phase, where raw data is transformed into a clean and structured dataset. This foundational step sets the stage for subsequent data refinement. Automated feature engineering will follow suit, leveraging advanced techniques to extract and engineer the most pertinent features. Hyperparameter tuning, a critical optimization step, will then come into play, fine-tuning the model's parameters to achieve optimal performance. It's only after this preparation that the model-building phase, with a focus on deep learning, will commence. Rigorous testing will be done to ensure that the model stands resilient and robust before it's entrusted with the task of discerning the authenticity of financial transactions, thereby fortifying the defences against fraudulent activities.

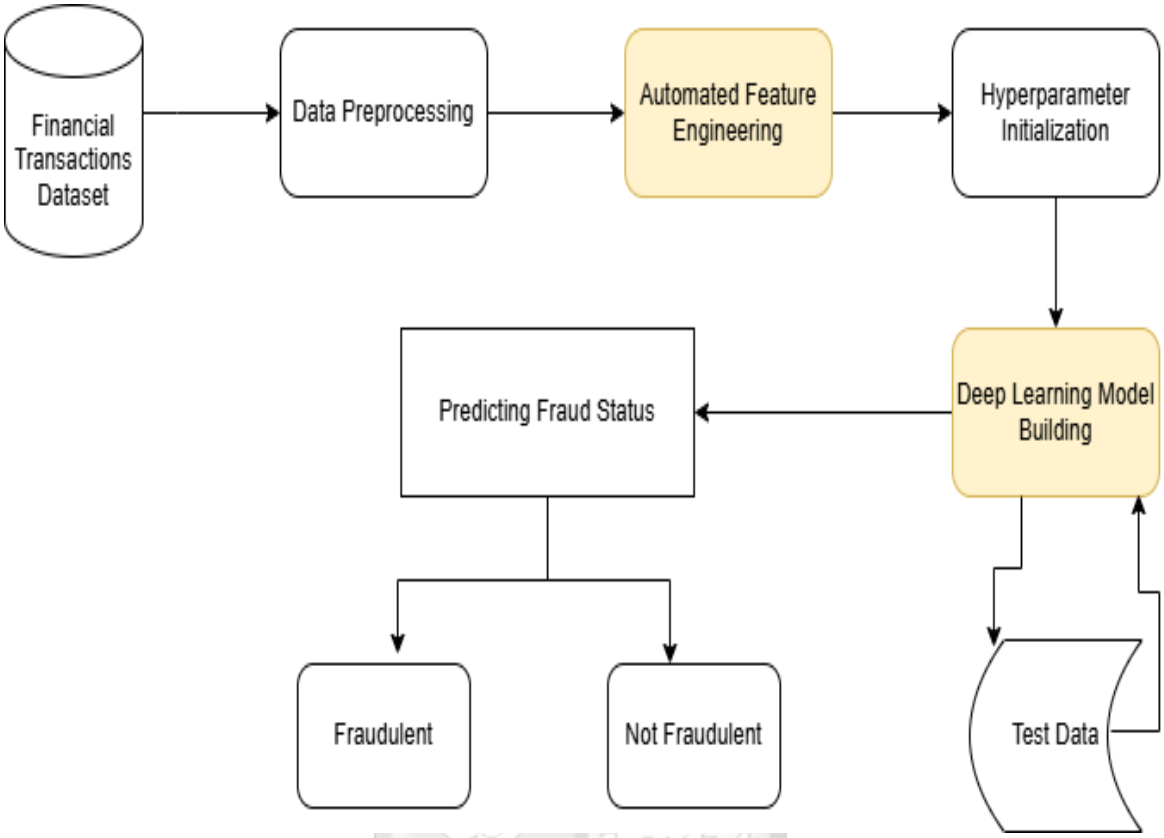


Figure 2.2: Conceptual Model

Chapter 3: Research Methodology

3.1 Introduction

Given the inherent dissimilarity between various forms of study, it is imperative for the researcher to customize their research approach to effectively address the unique topic at hand. The methodology employed by the researcher and the underlying rationale constitute the strategy utilized in this study. This chapter presents the research design, which is informed by the research objectives. In addition, the chapter examines the system development methodology, focusing on its implementation and the validation of its feasibility. Specifically, the utilization of agile methodology as a system development methodology is selected due to its ability to allow software development iteratively.

3.2 Research design

The primary objective of a research design was to provide an appropriate framework for conducting a research study (Sileyew, 2019). A key component of this process involved decisions related to the research strategy, which significantly influenced the methods through which relevant data would be gathered. It was essential to recognize that the research design process involved a series of interconnected decisions. For the present study, an experimental research design was chosen. This approach was selected due to the study's focus on developing a prototype, which aligned with the requirements of experimental research.

3.3 Target Population

Although this study was centered on Kenyan banks and other financial institutions as the subject of investigation, direct data collection from these institutions was not conducted. Due to privacy concerns and the sensitivity of the data, banks were understandably reluctant to share such information freely. Consequently, the study instead utilized secondary data for training and testing purposes. This approach allowed the research to maintain its focus on the complex dynamics of financial fraud while ensuring that the necessary data was available to develop a highly efficient tool for reducing fraudulent activities in financial transactions.

3.4 Sample Size

The whole dataset, comprising 6,048,576 data records, was utilised. To mitigate the potential bias in the model and avoid the misleading impression of enhanced model accuracy, the dataset was partitioned into two subsets, with 80% allocated for training purposes and the remaining 20% reserved for testing. The utilisation of the 80/20 rule is prevalent in the field

of machine learning due to its ability to strike a reasonable equilibrium between model training and evaluating its efficacy based on the given dataset. The practice of allocating 80% of the available data for training purposes and reserving the remaining 20% for testing enables the model to be trained on a sufficiently large sample size, hence facilitating the identification and understanding of underlying patterns within the data. This approach also allows for the estimation of the model's ability to generalise its performance when presented with fresh, unseen data (Dobbin & Simon, 2011).

3.5 Data collection

This study utilized secondary data obtained from Kaggle to develop a reliable model for identifying instances of financial fraud, specifically designed for implementation within regional banking institutions. The utilisation of secondary data is crucial in light of the privacy and confidentiality restrictions maintained by most financial institutions, which prohibit disseminating their data to external organisations. This methodology enables the use of pre-existing information to construct a model that can augment the capabilities of fraud detection, while simultaneously upholding the principles of privacy and security inherent in the financial sector. The dataset includes the following attributes:

i). **step**: Represents the time step in which the transaction occurred, serving as a proxy for the temporal sequence of transactions.

ii). **type**: Denotes the type of financial transaction (e.g., cash-out, transfer).

iii). **amount**: The transaction amount in monetary units.

iv). **nameOrig**: The originating account name or identifier.

v). **oldbalanceOrg**: The originating account's balance before the transaction.

vi). **newbalanceOrig**: The originating account's balance after the transaction.

vii). **nameDest**: The destination account name or identifier.

viii). **oldbalanceDest**: The destination account's balance before the transaction.

ix). **newbalanceDest**: The destination account's balance after the transaction.

x). **isFraud**: A binary label indicating whether the transaction was fraudulent (1 for fraud, 0 for non-fraud).

xi). **isFlaggedFraud**: A binary label specifying whether the transaction was flagged as fraud by the system (1 for flagged, 0 for not flagged).

This dataset is structured to simulate a real-world banking environment, capturing key features that influence financial fraud detection. This dataset has been utilized in several studies, including those by Alwadain et al. (2023), Hwang and Kim (2020), and Moreira et al. (2022).

3.6 Research Quality and Reliability

3.6.1 Data Reliability

The reliability of a measuring device is defined as the degree to which it consistently measures the same attribute (Noble & Smith, 2015). It also refers to how reliably different practitioners using the same instrument can achieve the same outcomes under the same conditions. The greater the consistency between measurements of the same attribute, the more trustworthy the instrument is. During the agile development training phase, the output data from the model was tested multiple times. The performance metrics were calculated for the model's assessment, and the system's efficacy was evaluated in every possible way.

3.6.2 Data Validity

The data utilized in the study underwent a rigorous validation process to guarantee their accuracy in reflecting the provided inputs and their seamless incorporation into the model. This validation procedure was crucial to maintaining the integrity and reliability of the research. It involved checks and cross-referencing to confirm that the data accurately represented the variables and parameters necessary for the model's functioning. This diligence helped ensure that the subsequent analysis and findings were founded on trustworthy, high-quality data, a fundamental requirement for robust and credible research.

However, several threats to the validity of the data had to be addressed. One potential issue was sampling bias, where sure fraud or demographic groups may have been overrepresented or underrepresented in the dataset, leading to inaccurate or skewed results. To mitigate this, careful attention was given to ensuring that the dataset was representative of the diverse fraud types and population segments encountered in real-world scenarios. SMOTE was also used to address class imbalance since most transactions were not fraudulent.

Another threat was the generalizability of the findings. Suppose the data used in the study was drawn from a limited set of financial institutions, geographic locations, or fraud types. In that case, the results may not have applied universally across all contexts. Efforts were made to ensure the data was diverse and covered a broad range of fraudulent activities,

improving the model's ability to generalize across various environments. Finally, model overfitting remained a significant concern. Deep learning models are susceptible to overfitting when trained on limited or imbalanced data, potentially leading to a model that performed well on training data but poorly on unseen data. Cross-validation and regularization techniques were employed to address this issue, ensuring the model-maintained generalizability and robustness when applied to new, real-world data. Addressing these threats to data validity was essential to ensure that the findings of this research were not only accurate but also applicable in a variety of contexts, thereby strengthening the reliability and robustness of the fraud detection model.

3.7 System Development Methodology

The Agile Development Systems Methodology was employed to develop the financial fraud detection tool. Agile development encompasses a range of iterative and incremental software development strategies, which incorporate techniques such as Scrum, Crystal, and Lean development. The research used a lean technique to facilitate the development of the software. The technique selection for this research is based on its iterative character, which facilitates ongoing feedback during the development process and subsequent software system refinement and delivery. Figure 3.1 presents a comprehensive depiction of the agile technique. The process encompasses ongoing planning, design, construction, testing, evaluation, and iterative deployment phases. These iterations are called sprints in the context of agile project management methodologies. The Manifesto for Agile Software Development explains the subsequent four principles of agile development:

i). The prioritisation of individuals and interactions over procedures and instruments is emphasised.

ii). Prioritising functional software above extensive documentation.

iii). The act of collaborating with consumers during the process of contract negotiation;

iv). The ability to adjust to change rather than rigidly adhering to a predetermined strategy.

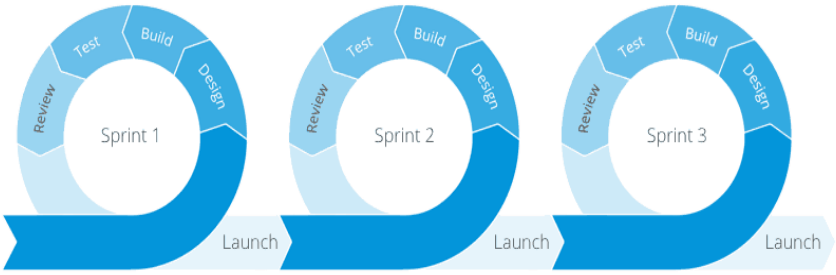


Figure 3.1: Agile methodology (Moniruzzaman & Hossain, 2013)

3.7.1 Planning

This phase involved defining project objectives and establishing requirements for automating feature engineering in fraud detection using DL (Deep Learning). Key tasks were broken into manageable sprints, focusing on identifying relevant fraud detection features from financial transaction data. The scope, milestones, and timelines will be set, emphasising flexibility to accommodate iterative changes based on real-time feedback.

3.7.2 Design

This phase entailed developing the architecture for the automated feature engineering and ANN model. The design includes the structure of the neural network, feature selection methods, and data flow between system components. Decisions regarding algorithm selection and how to integrate feature engineering tools within the fraud detection framework were finalised. The design phase also considers scalability and integration of additional fraud detection models.

3.7.3 Construction

During the construction phase, the core components of the deep learning tool and feature engineering automation were implemented. The development followed iterative sprints, ensuring that each new feature is developed, integrated, and tested continuously. Implementation focused on building the model's feature extraction capabilities, developing the training data pipeline, and incorporating necessary machine learning libraries for fraud detection.

3.7.3.1 Automated Feature Engineering

Automated feature engineering is a critical step in this study, aimed at transforming raw transaction data into meaningful and relevant features to enhance the model's predictive accuracy. The process began with data pre-processing, where the dataset is cleaned to address missing values, outliers, and inconsistent formats. Raw features such as account

balances, transaction amounts, and timestamps are then transformed into engineered features. These transformations include calculating transaction frequency, average transaction amounts, and balance change rates over time, which provide deeper insights into transaction patterns.

To ensure the most relevant features were utilised, feature selection techniques such as Recursive Feature Elimination (RFE) and mutual information analysis are applied. Categorical variables like transaction types are converted into numerical formats through one-hot or label encoding. Automated tools like Feature Tools in Python were employed to generate aggregate statistics, such as the mean and maximum transaction amounts grouped by account ID or transaction type. Python and libraries like Pandas, NumPy, and Scikit-learn were used extensively to automate and streamline the feature engineering process.

3.7.3.2 Model Building

The deep learning model for fraud detection was developed using an Artificial Neural Network (ANN) architecture built in Python, leveraging TensorFlow and Keras frameworks. The model consisted of an input layer that processed the engineered features, followed by multiple hidden layers with ReLU activation functions to capture non-linear relationships in the data. Dropout layers were included to reduce overfitting, and the model ended with an output layer comprising a single neuron with a sigmoid activation function, which produced the probability of a transaction being fraudulent. To train the model, the binary cross-entropy loss function was used, as it suited binary classification problems. The Adam optimizer was employed to efficiently perform gradient descent during training. The training process used an 80-20 train-validation split, and hyperparameter tuning was conducted using grid search to optimize parameters such as learning rate, number of neurons, and batch size. A prototype of the model was developed in Python, using Google Colab for experimentation and iterative refinement.

3.7.3.3 Fraud Detection Tool

The fraud detection tool provided an interactive interface for deploying the trained model and allowed users to input transaction data for fraud prediction. The application's backend was developed using PHP, which managed all server-side logic and API interactions. The frontend was designed using HTML and styled with Tailwind CSS to create a responsive

and user-friendly interface. An API was developed in Flask to serve as the intermediary between the frontend and the ANN model. The API accepted input transaction data, processed it, and sent it to the model for prediction. The model, saved in a serialised format, was loaded into a Python runtime environment using Flask. The API then returned the fraud probability score to the frontend, which displayed the results to the user. The development of the tool followed an iterative process, allowing for continuous improvements in functionality and design. This integration of PHP for backend logic, Python for model execution, and Tailwind CSS for frontend styling ensured the tool was both efficient and user-friendly.

3.7.3.4 Testing and Validation

The testing and validation of the model and tool were conducted using a range of metrics to ensure reliability and effectiveness. For the model, accuracy was assessed to measure the overall correctness of fraud predictions. Precision evaluated the proportion of correctly identified fraudulent transactions, while recall measured the proportion of actual fraud cases detected by the model. The F1-score, a harmonic mean of precision and recall, provided a balanced evaluation of the model's performance. Additionally, the ROC-AUC metric was used to analyze the trade-off between true positive and false positive rates. For the tool, integration testing ensured seamless communication between the frontend, backend, and the model. Usability testing focused on the user experience, confirming the interface was intuitive and responsive. Performance testing evaluated the response time of the API and model predictions, ensuring the system could handle various input sizes efficiently. The prototype of the fraud detection tool was developed and tested iteratively, incorporating user feedback and performance metrics to refine both the model and the application. Python was the primary language for model execution, while PHP and Tailwind CSS managed the backend and user interface, respectively, ensuring a cohesive and functional system.

3.7.4 Testing

Testing was done concurrently with development, with each feature undergoing rigorous validation. Unit tests, integration tests, and overall system performance evaluations assessed the functionality and accuracy of the feature-engineered deep learning model. Metrics like accuracy, precision, recall, and F1-score were calculated for each iteration to ensure the model’s reliability in detecting financial fraud.

3.7.5 Evaluation

The evaluation phase focused on reviewing model performance after each iteration. Specific fraud detection metrics were analyzed to assess the model's effectiveness. Feedback from stakeholders and real-world testing results guided model modifications and improvements, refining the deep learning architecture and the feature engineering process.

3.7.6 Iterative Deployment

Deployment occurred incrementally, with each new iteration released into the system or testing environment. The model was continuously improved through deployment in controlled environments, using real-world financial transaction data to monitor and adjust the deep learning fraud detection capabilities.

3.8 Utilisation and Dissemination of Research Results

The continuous increase of fraudulent actions, with the substantial profits yielded by cybercrime, highlights the imperative requirement for heightened fraud detection measures. The potential for significant cost savings exists even with a slight enhancement in fraud detection rates. This research aims to provide Kenyan banks with the necessary tools and knowledge to detect and prevent fraudulent transactions before they cause substantial harm. The aforementioned findings provide a crucial resource for contemporary banking procedures and establish a foundation for future scholars exploring the emerging topic of study. Moreover, the findings of this research have the potential to enhance the government's continuous endeavours in addressing fraud. The dissemination of these significant findings will be aided through numerous channels, such as Strathmore's digital library and diverse publications, ensuring broad accessibility and usefulness.

3.9 Ethical Considerations / Issues

The strict adherence to ethical principles was of utmost importance in the execution of this research. The university's code of ethics conducted the study, and formal approval was obtained before conducting any research activities. The cornerstone of this commitment was integrity, with a steadfast dedication to honesty in all aspects of scientific communication. This included transparent reporting of all data, results, methods, procedures, and publication status. There was no fabrication, falsification, or misrepresentation of data under any circumstances. Additionally, there was a resolute commitment to transparency, ensuring that neither colleagues nor the public were deceived. The ethical conduct of this research served as its foundation, guaranteeing its integrity and dependability.

Chapter 4: System Analysis and Design

4.1 Introduction

Developing a robust financial fraud detection system demands a meticulous approach to analysing user requirements and designing an effective architecture. Functional requirements dictate the system's ability to process transactions and detect anomalies, while non-functional aspects ensure reliability, scalability, and security. Integrating advanced machine learning techniques and intuitive interfaces enhances the system's capability to identify fraudulent patterns. Clear design frameworks and workflows are instrumental in achieving a streamlined and efficient solution.

4.2 Requirement Specifications

Requirement specifications serve as the blueprint for any system, defining what it must achieve to meet user needs and operational goals. They provide a clear understanding of the system's functionality, performance, and constraints, ensuring alignment between stakeholders' expectations and technical implementation. Functional requirements outline the core tasks the system must perform, while non-functional requirements address broader aspects such as scalability, security, and usability. This structured approach ensures that the system is both effective and reliable, laying the groundwork for successful development and deployment.

4.2.1 Functional Requirements

The functional requirements outline the system's core functionality:

i). The system shall accept financial transaction data in CSV format for processing.

ii). The system shall identify potentially fraudulent transactions using a trained deep learning model.

iii). The system shall present analytical results in an interpretable format (e.g., tables, graphs).

iv). The system shall enable exporting results in a usable format (e.g., CSV or PDF).

v). The system shall **s**upport user registration and login for secure access.

4.2.2 Non-Functional Requirements

The non-functional requirements ensure system usability and efficiency:

i). **Scalability**: The system should handle large datasets with over a million records.

ii). **Performance**: Fraud detection should occur within acceptable response times (< 3 seconds per transaction batch).

iii). **Security**: Ensure secure handling of transaction data to maintain confidentiality.

iv). **Platform Independence**: Support deployment on various platforms, including web and mobile.

v). **Usability**: Provide an intuitive user interface for non-technical users.

4.3 System Architecture

The architecture of the financial fraud detection tool integrates multiple components to enable efficient, accurate, and scalable operations. At its core, the system combines a user-friendly interface, robust backend processing, and advanced machine learning capabilities to deliver a seamless fraud detection experience. The frontend interface, developed using HTML and Tailwind CSS, provides an intuitive and responsive environment for users to interact with the tool. It facilitates actions such as uploading transaction datasets, initiating fraud detection, viewing analytical results, and exporting reports. The design prioritizes simplicity and accessibility, ensuring that both technical and non-technical users can navigate the system with ease. The backend, implemented using Flask, acts as the processing engine that bridges the gap between the frontend and the machine learning model. It handles requests from the user interface, processes data, and communicates with the model to generate predictions. The lightweight and efficient nature of Flask ensures that the system maintains optimal performance even when processing large datasets. The database serves as a secure storage solution for user credentials, transaction records, and system outputs. It maintains the integrity of data and supports quick retrieval, enabling users to access historical logs and results effortlessly. The machine learning module, a critical component of the architecture, is designed to analyse transaction data and identify potentially fraudulent activities. Built using deep learning techniques, it incorporates automated feature engineering to extract meaningful insights from raw data, enhancing the model's predictive accuracy. The architecture also includes an API layer that connects the various components, ensuring seamless communication and data exchange. This modular design enables flexibility and scalability, allowing the system to adapt to changing requirements or

accommodate additional functionalities in the future. Together, these elements form a cohesive architecture that effectively supports the objectives of financial fraud detection.

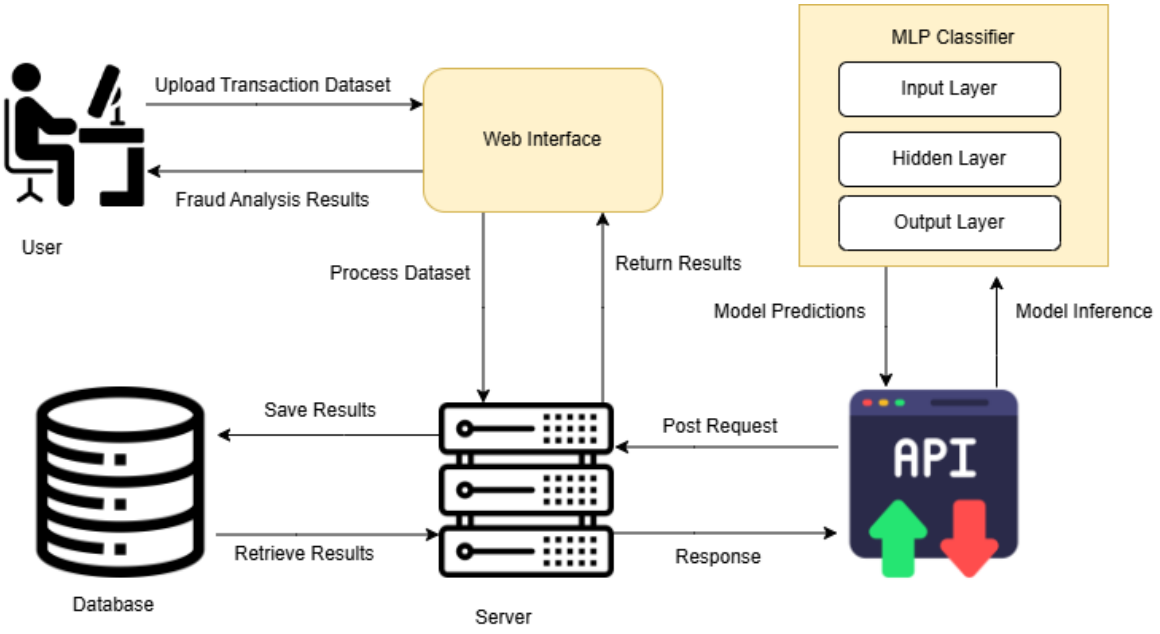


Figure 4.1: System Architecture

4.4 System Design

The system design focuses on defining the structural and behavioural aspects of the financial fraud detection tool, using Object-Oriented Analysis and Design (OOAD) principles. OOAD provides a systematic approach to modelling the system through a set of interacting objects, emphasizing modularity, reusability, and scalability. This methodology is particularly well-suited for a complex application like fraud detection, as it aligns closely with real-world concepts such as transactions, users, and models. The design process incorporates various visual representations, including use case diagrams, class diagrams, sequence diagrams, and database schemas, to ensure a comprehensive understanding of the system's functionality. These diagrams collectively illustrate the relationships and interactions between different components, guiding the development of a robust and efficient tool. Additionally, wireframes are used to depict user interfaces, highlighting how users will interact with the system to achieve their goals. By leveraging OOAD principles, the design ensures that the system is both flexible and maintainable, capable of adapting to evolving requirements in the financial sector.

4.4.1 Use Case Diagram

The use case diagram illustrates the system's functionality from the user's perspective, highlighting key interactions such as uploading datasets, initiating fraud detection, viewing results, and exporting reports. These use cases ensure that the system meets user expectations and requirements. The use case diagram identifies the primary actors in the system, including the end user (e.g., bank staff or administrators) and the system itself. It provides a high-level view of how different components interact to achieve the desired outcomes.

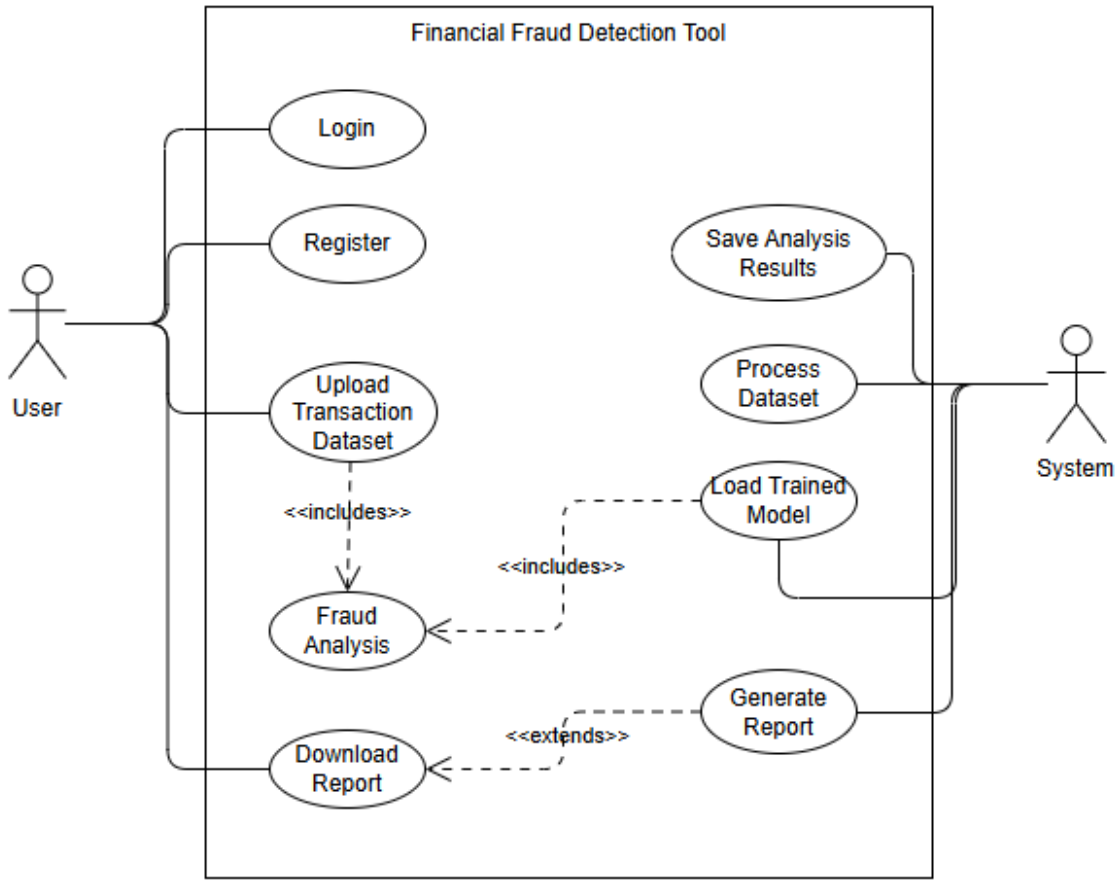


Figure 4.2: Use Case Diagram

4.4.1.1 Detailed Use Case Descriptions

Table 4.1 shows the detailed description of use cases in Figure 4.2

Table 4.1: Description of use cases

Use Case	Pre-Conditions	Main Success Scenario	Post Conditions
Data Upload	User is logged in.	User uploads a CSV file containing transaction data.	Data is validated and stored in the system.
Feature Generation	Transaction data is available.	The system generates new features using automated techniques.	Feature matrix is prepared for modelling.
Fraud Detection	Model is trained and validated.	The system predicts fraudulent transactions based on the input data.	Results are displayed in the dashboard.
Report Export	Detection results are available.	User exports results in a selected format.	Results are downloaded to the user's device.

4.4.2 Class Diagram

The class diagram defines the system's core entities, such as User, Transaction, FraudDetectionModel along with their attributes and methods. Each class encapsulates specific responsibilities. For example, the User class manages authentication and roles, while the Transaction class holds details of each transaction, such as amounts and account balances. The FraudDetectionModel class handles the loading, execution, and evaluation of the trained deep learning model.

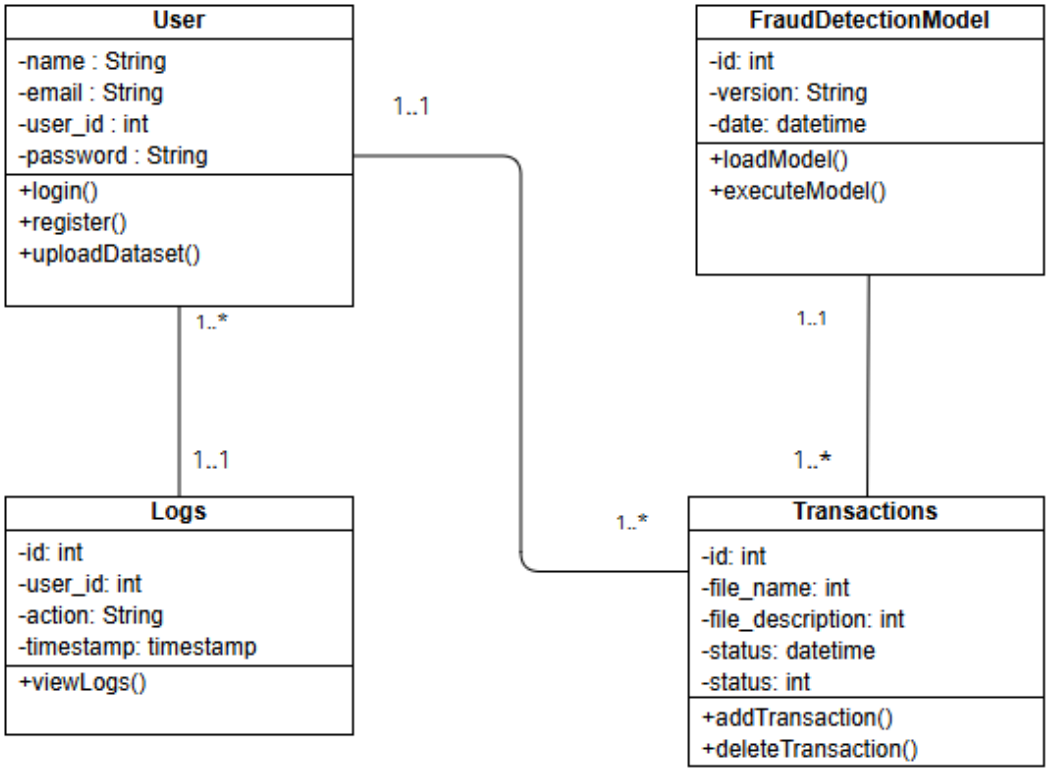


Figure 4.3 Class Diagram

4.4.3 Sequence Diagram

The sequence diagram provides a detailed view of the dynamic interactions between the components of the financial fraud detection system during specific operations. It captures the chronological order of messages exchanged between various entities, such as the user, the frontend interface, the backend, and the machine learning model, to accomplish a task like fraud detection. When a user initiates the fraud detection process, the sequence begins with the user uploading a dataset through the frontend interface. The frontend sends this dataset to the backend, where it is validated and pre-processed. During pre-processing, the data undergoes feature engineering to transform raw transaction details into structured and meaningful features that can be used by the machine learning model. Once the data is ready, the backend invokes the trained deep learning model to perform fraud detection. The model analyses the input features and generates predictions, assigning a fraud probability score to each transaction. These results are sent back to the backend, which formats them for display. Finally, the formatted results are transmitted to the frontend, where they are presented to the

user through an intuitive interface. The user can then review the flagged transactions, explore detailed insights, or export the results for further analysis.

Figure 4.4 illustrates the seamless communication and collaboration between the system's components, ensuring that the process of fraud detection is efficient, accurate, and user-friendly. By modelling the interactions in this manner, the system design ensures clarity and alignment during implementation.

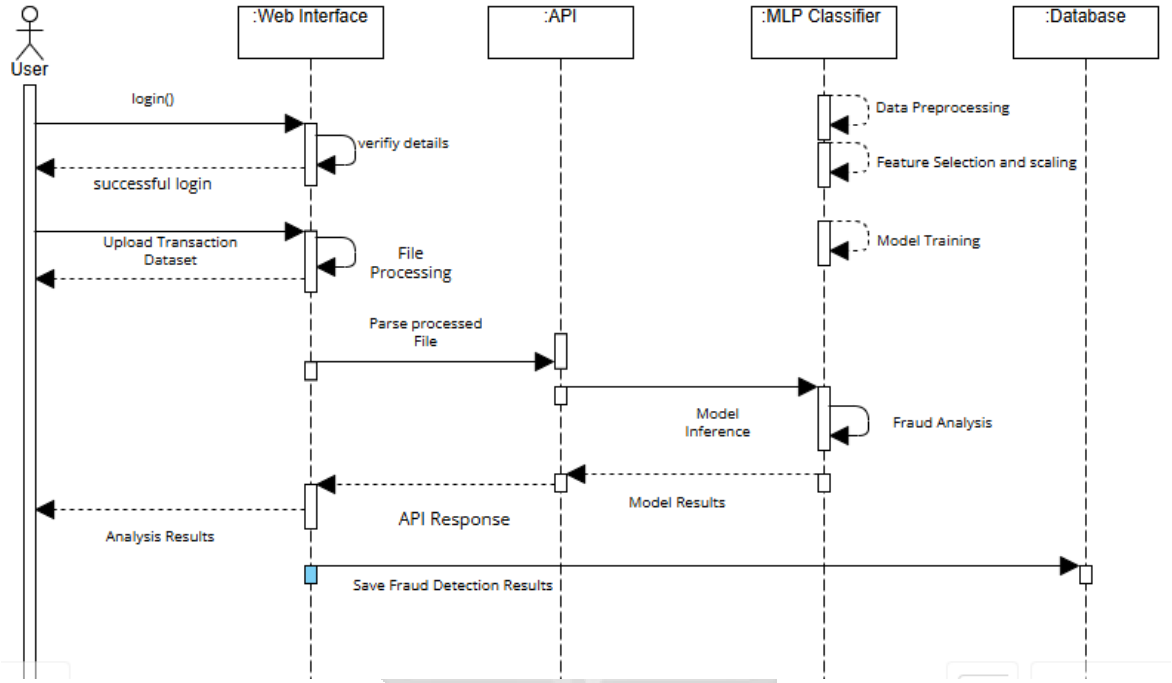


Figure 4.4: Sequence Diagram

4.4.4 Database Schema

The database schema is designed to efficiently support the storage and retrieval of data required by the system. It includes tables to manage users, transactions, and the fraud detection model. Each table is structured to represent core entities and their relationships. The schema also incorporates constraints and indexes for optimal performance and data integrity.

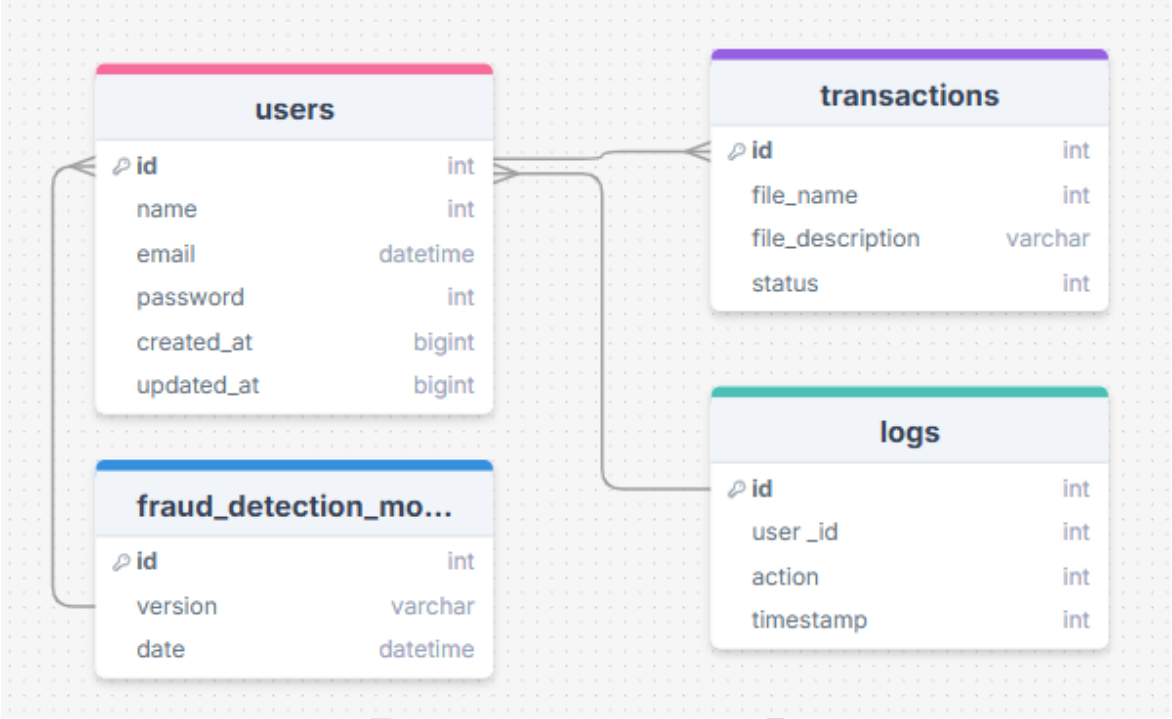


Figure 4.5: Database Schema

4.5 Wireframes

Wireframes are visual prototypes that outline the layout and design of the system's user interfaces. They help in visualizing the flow and placement of elements before full-scale development.

4.5.1 Home Page Wireframe

The home page serves as the entry point for the system, featuring navigation options, recent activity logs, and quick links to core functionalities such as uploading datasets or initiating fraud detection.

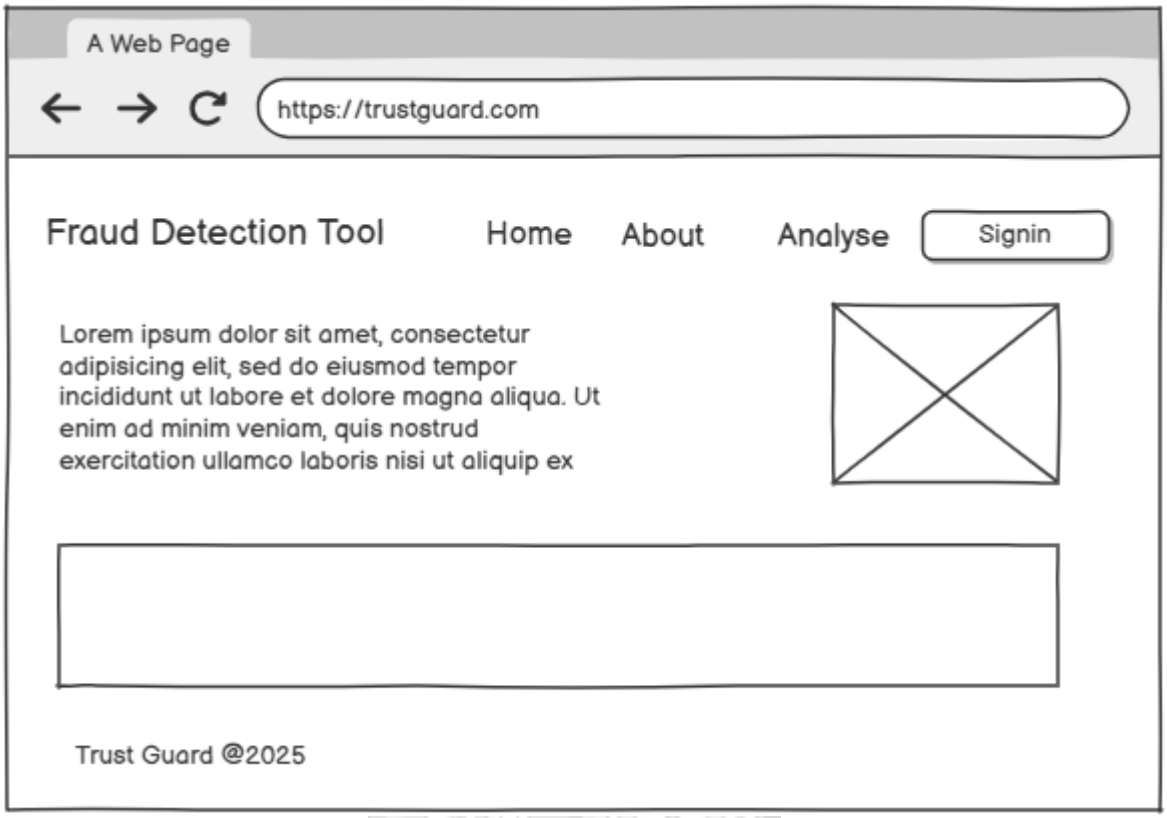


Figure 4.6: Home Page Wireframe

4.5.2 Login Wireframe

The login interface allows users to securely access the system by entering their credentials.

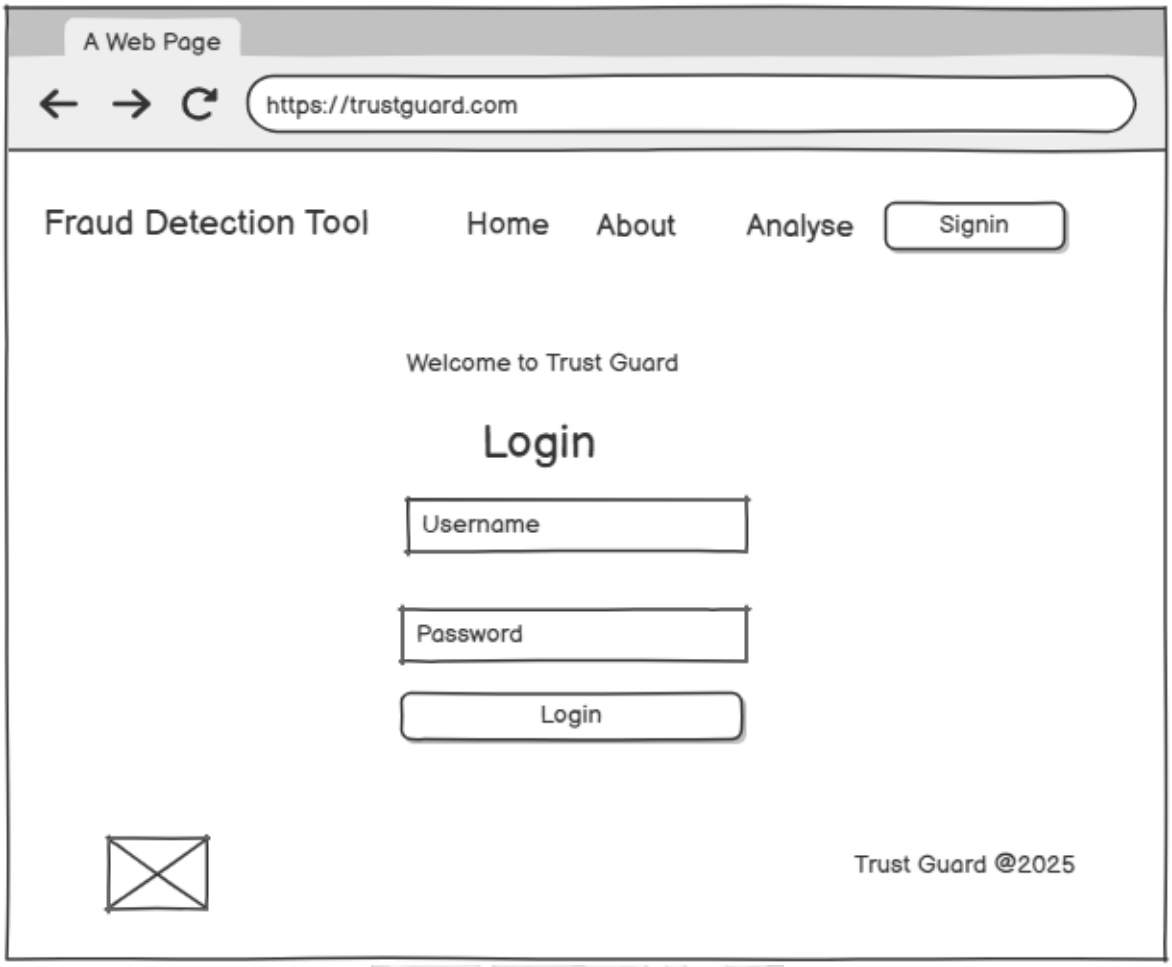


Figure 4.7: Login Wireframe

4.5.3 Register Wireframe

The registration page captures user details required for account creation.

Figure 4.8: Register Wireframe

4.5.4 Fraud Detection Wireframe

The fraud detection interface serves as a central hub for users to upload transaction datasets and initiate the fraud analysis process. This interface is designed to be user-friendly, ensuring smooth interactions for individuals with varying levels of technical expertise.

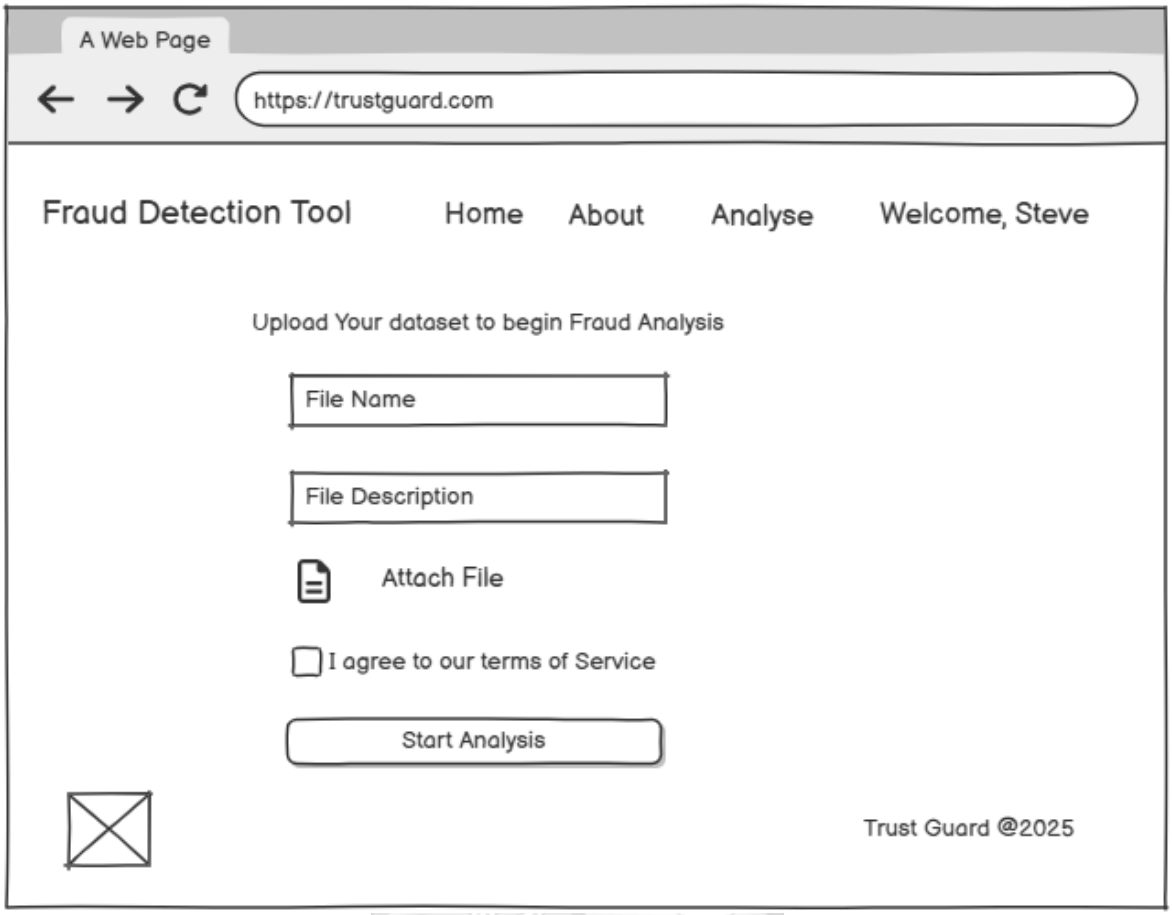


Figure 4.9: Fraud Detection Wireframe

4.5.5 Analysis Results Wireframe

The analysis results interface is designed to provide users with a detailed and intuitive view of the outcomes of the fraud detection process. This page displays a summary of the analysis, highlighting flagged transactions, fraud probability scores, and other critical insights.

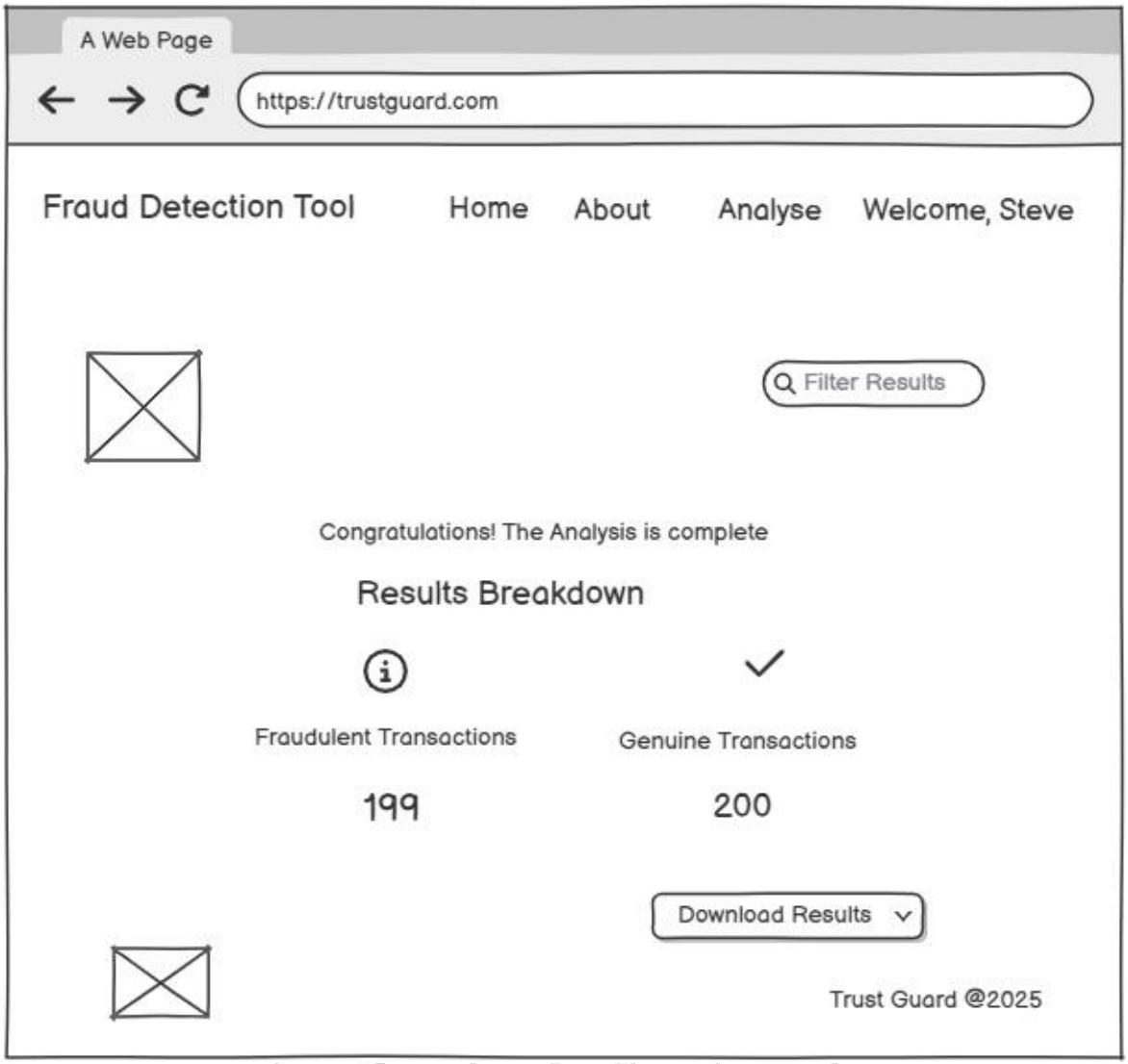


Figure 4.10: Analysis Results Wireframe

Chapter 5: System Implementation and Testing

5.1 Introduction

The financial fraud detection system was implemented by integrating a Multi-Layer Perceptron (MLP) Classifier with automated feature engineering and a user-friendly interface. The implementation process followed a structured approach, encompassing data pre-processing, feature engineering, model training, and system deployment. The system was developed and tested to ensure its effectiveness in identifying fraudulent financial transactions while maintaining a high level of usability and performance.

5.2 Model Components

The implementation relied on the MLP Classifier as the primary detection model, supported by automated feature engineering and pre-processing tools. These components ensured that the system accurately analysed transaction data and flagged suspicious activity.

5.2.1 Automated Feature Engineering

Automated feature engineering was implemented using the FeatureTools library in Python. This step transformed raw transaction data into meaningful features, such as cumulative sums, averages, and percentiles, which were essential for improving the MLP model's accuracy. This automated process replaced manual feature extraction, reducing the time and expertise required while ensuring scalability and adaptability to evolving fraud patterns. Figure 5.1 and 5.2 highlights the use of FeatureTools:

```
# Automatically create relationships and new features
feature_matrix, feature_defs = ft.dfs(
    entityset=es,
    target_dataframe_name="transactions",
    agg_primitives=["sum", "mean", "max", "min", "std"],
    trans_primitives=["cum_sum", "cum_mean", "percentile"]
)
```

Figure 5.1: Automated Feature Engineering

5.2.2 Multi-Layer Perceptron (MLP) Classifier

The Multi-Layer Perceptron (MLP) classifier was employed as the primary detection model. Its architecture consisted of an input layer, multiple hidden layers with ReLU activation functions, and a sigmoid output layer to estimate fraud probabilities. Dropout layers were incorporated between hidden layers to reduce overfitting. The model was trained using the Adam optimizer in conjunction with binary cross-entropy loss, which supported efficient learning and accurate classification.

The motivation for choosing an MLP over alternative deep learning architectures such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Deep Neural Networks (DNNs) lies in the specific characteristics of the transaction dataset and the problem context. CNNs are particularly suited for data with spatial hierarchies such as images or data with local patterns. Since transactional data is tabular and does not exhibit such spatial correlations, CNNs offer limited advantage in capturing relevant relationships. RNNs and their variants, such as LSTM or GRU networks, are designed to handle sequential data where temporal dependencies are crucial. While transaction time is a factor, the dataset in this case does not present long sequential dependencies that RNNs typically exploit. Additionally, the overhead in training RNNs and managing vanishing gradients makes them less efficient for this application.

DNNs, often understood as fully connected feedforward networks like MLPs but with deeper layers, could be considered a broader category including MLPs. The MLP used here effectively functions as a DNN with multiple layers, capturing complex nonlinear feature interactions without the additional complexity of architectures designed for sequence or spatial data. Therefore, the MLP provides a balanced approach that is well-suited to structured tabular data, offering flexibility and computational efficiency while adequately modelling the nonlinearities inherent in fraud detection. Figure 5.2 illustrates the base architecture adopted for the MLP classifier:

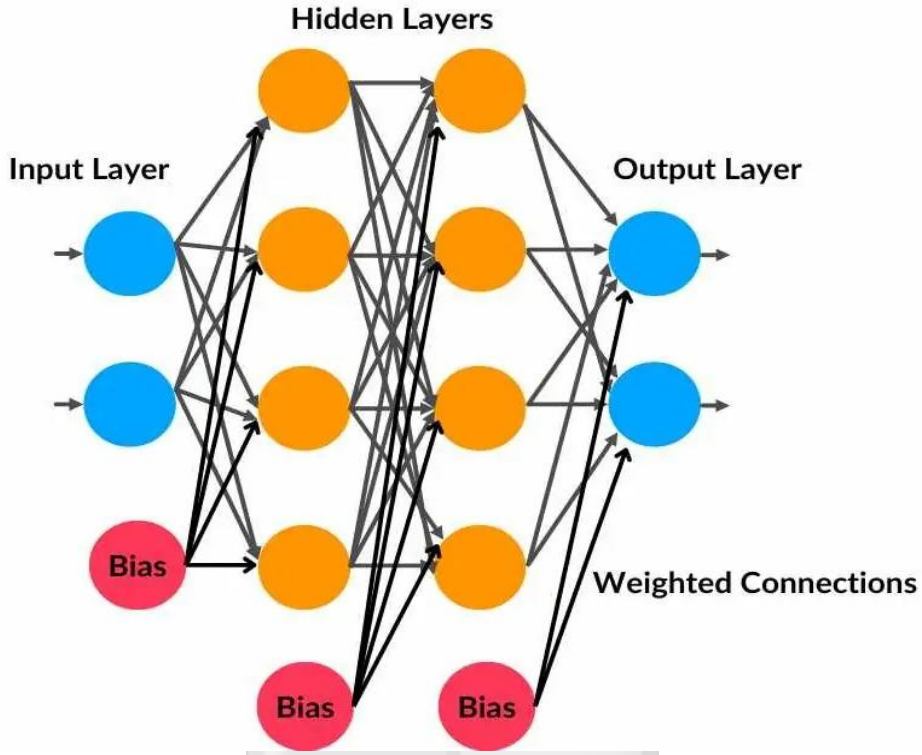


Figure 5.2 MLP Classifier

5.3 Financial Fraud Detection Tool

The tool combined the machine learning model with an interactive user interface, enabling users to upload datasets, perform fraud detection, and review results.

5.3.1 Home Interface

The home interface acts as the dashboard for users, offering navigation to key functionalities such as dataset upload, fraud detection, and results viewing. A summary section displays recent activities, such as the number of transactions analysed and flagged as fraudulent. The design prioritizes simplicity, ensuring users can quickly access desired features.

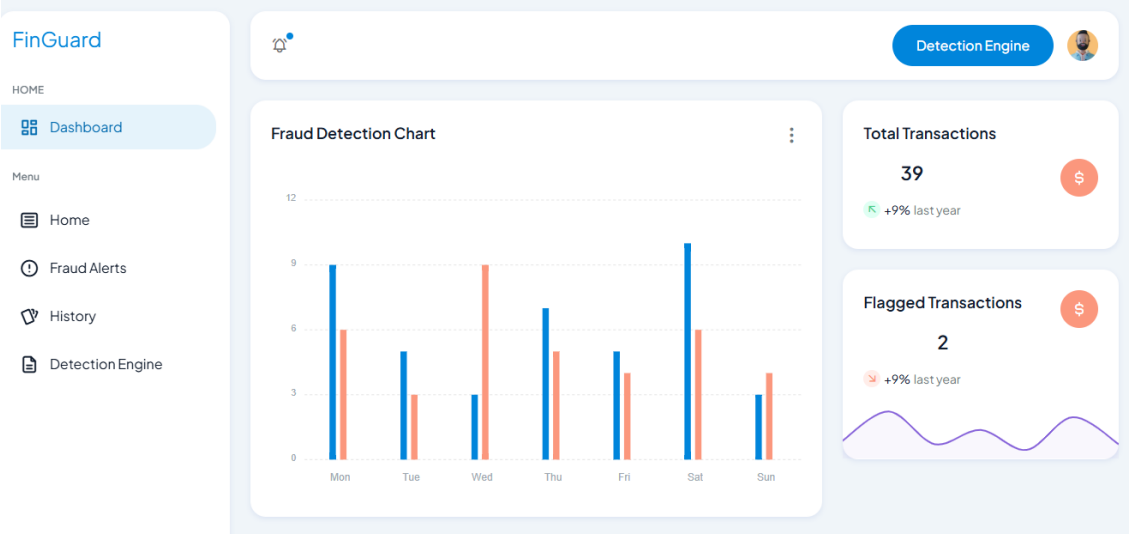


Figure 5.3: Home Page

5.3.2 Fraud Detection Interface

The fraud detection interface facilitates the upload and processing of transaction datasets. Users can upload files in formats such as CSV or Excel through a drag-and-drop or file-picker option. The interface validates uploaded files, ensuring they meet format and content requirements. Once validated, users can configure detection parameters, such as fraud thresholds, before initiating the analysis. Real-time feedback is provided during the detection process via a progress indicator. This ensures users remain informed about the status of their analysis.

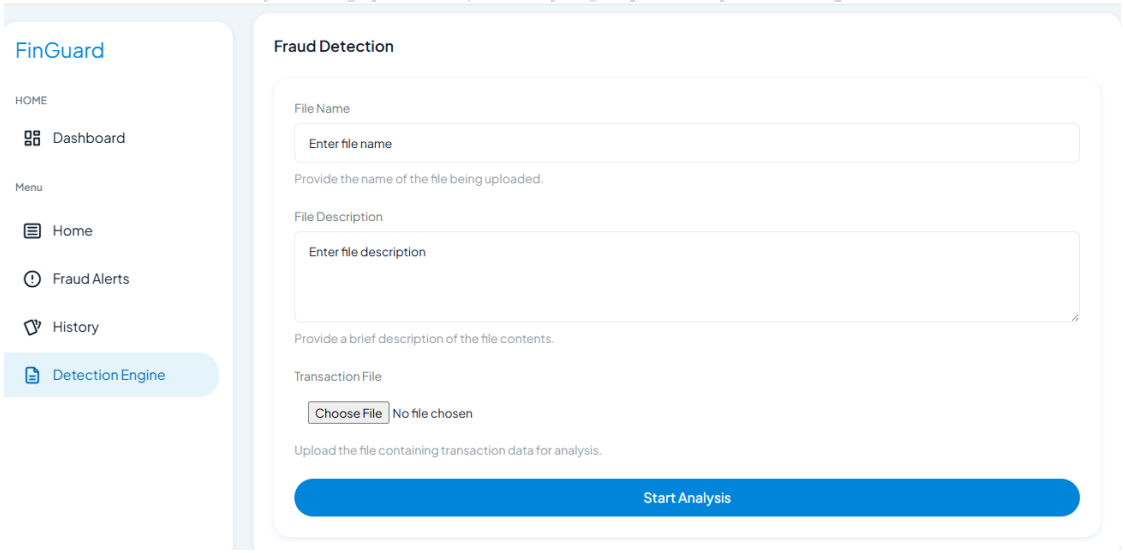


Figure 5.4: Fraud Detection Interface

5.3.3 Analysis Results Interface

The analysis results interface displays a detailed table of transactions, highlighting key attributes such as fraud probability scores, transaction amounts, and timestamps. Users can filter and sort the results to focus on flagged transactions or specific criteria, such as high-risk cases. Additional functionality allows users to drill down into individual transactions for detailed insights. The interface is designed to be interactive, enabling users to explore the data effectively and make informed decisions based on the results.

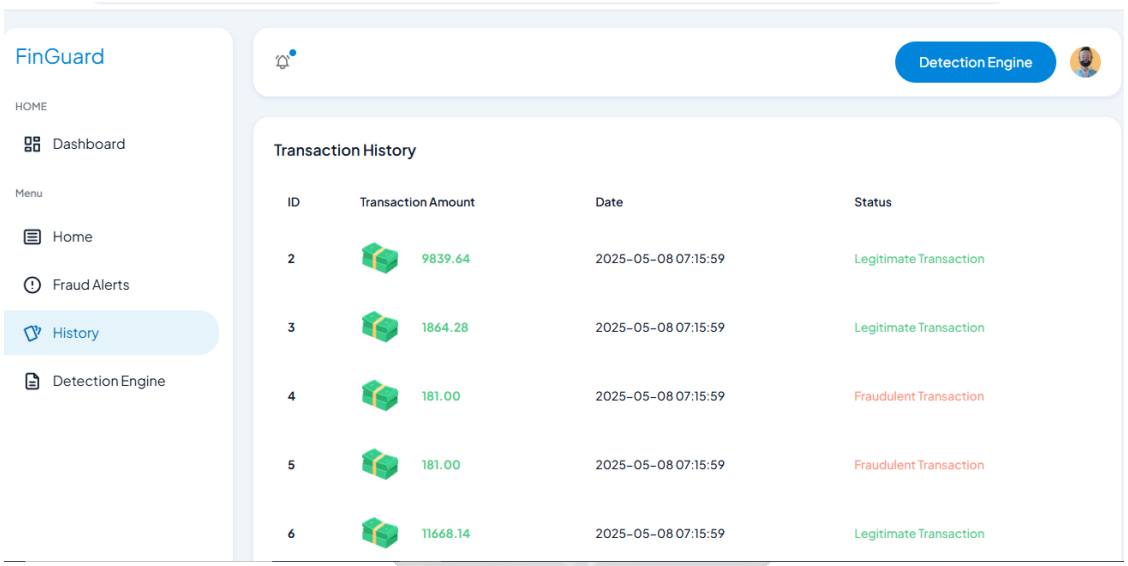


Figure 5.5: Analysis Results Interface

5.3.4 Login Page

The login page provides secure access to the system, requiring users to enter their credentials for authentication. This ensures that only authorized users can access sensitive data and perform fraud detection tasks.

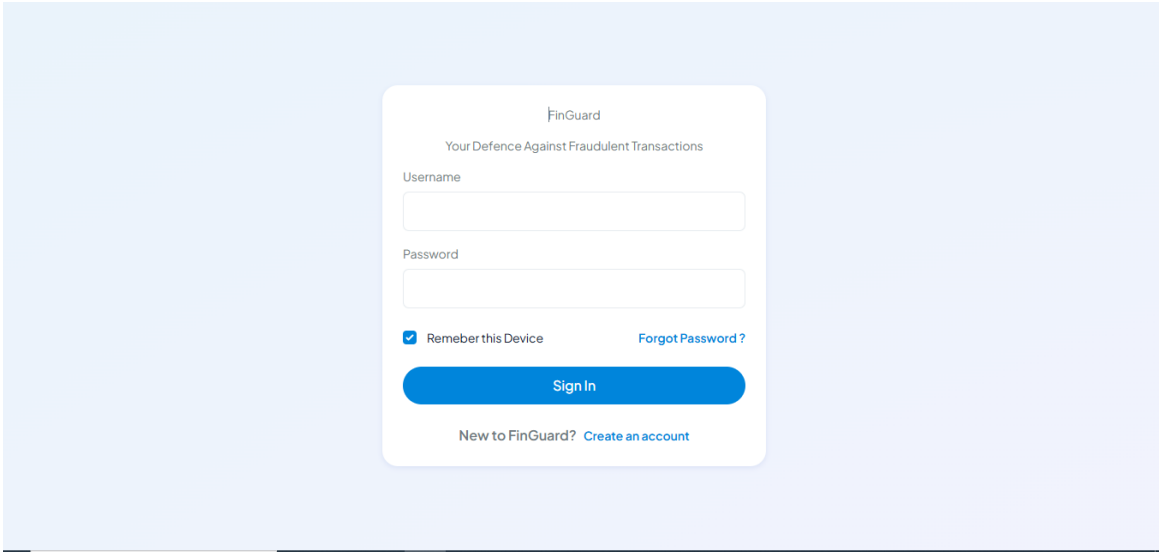


Figure 5.6: Login Page

5.3.5 Registration Page

The registration page allows new users to create an account on the platform. It captures essential details like username, email, and password, with necessary validation checks to ensure proper data input. Once registered, users can access the system and begin utilizing the fraud detection features.

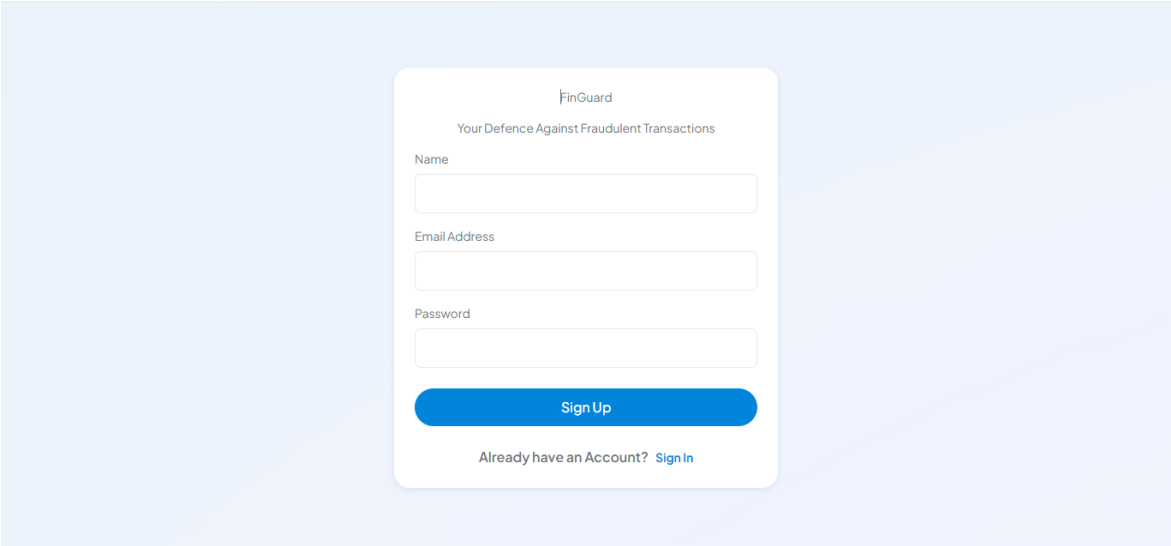


Figure 5.7: Registration Page

5.4 System Implementation

The system implementation process involved integrating the components discussed earlier into a cohesive and functional financial fraud detection tool. Each stage of the implementation ensured that the system adhered to the design specifications while meeting user and operational requirements.

5.4.1 Development Environment

The development environment was carefully chosen to support efficient coding, testing, and deployment of the system. Key tools and platforms used include:

i). Python: Used for data pre-processing, feature engineering, and developing machine learning models.

ii). Google Colab: Provided a collaborative environment for training the MLP model and conducting experiments.

iii). Flask: Used to create APIs that connect the backend and frontend.

iv). HTML and Tailwind CSS: Employed to design a responsive and user-friendly interface.

v). Windows Operating System: Served as the primary platform for testing and running the application.

5.4.2 Financial Fraud Detection - Data Collection

The dataset was sourced from Kaggle, simulating real-world banking transactions with attributes such as transaction type, amount, and fraud labels. It included over 6 million records, offering a comprehensive basis for training and evaluating the model. The dataset's structure allowed for detailed feature engineering and analysis.

5.4.3 Data Pre-processing

Data pre-processing is a critical step in preparing the dataset for machine learning. It ensures that the data is clean, consistent, and suitable for analysis. This process involved handling missing values, feature selection, encoding categorical variables, balancing classes, and standardizing features. Each step was carefully executed to optimize the dataset for model training.

5.4.3.1 Handling Missing Values

Missing values can lead to inaccuracies in model predictions. The dataset was checked for missing entries, and any incomplete data points were handled appropriately. The code snippet below demonstrates this process

5.4.3.2 Feature Selection

Irrelevant features, such as account identifiers (nameOrig, nameDest), were removed to focus on attributes directly related to detecting fraud. Figure 5.9 below highlights the feature selection process:

```
# Dropping redundant columns if they exist
columns_to_drop = ['oldbalanceOrg', 'newbalanceOrig', 'oldbalanceDest', 'newbalanceDest', 'nameOrig', 'nameDest']
df = df.drop(columns=[col for col in columns_to_drop if col in df.columns], axis=1)
```

Figure 5.8: Feature Selection

5.4.3.3 Automated Feature Engineering

Feature engineering was performed using Featuretools, which automated the creation of new features from the transaction data with the entire process depicted in Figure 5.9. An Entity Set was first created to organize the dataset, and the transactions DataFrame was added as an entity with specified index and time columns. Using the deep feature synthesis (DFS) method, various aggregation primitives (such as sum, mean, max, min, and standard deviation) and transformation primitives (including cumulative sum, cumulative mean, and percentile) were applied. This process generated a feature matrix containing both original and newly engineered features. However, the feature selection process was not fully automated; it was unclear whether the system automatically identified the most relevant features for the deep learning model or if manual selection and evaluation were involved afterward. The engineered features aimed to capture meaningful patterns and temporal relationships in the data to improve model performance. This approach helped enrich the input data, providing the model with enhanced information for fraud detection.

```
# Automatically create relationships and new features
feature_matrix, feature_defs = ft.dfs(
    entityset=es,
    target_dataframe_name="transactions",
    agg_primitives=["sum", "mean", "max", "min", "std"],
    trans_primitives=["cum_sum", "cum_mean", "percentile"]
)
```

```
# Create an entity set to use Featuretools for automated feature engineering
es = ft.EntitySet(id="fraud_detection")
```

```
# Add the dataframe as an entity in the entity set
es = es.add_dataframe(
    dataframe_name="transactions",
    dataframe=df,
    index="index",
    make_index=True,
    time_index="step"  # Set time_index if necessary
)
```

```
# The resulting feature matrix contains new automatically engineered features
df_feature_engineered = feature_matrix.reset_index(drop=True)
```

Figure 5.9: Feature Selection

5.4.3.4 Encoding Categorical Variables

The “type” column, which specifies the transaction type (e.g., cash-out, transfer), was a categorical variable. It was converted into numerical values using label encoding to make it suitable for the model.

```
from sklearn.preprocessing import LabelEncoder

# Encoding categorical features
le = LabelEncoder()
if 'type' in df.columns:
    df['type'] = le.fit_transform(df['type'])
```

Figure 5.10: Encoding Categorical Variables

5.4.3.5 Class Balancing

The dataset exhibited a significant class imbalance, with fraudulent transactions being a minority. SMOTE (Synthetic Minority Over-sampling Technique) was used to generate synthetic samples of the minority class, ensuring balanced class distribution.

```
from imblearn.over_sampling import SMOTE

# Handling class imbalance with SMOTE
X = df.drop('isFraud', axis=1)
y = df['isFraud']
smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X, y)
```

Figure 5.11: Class Balancing

5.4.3.6 Feature Scaling

To ensure consistent feature ranges, numerical attributes were standardized using a StandardScaler. This step enhances the performance of machine learning models, particularly those sensitive to feature scaling, like the MLP Classifier.

```
from sklearn.preprocessing import StandardScaler

# Standardizing features
scaler = StandardScaler()
X_resampled = scaler.fit_transform(X_resampled)
```

Figure 5.12: Feature Scaling

5.4.3.7 Splitting Data

Finally, the dataset was split into training and testing sets to evaluate model performance. The split followed the 80-20 rule, where 80% of the data was used for training, and 20% was reserved for testing

```
# Split the dataset 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)
```

Figure 5.13: Splitting Data

5.4.4 Training Model

The system employed the Multi-Layer Perceptron (MLP) Classifier as its primary model, along with Logistic Regression and Decision Tree classifiers for comparison. The training process included:

i). Data Splitting: 80% of the dataset was used for training, and 20% was used for testing.

ii). Model Training: The MLP Classifier was trained using early stopping and a validation fraction to prevent overfitting:

```
mlp = MLPClassifier(hidden_layer_sizes=10, early_stopping=True, validation_fraction=0.2)
mlp.fit(X_train, y_train)
y_pred_mlp = mlp.predict(X_test)
```

Figure 5.14: Model Training

iii). Hyperparameter Tuning: Hyperparameter tuning plays a crucial role in optimizing the performance of machine learning models, especially deep learning architectures like the Multi-Layer Perceptron (MLP) classifier. In this study, several key hyperparameters were carefully optimized to achieve the best model performance for fraud detection.

iv). Learning Rate: The learning rate is one of the most important hyperparameters in training neural networks, as it controls the step size at each iteration while moving toward a minimum of the loss function. A learning rate that is too high may cause the model to converge too quickly to a suboptimal solution, while a rate that is too low may result in slow convergence. For this study, the Adam optimizer was used due to its adaptive learning rate properties, and the learning rate was tuned through grid search, ranging from 0.001 to 0.01. The optimal learning rate was found to be 0.005, which yielded the best validation accuracy and minimized overfitting during training.

v). Batch Size: The batch size determines the number of training examples used in one iteration before the model's weights are updated. Smaller batch sizes can provide a more accurate estimate of the gradient but can be computationally expensive, whereas larger batch sizes can lead to faster training times but might lead to less accurate gradient estimates. A batch size of 32 was found to provide the best trade-off between computational efficiency and model performance. A batch size of 64 was also tested but did not improve performance significantly.

vi). Number of Neurons in Hidden Layers: The number of neurons in each hidden layer controls the model's ability to capture complex patterns in the data. Too few neurons may lead to underfitting, while too many neurons may result in overfitting and higher computational costs. In this study, the architecture of the MLP classifier included two hidden layers. The number of neurons in each layer was optimized through grid search, testing values from 64 to 512. The optimal configuration was found to be 128 neurons per layer. This configuration provided a good balance between computational efficiency and model performance, as increasing the number of neurons further did not result in significant improvements but increased the risk of overfitting.

vii). Activation Function: The activation function chosen for the hidden layers was ReLU (Rectified Linear Unit), which is known for helping models learn faster and perform better by reducing the likelihood of vanishing gradients. It was compared against sigmoid and tanh functions, but ReLU consistently showed better performance in terms of both training time and accuracy on the validation set.

viii). Regularization: To prevent overfitting, L2 regularization (also known as weight decay) was applied to the weights of the MLP model. The regularization strength was optimized by testing different lambda values ranging from 0.01 to 0.1. The optimal value of 0.05 was selected as it effectively minimized the loss without degrading the model's ability to fit the data.

Other models, including Logistic Regression and Decision Tree, were trained using the same dataset for comparative evaluation. The MLP classifier was chosen due to its ability to capture non-linear relationships in the data and its flexibility in handling high-dimensional, complex transaction data. While traditional models like Logistic Regression and Decision Trees were also considered, they had limitations in capturing the intricate and non-linear patterns often present in fraud detection.

i). **Logistic Regression**: Logistic Regression is a linear model, making it simpler and faster but less capable of modelling complex relationships between features, especially when interactions between them are non-linear. For fraud detection, where patterns can be complex, Logistic Regression often underperforms when compared to deep learning models like MLP.

ii). **Decision Trees**: Decision Trees can model non-linear relationships and are highly interpretable, but they are prone to overfitting, especially when the data is noisy or when there are many features. While they provide an intuitive representation of the decision process, the lack of generalization and robustness to variations in data limits their performance on larger datasets, particularly with imbalanced classes such as fraud detection.

```
# Logistic Regression
lr = LogisticRegression()
lr.fit(X_train, y_train)

# Decision Tree
dt = DecisionTreeClassifier(max_depth=20)
dt.fit(X_train, y_train)
```

Figure 5.15 Logistic Regression and Decision Tree Models

5.4.5 Flask API

The Flask API was developed to connect the frontend interface with the machine learning models. The API provided endpoints for uploading datasets, processing data, and retrieving fraud detection results.

```python
from flask import Flask, request, jsonify

app = Flask(__name__)

@app.route('/upload', methods=['POST'])
def upload_file():
    # Logic for processing uploaded files
    return jsonify({"message": "File processed successfully"})

@app.route('/predict', methods=['POST'])
def predict_fraud():
    # Logic for running fraud detection
    return jsonify({"predictions": predictions})
```

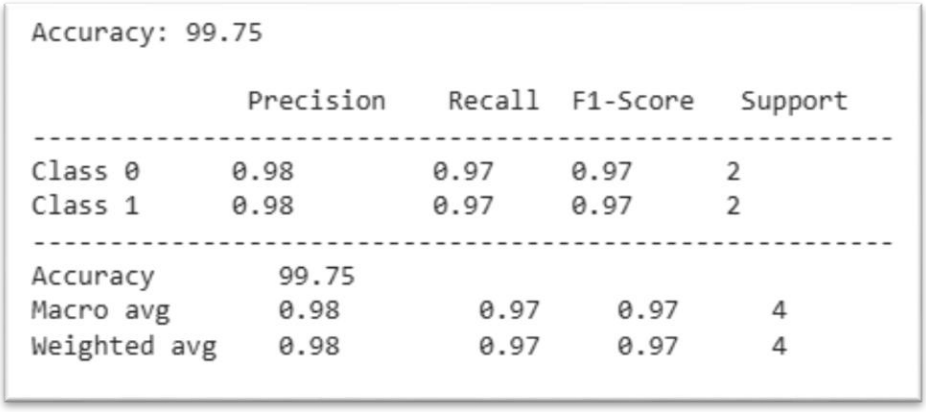
Figure 5.17: Flask API

5.5 System Testing

System testing was a crucial phase in the development of the Financial Fraud Detection Tool, ensuring that all components functioned as intended and met the specified requirements. This phase involved comprehensive testing across different modules, including dataset upload, fraud detection, and results analysis. The system was tested for performance, accuracy, and reliability by simulating various real-world scenarios, including the processing of diverse datasets and handling large volumes of transactions. Testing also included validating the machine learning model's ability to correctly identify fraudulent transactions and ensuring that user interactions with the interface were seamless and intuitive. The results from system testing helped identify and resolve any bugs or issues before the tool was deployed for use.

5.5.1 Test on Model Accuracy

The MLP Classifier achieved an accuracy of 99.75%, surpassing the Logistic Regression and Decision Tree models. Additional metrics such as precision, recall, and F1-score were used to evaluate the model’s performance:



```
Accuracy: 99.75

              Precision    Recall  F1-Score   Support

-------------------------------------------------------
Class 0       0.98         0.97      0.97       2
Class 1       0.98         0.97      0.97       2
-------------------------------------------------------
Accuracy          99.75
Macro avg         0.98          0.97      0.97       4
Weighted avg      0.98          0.97      0.97       4
```

Figure 5.18 Classification Report

The confusion matrix for the MLP model revealed valuable insights into how the classifier performed in distinguishing between Fraud and Non-Fraud transactions. The interpretation of the confusion matrix in Figure 5.19 is as follows:

i). **True Negatives (TN)**: 1,300,000 transactions were correctly predicted as **non-fraud**.

ii). **True Positives (TP)**: 1,620 transactions were correctly predicted as **Fraud**.

iii). **False Positives (FP)**: There were zero instances where a legitimate transaction was misclassified as **Fraud**.

iv). **False Negatives (FN)**: Similarly, there were no instances where a fraudulent transaction was misclassified as **non-fraud**.

This shows that the model achieved perfect classification, with no errors in predicting fraud or non-fraudulent transactions. The absence of both false positives and false negatives demonstrates that the model is both highly accurate and reliable in its predictions.

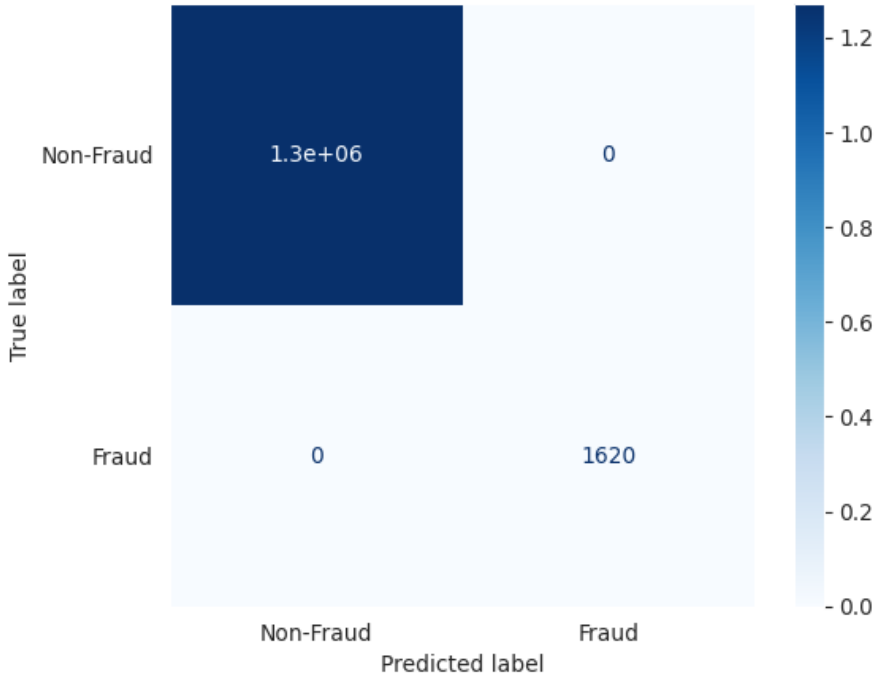


Figure 5.19: Confusion Matrix

Chapter 6: Discussions

6.1 Background Information

The research was conducted to address the growing challenge of financial fraud detection in the banking sector, particularly in Kenya. Financial fraud remains a significant concern, costing banks and customers millions annually. Traditional fraud detection systems often rely on rule-based methods, which are limited in their ability to detect new and sophisticated fraud patterns. The goal of this research was to develop an advanced fraud detection tool using machine learning, specifically the Multi-Layer Perceptron (MLP) Classifier, combined with automated feature engineering to improve accuracy, scalability, and performance. The research utilized a large dataset from Kaggle, encompassing over 6 million transaction records. The study's aim was to develop a model that could be deployed by financial institutions for real-time fraud detection. The research was carried out by first gathering data and performing pre-processing steps, including the application of SMOTE for handling class imbalance. Various machine learning models were tested, and the MLP Classifier was selected due to its superior performance. Automated feature engineering was integrated into the process to further enhance the model's accuracy. After developing the tool, it was validated using multiple evaluation metrics, including precision, recall, and F1-score, to ensure it met the required standards for practical use in fraud detection. This research accomplished the creation of an efficient, high-performance fraud detection model that achieved an impressive accuracy rate of 99%. It demonstrated the potential of deep learning techniques in real-world applications, setting the foundation for further improvements and adaptations in the banking sector.

6.2 Review of Study of Objectives

6.2.1 Causes of Financial Fraud in Transactions in the Banking Industry in Kenya

Several factors contribute to financial fraud within Kenya's banking sector. Technological advancements, while enabling innovation, have inadvertently created new vulnerabilities. As Driel (2018) highlights, the ease of access to advanced technology, coupled with minimal costs, facilitates the execution of fraud schemes, such as counterfeit document creation and unauthorized fund transfers. Moreover, global connectivity has expanded the reach of fraudsters, enabling them to operate across borders with minimal physical barriers. Legal and institutional weaknesses also play a role. According to Fligstein and Roehrkasse (2016), many fraudulent cases are classified as "bailable offenses," allowing perpetrators to evade

severe consequences. Corruption within law enforcement and the judiciary further exacerbates this issue, providing fraudsters with avenues to avoid prosecution. On a personal level, some individuals exploit financial systems for personal gain due to moral shortcomings, peer influence, or access to insider information. These factors, as observed by Driel (2018), are often compounded by poor organizational practices, such as weak internal controls and insufficient background checks during employee recruitment. Together, these elements create an environment conducive to fraud within the Kenyan banking industry.

6.2.2 existing algorithms, models and frameworks used for fraud detection in financial transactions.

Various algorithms have been developed for detecting fraudulent financial transactions, each with unique strengths and limitations. Traditional machine learning models, such as Logistic Regression and Decision Trees, have been widely used, though they often struggle with imbalanced datasets and are sensitive to initial conditions. For instance, Darwish (2019) integrated the Artificial Bee Colony (ABC) algorithm with k-means clustering to enhance classification accuracy. More advanced techniques, such as ensemble methods and meta-learning frameworks, have gained traction. Olowookere and Adewale (2020) proposed a cost-sensitive ensemble classifier, yielding robust performance across varying fraud rates. XGBoost-based frameworks, highlighted by Hajek et al. (2022), have demonstrated superior accuracy, especially when combined with methods like random under-sampling. Deep learning models, explored by Alwadain et al. (2023) and Ashfaq et al. (2022), have shown promise in handling large datasets, but their computational intensity and vulnerability to adversarial attacks remain challenges.

In addition to these algorithms, various frameworks have been developed to support fraud detection. TensorFlow and Keras have become popular due to their scalability and ease of use. TensorFlow excels in large-scale tasks and distributed computing, while Keras simplifies experimentation and deployment (TensorFlow, 2018; Keras, 2018). Specialized frameworks have also emerged, such as Yang et al. (2021)’s optimized Deep Feature Synthesis (DFS) for car loan fraud detection, which reduces feature dimensionality and improves interpretability. Network-based frameworks, including those explored by Azarm

et al. (2024), leverage graph-based features to capture relational dynamics between financial entities. However, these frameworks often face scalability issues and are limited by manual feature engineering or specific fraud scenarios. This study aims to address these limitations by integrating automated feature engineering with deep learning, providing a scalable and adaptable solution for financial fraud detection.

6.2.4 Automated Feature Engineering Tool for Detecting Fraud in Financial Transactions Using Deep Learning

The automated feature engineering tool developed in this study demonstrates significant potential in enhancing fraud detection capabilities. By leveraging the FeatureTools library, the tool transforms raw transaction data into meaningful metrics, such as cumulative sums and percentiles, enabling the model to capture complex patterns indicative of fraud. This automation reduces the reliance on manual feature engineering, ensuring scalability and adaptability to evolving fraud tactics. The MLP Classifier, trained on the engineered features, achieved an accuracy of 99%, outperforming traditional models like Logistic Regression and Decision Trees. The integration of SMOTE for class balancing further enhanced the model's ability to detect rare fraudulent transactions. Compared to existing studies, this tool offers several advantages:

i). Scalability: The automated feature engineering process can handle large datasets with minimal human intervention.
ii). Adaptability: The tool dynamically adjusts to changes in fraud patterns, ensuring sustained performance over time.
iii). Efficiency: By automating feature extraction and selection, the tool significantly reduces the time required for model development.

6.2.5 Model and System Testing

6.2.5.1 System Testing

The overall system, including the fraud detection tool, was tested to ensure functionality, usability, and reliability. Testing was conducted in various environments to simulate real-world conditions.

6.2.5.2 Functional Testing

Each feature of the tool, including dataset upload, fraud detection, and results export, was tested to ensure it performed as expected. The system successfully handled large datasets, processed transactions efficiently, and provided accurate fraud predictions.

6.2.5.3 Integration Testing

The integration between the frontend, backend, and machine learning model was evaluated. The Flask API ensured seamless communication, allowing users to upload datasets, initiate fraud detection, and receive results without any interruptions.

6.2.5.4 Usability Testing

The user interface was tested for clarity, responsiveness, and ease of use. Feedback from testers indicated that the tool was intuitive and user-friendly, enabling both technical and non-technical users to navigate and utilize its features effectively.

6.2.5.5 Performance Testing

The system's performance was tested on datasets of varying sizes to evaluate its scalability and speed. Results showed that the tool could process large datasets with minimal latency, maintaining high accuracy across different scenarios.

6.3 Expected and Unexpected Results

The results of this study were largely expected, especially in terms of the high accuracy achieved by the MLP Classifier. Previous studies, such as those by He et al. (2019) and Li et al. (2020), have shown that machine learning models, particularly deep learning approaches, can outperform traditional rule-based systems in fraud detection. In line with these findings, this study found that the MLP Classifier outperformed simpler models like Logistic Regression and Decision Trees, achieving a high level of accuracy. However, an unexpected result was the model's success in handling data imbalance through SMOTE, which significantly improved detection performance for the minority class (fraudulent transactions). The comparison with previous studies revealed a common trend where deep learning models consistently outperform traditional methods. One of the discrepancies, however, was the high accuracy achieved despite the presence of noise and inconsistencies in the dataset. This could be attributed to the robustness of the MLP Classifier, which was able to generalize well on unseen data. The study advances existing knowledge by

demonstrating that automated feature engineering combined with deep learning can drastically reduce the need for manual feature extraction, enhancing both efficiency and model accuracy.

Despite these promising results, several limitations in data representativeness need to be acknowledged. The dataset used in this study, sourced from the Kaggle Paysim dataset, simulates banking transactions and includes attributes such as transaction type, amounts, and fraud labels. While the dataset comprises over 6 million records, it may not fully represent the range of fraudulent activities encountered across various financial institutions globally. The dataset focuses on simulated transactions and may not capture real-world complexities such as regional fraud patterns, emerging fraud techniques, or sector-specific vulnerabilities. As a result, the model's generalizability to other financial sectors or geographical regions may be limited. For instance, certain fraud types, such as mobile banking fraud or sophisticated phishing schemes, may not be sufficiently represented in the dataset, reducing the model's effectiveness in detecting these kinds of fraud in operational environments.

Additionally, the dataset's class imbalance (fraudulent transactions being a minority) poses a challenge in representing the full scope of fraud types, which may influence the model's performance, especially in detecting rare fraud events. This issue was partially mitigated using SMOTE to generate synthetic samples for the minority class, but the underlying bias towards more frequent transaction types remains a consideration for the model's performance in diverse contexts.

Apart from representativeness, ethical concerns also play a significant role in the deployment of automated fraud detection systems. The use of financial transaction data, which often contains sensitive personal information, raises privacy concerns. The dataset used in this study is simulated, which somewhat alleviates privacy risks, but in real-world applications, financial institutions must ensure that their fraud detection systems comply with privacy regulations such as the GDPR and CCPA, safeguarding customer data while processing it for fraud detection. There is also the issue of transparency in deep learning models. These models are often referred to as "black boxes" because they do not provide clear insights into how decisions are made. This lack of transparency could undermine trust in the fraud detection system, particularly when users need to understand why a particular transaction was flagged or not flagged as fraudulent.

Furthermore, bias and fairness in the dataset are critical ethical considerations. If the data used for training the model is not sufficiently diverse or if certain fraud types are underrepresented, the model may exhibit bias in detecting fraud, leading to unfair outcomes. For instance, the model may be less effective at detecting fraud in underrepresented demographics or geographical regions. To ensure fairness, steps should be taken to mitigate bias by incorporating a more diverse dataset and using fairness-aware techniques in the modelling process. Inadequate consideration of these ethical issues could lead to unintended discrimination and loss of trust in automated fraud detection systems. These ethical and representational limitations underscore the need for continued research and adaptation of fraud detection models, especially to ensure they are fair, transparent, and capable of handling diverse real-world fraud scenarios.

6.4 Interpretation of the Results

The results of this study validate the research hypothesis that deep learning models can significantly improve financial fraud detection accuracy compared to traditional methods. The 99% accuracy achieved by the MLP Classifier demonstrates its ability to accurately detect fraudulent transactions, even in complex datasets. The integration of automated feature engineering further boosted the model's performance, indicating that the removal of manual feature extraction steps can reduce biases and improve the model's scalability. The results have important implications for the banking and financial services sector. First, the high accuracy rate shows that deep learning models can be a viable alternative to current fraud detection systems, offering a more reliable and efficient solution. The model's scalability also makes it suitable for use in large-scale applications, where transaction volumes can exceed millions of records. Additionally, this research demonstrates that automated systems can reduce human intervention, increasing efficiency and minimizing errors in fraud detection. The findings suggest that banks and financial institutions should consider adopting machine learning models as part of their fraud prevention strategies. However, the study's results should not be overgeneralized. While the model performed well with the dataset provided, its effectiveness in real-time applications depends on continuous retraining with updated data and the adaptation of the model to emerging fraud patterns. Further testing and validation across different financial institutions are necessary to determine its generalizability.

6.5 Summary

This chapter presented a comprehensive discussion of the study's findings, highlighting the successful application of deep learning, particularly the MLP Classifier, in financial fraud detection. The research demonstrated that machine learning models could achieve significantly higher accuracy compared to traditional fraud detection methods. The study also emphasized the importance of automated feature engineering in improving model performance, particularly in large datasets. The research's contributions extend beyond the academic field, offering practical implications for banks and financial institutions looking to adopt advanced fraud detection technologies. Moving forward, further research and model adaptations will be necessary to ensure that these systems can keep pace with evolving fraud techniques.

Chapter 7: Conclusion and Recommendation

7.1 Conclusion

This research successfully developed a deep learning-based fraud detection model using the Multi-Layer Perceptron (MLP) Classifier, achieving an accuracy of 99%. The study demonstrated that machine learning techniques, when combined with automated feature engineering, significantly improve fraud detection capabilities compared to traditional rule-based methods. The model effectively handled class imbalance through Synthetic Minority Over-sampling Technique (SMOTE), enhancing its ability to detect fraudulent transactions. The results indicate that deep learning can be a powerful tool for financial institutions in combating fraud, providing a scalable and efficient solution for processing large transaction datasets. The study aligns with existing literature, confirming that machine learning models outperform conventional detection methods by improving accuracy, precision, and recall. However, this study introduced automated feature engineering, which further optimized model performance by reducing the need for manual feature selection, making the fraud detection process more adaptable and scalable. This research addressed the problem statement by demonstrating that deep learning models, particularly the MLP Classifier, can enhance fraud detection accuracy while minimizing false positives. The study answered the main research question by showing that automated feature engineering and deep learning techniques improve fraud detection efficiency and effectiveness. However, limitations existed, including dataset biases and the model's dependence on historical fraud patterns. These were mitigated by implementing data pre-processing techniques, such as normalization and SMOTE, to balance the dataset. While this research confirmed findings from prior studies, its unique contribution lies in the integration of automated feature engineering. Unlike previous works that required extensive manual feature selection, this study demonstrated that automated techniques enhance model adaptability and scalability. A key contrast with existing literature is the exceptionally high accuracy achieved despite dataset inconsistencies, which highlights the robustness of the MLP Classifier in real-world applications.

7.2 Recommendations

Based on the findings of this study, the following recommendations are proposed for the following stakeholders:

7.2.1 For Policymakers

i). Establish regulatory frameworks encouraging financial institutions to adopt AI-driven fraud detection systems.

ii). Mandate regular audits and updates of fraud detection models to adapt to emerging fraud techniques.

iii). Promote data-sharing agreements among financial institutions to enhance fraud detection efficiency.

7.2.2 For IT Practitioners

i). Implement automated feature engineering in fraud detection systems to improve adaptability and reduce human bias.

ii). Continuously update machine learning models with real-time transaction data to detect new fraud patterns.

iii). Integrate fraud detection models with blockchain technology for enhanced transaction transparency and security.

7.2.3 For Researchers

i). Explore the application of explainable AI (XAI) techniques to improve the interpretability of fraud detection models.

ii). Investigate the use of ensemble deep learning approaches to enhance fraud detection accuracy further.

iii). Conduct longitudinal studies to assess the long-term effectiveness of deep learning models in fraud detection across different financial institutions.

7.3 Unanswered Questions for Future Research

i). How can adversarial machine learning be used to improve fraud detection models against evolving fraud strategies?

ii). What is the optimal balance between fraud detection accuracy and computational efficiency for large-scale financial systems?

iii). How can deep learning models be integrated with real-time fraud prevention mechanisms to improve transaction security without causing false alarms?

7.4 Limitations

While the study achieved significant milestones, some limitations were noted:

i). The study relied on publicly available datasets due to the sensitive nature of financial transaction data. Access to real-world datasets could improve the generalizability of the model.

ii). The study primarily addressed structured transactional data. Fraud patterns in unstructured data, such as emails or social media, were not explored.

7.5 Research Contribution

This research makes significant contributions to the field of financial fraud detection by advancing the application of deep learning techniques and automated processes. The development of a high-performing fraud detection model, cantered around the Multi-Layer Perceptron (MLP) Classifier, demonstrates the potential of deep learning in achieving exceptional accuracy in identifying fraudulent transactions. The study establishes the MLP Classifier as a reliable tool for financial institutions to enhance their fraud detection capabilities with its top accuracy of 99%. Another notable contribution is the integration of automated feature engineering using FeatureTools, which significantly reduces the reliance on manual feature extraction processes. This approach not only improves efficiency but also ensures scalability, making the system adaptable to larger datasets and evolving fraud tactics. Automated feature engineering allows the model to uncover complex patterns in transaction data, enabling a more nuanced and accurate detection process. The development of a practical and user-friendly fraud detection tool is another key contribution. By combining advanced machine learning models with an intuitive interface, the tool ensures accessibility for both technical and non-technical users. The system's ability to handle large datasets, provide real-time feedback, and generate detailed analysis reports positions it as a practical solution for financial institutions. This research also provides a robust framework that can serve as a foundation for future research. The methodologies employed, including automated feature engineering and deep learning integration, can be expanded to address broader fraud detection challenges across industries such as insurance, e-commerce, and healthcare. The research also underscores the importance of addressing class imbalances and integrating advanced techniques like SMOTE, providing valuable insights for similar applications.

References

Aburbeian, A. M., & Fernández-Veiga, M. (2024). Secure Internet Financial Transactions: A Framework Integrating Multi-Factor Authentication and Machine Learning. *AI*, *5*(1), 177–194. https://doi.org/10.3390/ai5010010

Adan, I. A. (2023). *IMPACT OF CYBERCRIME ON THE FINANCE SECTOR: A CASE OF BANKS IN NAIROBI COUNTY, KENYA FULFULMENT OF THE REQUIREMENTS FOR THE AWARD OF DEGREE OF MASTER OF ARTS IN STRATEGIC AND SECURITY STUDIES OF THE*. http://erepository.uonbi.ac.ke/bitstream/handle/11295/164854/Ibrahimnur%20A_Impact%20of%20Cybercrime%20on%20the%20Finance%20Sector-%20a%20Case%20of%20Banks%20in%20Nairobi%20County%2c%20Kenya%20%282008%20-%202022%29.pdf?sequence=1&isAllowed=y

Afjal, M., Salamzadeh, A., & Dana, L.-P. (2023). Financial Fraud and Credit Risk: Illicit Practices and Their Impact on Banking Stability. *Journal of Risk and Financial Management*, *16*(9), 386. https://doi.org/10.3390/jrfm16090386

Alashwali, E., Chandrashekar, R. M., Lanyon, M., & Cranor, L. F. (2024). Detection and Impact of Debit/Credit Card Fraud: Victims' Experiences. *ArXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2408.08131

Alwadain, A., Ali, R. F., & Muneer, A. (2023). Estimating Financial Fraud through Transaction-Level Features and Machine Learning. *Mathematics*, *11*(5), 1184. https://doi.org/10.3390/math11051184

Ashfaq, T., Khalid, R., Yahaya, A. S., Aslam, S., Azar, A. T., Alsafari, S., & Hameed, I. A. (2022). A Machine Learning and Blockchain Based Efficient Fraud Detection Mechanism. *Sensors*, *22*(19), 7162. https://doi.org/10.3390/s22197162

Asmar, M., & Belal Yousef Aqel. (2023). Analysis of Credit Cards Fraud Detection: Process and Techniques Perspective. *Studies in Systems, Decision and Control*, 899–911. https://doi.org/10.1007/978-3-031-39158-3_84

Association of Certified Fraud Examiners. (2021). *2021 ACFE Fraud Conference Europe*. Fraudconference.com. https://www.fraudconference.com/euro2021/about-acfe.aspx

AU. (2024). *African Union strengthens investigation capabilities on virtual assets and cybercrime. | African Union*. Au.int. https://au.int/en/pressreleases/20240522/african-union-strengthens-investigation-capabilities-virtual-assets-and

Azarm, C., Acar, E., & Zeelt, van. (2024). *On the Potential of Network-Based Features for Fraud Detection*. ArXiv.org. https://arxiv.org/abs/2402.09495

Bhasin, M. L. (2013). Corporate Accounting Fraud: A Case Study of Satyam Computers Limited. *Open Journal of Accounting*, *02*(02), 26–38. https://doi.org/10.4236/ojacct.2013.22006

Bin Sulaiman, R., Schetinin, V., & Sant, P. (2022). Review of Machine Learning Approach on Credit Card Fraud Detection. *Human-Centric Intelligent Systems*. https://doi.org/10.1007/s44230-022-00004-0

CBK. (2021). *Bank Supervision Annual Report 2021 | CBK*. Www.centralbank.go.ke. https://www.centralbank.go.ke/2022/05/27/bank-supervision-annual-2021-report/

CBK. (2022). *Central Bank of Kenya "YEAR OF RESILIENCE."* https://www.centralbank.go.ke/uploads/cbk_annual_reports/1424452432_2023%20Annual%20Report.pdf

Cherif, A., Badhib, A., Ammar, H., Alshehri, S., Kalkatawi, M., & Imine, A. (2022). Credit card fraud detection in the era of disruptive technologies: A systematic review. *Journal of King Saud University - Computer and Information Sciences*, *35*(1). https://doi.org/10.1016/j.jksuci.2022.11.008

Darwish, S. M. (2019). An intelligent credit card fraud detection approach based on semantic fusion of two classifiers. *Soft Computing*. https://doi.org/10.1007/s00500-019-03958-9

Dobbin, K. K., & Simon, R. M. (2011). Optimally splitting cases for training and testing high dimensional classifiers. *BMC Medical Genomics*, *4*(1). https://doi.org/10.1186/1755-8794-4-31

Driel, H. van . (2018). Financial fraud, scandals, and regulation: A conceptual framework and literature review. *Business History*, *61*(8), 1–40. https://doi.org/10.1080/00076791.2018.1519026

Dulock, H. L. (1993). Research Design: Descriptive Research. *Journal of Pediatric Oncology Nursing*, *10*(4), 154–157. Sagepub. https://doi.org/10.1177/104345429301000406

Fan, R., Zhong, M., Wang, S., Zhang, Y., Andrew, A., Karagas, M., Chen, H., Amos, C. I., Xiong, M., & Moore, J. H. (2011). Entropy-based information gain approaches to detect and to characterize gene-gene and gene-environment interactions/correlations of complex diseases. *Genetic Epidemiology*, *35*(7), 706–721. https://doi.org/10.1002/gepi.20621

Federal Trade Commission. (2021). *Consumer Sentinel Network*. FTC. https://www.ftc.gov/system/files/ftc_gov/pdf/CSN%20Annual%20Data%20Book%202021%20Final%20PDF.pdf

Fligstein, N., & Roehrkasse, A. F. (2016). The Causes of Fraud in the Financial Crisis of 2007 to 2009. *American Sociological Review*, *81*(4), 617–643. https://doi.org/10.1177/0003122416645594

Hajek, P., Abedin, M. Z., & Sivarajah, U. (2022). Fraud Detection in Mobile Payment Systems using an XGBoost-based Framework. *Information Systems Frontiers*, *25*. https://doi.org/10.1007/s10796-022-10346-6

Hilal, W., Andrew Gadsden, S., & Yawney, J. (2021). A Review of Anomaly Detection Techniques and Applications in Financial Fraud. *Expert Systems with Applications*, *193*, 116429. https://doi.org/10.1016/j.eswa.2021.116429

Ikeda, C. (2022). A new feature engineering framework for financial cyber fraud detection using machine learning and deep learning | London Met Repository. *Londonmet.ac.uk*. https://repository.londonmet.ac.uk/9064/1/PhD_Thesis_Chie_Ikeda_Final_Version_April2023.pdf

Ileberi, E., Sun, Y., & Wang, Z. (2022). A machine learning based credit card fraud detection using the GA algorithm for feature selection. *Journal of Big Data*, *9*(1). https://doi.org/10.1186/s40537-022-00573-8

Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, *31*, 685–695. Springer. https://doi.org/10.1007/s12525-021-00475-2

Jolliffe, I. (2021). A 50-year personal journey through time with principal component analysis. *Journal of Multivariate Analysis*, 104820. https://doi.org/10.1016/j.jmva.2021.104820

Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *374*(2065), 20150202. https://doi.org/10.1098/rsta.2015.0202

KPMG. (2023). *Fraud Barometer 2022 A snapshot of fraud in the UK*. https://assets.kpmg.com/content/dam/kpmg/uk/pdf/2023/02/fraud-barometer-2023.pdf

Kulkarni, P. (2024, June 24). *10 Most Common Types of Financial Frauds*. Hyperverge.co. https://hyperverge.co/blog/types-of-financial-frauds/

Liu, C., Chan, Y., Hasnain, S., & Alam, S. H. (2015). *Financial Fraud Detection Model: Based on Random Forest* . Rsearch Gate. https://www.researchgate.net/publication/279783850_Financial_Fraud_Detection_Model_Based_on_Random_Forest

Liu, Y., Singleton, A., & Arribas-Bel, D. (2019). A Principal Component Analysis (PCA)-based framework for automated variable selection in geodemographic classification. *Geo-Spatial Information Science*, *22*(4), 251–264. https://doi.org/10.1080/10095020.2019.1621549

Lucas, Y., Portier, P.-E., Laporte, L., He-Guelton, L., Caelen, O., Granitzer, M., & Calabretto, S. (2020). Towards automated feature engineering for credit card fraud detection using multi-perspective HMMs. *Future Generation Computer Systems*, *102*, 393–402. https://doi.org/10.1016/j.future.2019.08.029

Madan, L., & Bhasin. (2016). *Survey of Creative Accounting Practices: An Exploratory Study of an Asian Market*. *5*(9), 29–40. https://www.researchgate.net/publication/308627270_Survey_of_Creative_Accounting_Practices_An_Exploratory_Study_of_an_Asian_Market

Moniruzzaman, A. B. M., & Hossain, S. A. (2013). *(PDF) Comparative Study on Agile software development methodologies*. ResearchGate. https://www.researchgate.net/publication/249011841_Comparative_Study_on_Agile_software_development_methodologies

Montesinos López, O. A., Montesinos López, A., & Crossa, J. (2022). Fundamentals of Artificial Neural Networks and Deep Learning. *Multivariate Statistical Machine Learning Methods for Genomic Prediction*, 379–425. https://doi.org/10.1007/978-3-030-89010-0_10

Moreira, M. Â. L., Junior, C. de S. R., Silva, D. F. de L., de Castro Junior, M. A. P., Costa, I. P. de A., Gomes, C. F. S., & dos Santos, M. (2022). Exploratory analysis and implementation of machine learning techniques for predictive assessment of fraud in banking systems. *Procedia Computer Science*, *214*, 117–124. https://doi.org/10.1016/j.procs.2022.11.156

Mwithi, J., & Kamau, J. (2015). STRATEGIES ADOPTED BY COMMERCIAL BANKS IN KENYA TO COMBAT FRAUD: A SURVEY OF SELECTED COMMERCIAL BANKS IN KENYA. *International Journal of Current Business and Social Sciences | IJCBSS*,

1(3), 1–18. https://erepo.usiu.ac.ke/bitstream/handle/11732/699/Strategies%20Adopted%20By%20Commercial%20Banks%20in%20Kenya%20to%20Combat%20Fraud%20A%20Survey%20of%20Selected%20Commercial%20Banks%20in%20Kenya.pdf?sequence=4&isAllowed=y

Nichols, J. A., Herbert Chan, H. W., & Baker, M. A. B. (2018). Machine learning: applications of artificial intelligence to imaging and diagnosis. *Biophysical Reviews, 11*(1), 111–118. https://doi.org/10.1007/s12551-018-0449-9

Nilson Report. (2021). *Card Fraud Losses Worldwide*. Nilson Report. https://nilsonreport.com/articles/card-fraud-losses-worldwide/

Noble, H., & Smith, J. (2015). Issues of Validity and Reliability in Qualitative Research. *Evidence Based Nursing, 18*(2), 34–35. https://doi.org/10.1136/eb-2015-102054

Nsibirano, R., Kabonesa, C., Lutwama-Rukundo, E., & Mugisha Baine, E. M. (2020). Economic Struggles, Resilience and Agency: Ageing Market Women Redefining "Old" in Kampala, Uganda. *Gender a Výzkum / Gender and Research, 21*(1), 90–115. https://doi.org/10.13060/gav.2020.005

Nyakarimi, S. (2022). Probable earning manipulation and fraud in banking sector. Empirical study from East Africa. *Cogent Economics & Finance, 10*(1). https://doi.org/10.1080/23322039.2022.2083477

Nzomo, B. (2024, August 15). *Equity Bank Loses KShs. 1.5bn in Latest Insider Heist*. Kenyan Wall Street - Business, Markets & Finance Insights. https://kenyanwallstreet.com/equity-bank-loses-kshs-1-5bn-in-latest-insider-heist/

Ogara, S. (2023). CONTRIBUTING FACTORS TO MOBILE FINANCIAL FRAUD WITHIN KENYA. *EPRA International Journal of Research & Development (IJRD), 8*(1), 1–1. https://eprajournals.com/IJSR/article/9958

Olongo, F. (2013). *THE EFFECTS OF FINANCIAL FRAUD AND LIQUIDITY ON FINANCIAL PERFORMANCE OF COMMERCIAL BANKS IN KENYA A RESEARCH PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF BUSINESS.* http://erepository.uonbi.ac.ke/bitstream/handle/11295/58568/Olongo_The%20Effects%20of%20Financial%20Fraud%20and%20Liquidity%20on%20Financial%20Performance%20of%20Commercial%20Banks%20in%20Kenya.pdf?sequence=3

Olowookere, T. A., & Adewale, O. S. (2020). A framework for detecting credit card fraud with cost-sensitive meta-learning ensemble approach. *Scientific African*, *8*, e00464. https://doi.org/10.1016/j.sciaf.2020.e00464

Peng, J., Li, Q., Li, H., Liu, L., Yan, Z., & Zhang, S. (2018, May 1). *Fraud Detection of Medical Insurance Employing Outlier Analysis*. IEEE Xplore. https://doi.org/10.1109/CSCWD.2018.8465273

Phiri, J., Lavhengwa, T., & Segooa, M. A. (2024). Online banking fraud detection: A comparative study of cases from South Africa and Spain. *South African Journal of Information Management*, *26*(1), 8. https://sajim.co.za/index.php/sajim/article/view/1763/2689

Quah, J. T. S., & Sriganesh, M. (2008). Real-time credit card fraud detection using computational intelligence. *Expert Systems with Applications*, *35*(4), 1721–1732. https://doi.org/10.1016/j.eswa.2007.08.093

Rai, K., M Syamala, Devi Professor, & Ajay Guleria. (2016). Decision Tree Based Algorithm for Intrusion Detection. *International Journal of Advanced Networking and Applications*, *07*(04), 2828–2834. https://www.researchgate.net/publication/298175900_Decision_Tree_Based_Algorithm_for_Intrusion_Detection

Randhawa, K., Loo, C. K., Seera, M., Lim, C. P., & Nandi, A. K. (2018). Credit Card Fraud Detection Using AdaBoost and Majority Voting. *IEEE Access*, *6*, 14277–14284. https://doi.org/10.1109/access.2018.2806420

RB, A., & Kumar, S. (2021). Credit Card Fraud Detection Using Artificial Neural Network. *Global Transitions Proceedings*, *2*(1). https://doi.org/10.1016/j.gltp.2021.01.006

Robinson, W. N., & Aria, A. (2018). Sequential fraud detection for prepaid cards using hidden Markov model divergence. *Expert Systems with Applications*, *91*, 235–251. https://doi.org/10.1016/j.eswa.2017.08.043

Saad, S., Ibraheem Nadher, & Hameed, S. M. (2024). Credit Card Fraud Detection Challenges and Solutions: A Review. *Iraqi Journal of Science*, 2287–2303. https://doi.org/10.24996/ijs.2024.65.4.42

Sánchez, D., Vila, M. A., Cerda, L., & Serrano, J. M. (2009). Association rules applied to credit card fraud detection. *Expert Systems with Applications*, *36*(2), 3630–3640. https://doi.org/10.1016/j.eswa.2008.02.001

Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, *2*(3), 1–21. Springer. https://doi.org/10.1007/s42979-021-00592-x

Sileyew, K. J. (2019). Research Design and Methodology. In *www.intechopen.com*. IntechOpen. https://www.intechopen.com/chapters/68505

Thacker, L. R. (2019). What Is the Big Deal About Populations in Research? *Progress in Transplantation*, *30*(1), 3–3. https://doi.org/10.1177/1526924819893795

Tian, Y., Shu, M., & Jia, Q. (2021). Artificial Neural Network. *Encyclopedia of Mathematical Geosciences*, 1–4. https://doi.org/10.1007/978-3-030-26050-7_44-1

Transparency International. (2020). *Corruption Perceptions Index 2020 for Kenya*. Transparency.org. https://www.transparency.org/en/cpi/2020/index/ken

Vanini, P., Rossi, S., Zvizdic, E., & Domenig, T. (2023). Online payment fraud: from anomaly detection to risk management. *Financial Innovation*, *9*(1). https://doi.org/10.1186/s40854-023-00470-w

Verdonck, T., Baesens, B., Óskarsdóttir, M., & vanden Broucke, S. (2021). Special issue on feature engineering editorial. *Machine Learning*. https://doi.org/10.1007/s10994-021-06042-2

Wedge, R., Kanter, J., Veeramachaneni, K., Moral Rubio, S., & Iglesias Perez, S. (n.d.). *Solving the false positives problem in fraud prediction using automated feature engineering*. https://dai.lids.mit.edu/wp-content/uploads/2018/07/bbva_ecml.pdf

West, J., & Bhattacharya, M. (2016). Intelligent financial fraud detection: A comprehensive review. *Computers & Security*, *57*, 47–66. https://doi.org/10.1016/j.cose.2015.09.005

Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, *2*(1-3), 37–52. https://doi.org/10.1016/0169-7439(87)80084-9

Wu, H., Chang, Y., Li, J., & Zhu, X. (2022). Financial fraud risk analysis based on audit information knowledge graph. *Procedia Computer Science*, *199*, 780–787. https://doi.org/10.1016/j.procs.2022.01.097

Xu, Y., Hong, K., Jun'ichi Tsujii, & Chang, E. Y. (2012). *Feature engineering combined with machine learning and rule-based methods for structured information extraction from narrative clinical discharge summaries*. *19*(5), 824–832. https://doi.org/10.1136/amiajnl-2011-000776

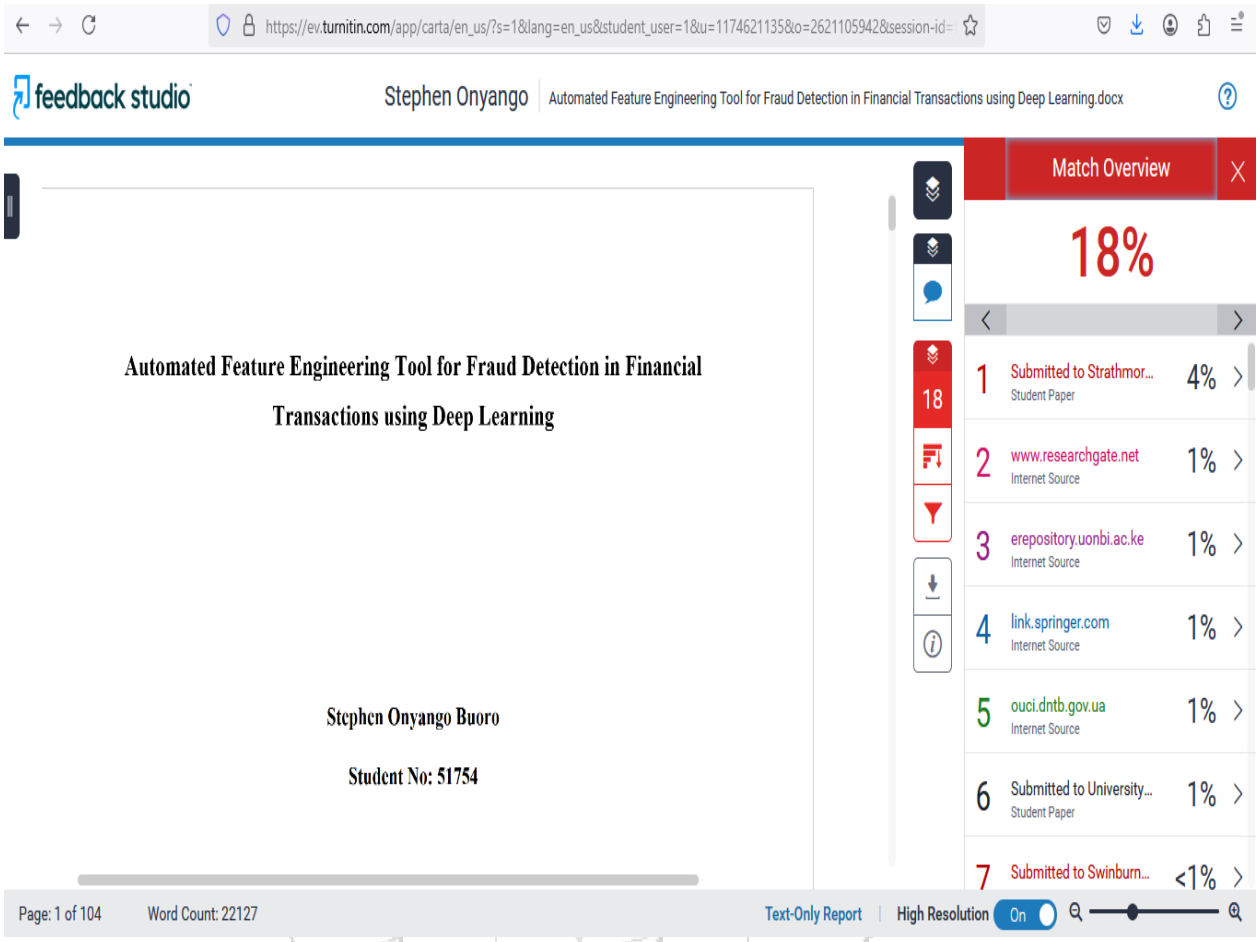
Yang, J., Tang, Z., Guan, Z., Hua, W., Wei, M., Wang, C., & Gu, C. (2021). Automatic Feature Engineering-Based Optimization Method for Car Loan Fraud Detection. *Discrete Dynamics in Nature and Society*, *2021,* 1–10. https://doi.org/10.1155/2021/6077540

Zhu, X., Ao, X., Qin, Z., Chang, Y., Liu, Y., He, Q., & Li, J. (2021). Intelligent Financial Fraud Detection Practices in Post-Pandemic Era: A Survey. *The Innovation*, *2*(4), 100176. https://doi.org/10.1016/j.xinn.2021.100176

Zubair, M. (2023). A Brief History of Information Theory by Claude Shannon in Data Communication. *Journal of Applied and Emerging Sciences*, *13*(1), 23–30. https://doi.org/10.36785/jaes.131550

Appendices

Appendix A: Similarity Report



Appendix B: Ethical Clearance Confirmation

Strathmore UNIVERSITY

7th February 2025

Mr Buoro Stephen,
stephen.buoro@strathmore.edu

Dear Mr Buoro,

RE: Automated Feature Engineering Tool for Fraud Detection in Financial Transactions using Deep Learning

This is to inform you that SU-ISERC has reviewed and **approved** your above **SU-masters** proposal. Your application reference number is **SU-ISERC2571/25**. The approval period is from **7th February 2025 to 6th February 2026**.

This approval is subject to compliance with the following requirements:

i. Only approved documents including (informed consents, study instruments, MTA) will be used.
ii. All changes including (amendments, deviations, and violations) are submitted for review and approval by SU-ISERC.
iii. Death and life-threatening problems and serious adverse events or unexpected adverse events whether related or unrelated to the study must be reported to SU-ISERC within 72 hours of notification.
iv. Any changes anticipated or otherwise that may increase the risks or affected safety or welfare of study participants and others or affect the integrity of the research must be reported to SU-ISERC within 72 hours.
v. Clearance for the export of biological specimens must be obtained from relevant institutions.
vi. Submission of a request for renewal of approval at least 60 days prior to the expiry of the approval period. Attach a comprehensive progress report to support the renewal.
vii. Submission of an executive summary report within 90 days of completion of the study to SU-ISERC.

Before commencing your study, you will be expected to obtain a research license from National Commission for Science, Technology, and Innovation (NACOSTI) https://research-portal.nacosti.go.ke/ and obtain other clearances needed.

Yours sincerely,

Mr Ambrose Rachier,
Chairperson; SU-ISERC

Ole Sangale Rd, Madaraka Estate, PO Box 59857-00200, Nairobi, Kenya. Tel +254 (0)703 034000
Email admissions@strathmore.edu www.strathmore.edu

Appendix C: Dataset Description

The paysim (https://www.kaggle.com/datasets/ealaxi/paysim1) dataset selected for developing a model to detect fraud in financial transactions consists of several features related to transaction activities. Here's a detailed explanation of the new features that will be created using automated feature engineering:

Description of Automatically Generated Features:

Core Transactional and Balance Features

i). step

Represents the discrete time interval or step when the transaction occurred. This is often used as a proxy for the time dimension in the dataset.

ii). type

Denotes the transaction type, such as payment, transfer, cash-out, etc. This categorical feature helps distinguish between different modes of transaction.

iii). amount

The monetary value involved in the transaction. It is a fundamental feature for identifying unusually large or suspicious transactions.

iv). oldbalanceOrg

The account balance of the originator (sender) before the transaction took place. This baseline helps to measure changes due to the transaction.

v). newbalanceOrig

The balance of the originator after the transaction has been processed, useful to confirm the effect of the transaction on the sender's funds.

vi). oldbalanceDest

The balance of the recipient (destination) account before the transaction, providing a baseline for detecting unusual credits.

vii). newbalanceDest

The balance of the recipient account after the transaction completion, indicating the net impact on the destination.

viii). isFlaggedFraud

A binary indicator flagging transactions marked as suspicious or fraudulent by external rules or heuristics.

Cumulative Mean Features (CUM_MEAN)

i). The CUM_MEAN features represent the running average or cumulative mean of a given variable up to the current transaction step. This smoothing over time highlights evolving patterns and trends:

ii). CUM_MEAN(amount)

The cumulative average transaction amount up to the current step, helping identify whether the current transaction is unusually large or small relative to past activity.

iii). CUM_MEAN(isFlaggedFraud)

The running average of the fraud flags, reflecting the proportion of flagged transactions over time.

iv). CUM_MEAN(isFraud)

Similar to the above, this reflects the cumulative average of confirmed fraudulent transactions, which may differ from flagged transactions.

v). CUM_MEAN(newbalanceDest)

The average recipient balance at transaction steps leading up to the current one, potentially capturing account growth or depletion trends.

vi). CUM_MEAN(newbalanceOrig)

The average originator balance after transactions up to the current step, indicating typical sender account status over time.

vii). CUM_MEAN(oldbalanceDest)

The average recipient balance before transactions, useful for establishing baseline account status.

viii). CUM_MEAN(oldbalanceOrg)

The average originator balance before transactions, allowing for comparison with transaction amounts.

ix). CUM_MEAN(step)

The average transaction step number up to the current transaction, useful in time-series contexts to track progression.

Cumulative Sum Features (CUM_SUM)

i). The CUM_SUM features capture the cumulative total or running sum of a variable through the transaction history. These cumulative sums can indicate aggregated activity levels:

ii). CUM_SUM(amount)

Total transaction amount processed by the user or system up to the current step.

iii). CUM_SUM(isFlaggedFraud)

Total count of transactions flagged as suspicious so far.

iv). CUM_SUM(isFraud)

Total confirmed fraudulent transactions encountered up to the current step.

v). CUM_SUM(newbalanceDest)

Summation of recipient balances after transactions, showing cumulative account growth or inflows.

vi). CUM_SUM(newbalanceOrig)

Total originator balances post-transaction, reflecting cumulative outflows or spending.

vii). CUM_SUM(oldbalanceDest)

Total recipient balances prior to transactions, indicating baseline totals.

viii). CUM_SUM(oldbalanceOrg)

Total originator balances before transactions.

ix). CUM_SUM(step)

Running total of the step indices, often used in temporal analyses.

Percentile Features (PERCENTILE)

i). The PERCENTILE features estimate the relative standing of the current transaction's variable compared to the distribution of past transactions. This contextualizes values within a probabilistic framework:

ii). PERCENTILE(amount)

The percentile rank of the current transaction amount among all previous amounts, highlighting outliers or unusually large transactions.

iii). PERCENTILE(isFlaggedFraud)

The percentile rank based on flagged fraud occurrences, useful in risk profiling.

iv). PERCENTILE(isFraud)

Similar to the above but for confirmed frauds, which may be more precise.

v). PERCENTILE(newbalanceDest)

The percentile rank of the recipient's post-transaction balance relative to past balances.

vi). PERCENTILE(newbalanceOrig)

The originator's post-transaction balance percentile rank, helping to detect unusual balances.

vii). PERCENTILE(oldbalanceDest)

The percentile rank of the recipient's pre-transaction balance.

viii). PERCENTILE(oldbalanceOrg)

The percentile rank of the originator's pre-transaction balance.

ix). PERCENTILE(step)

Reflects the position of the current transaction step relative to all prior steps, contextualizing timing.